Proceedings of ICCM 2019

17th International Conference on Cognitive Modelling[1]

Editted by

Terrence C. Stewart

[1] Co-located with the 52nd Annual Meeting of the Society for Mathematical Psychology in Montreal, Canada.

Proceedings of the 17th International Conference of Cognitive Modelling (ICCM 2019)

Preface

The International Conference on Cognitive Modeling (ICCM) is the premier conference for research on computational models and computation-based theories of human cognition. ICCM is a forum for presenting and discussing the complete spectrum of cognitive modelling approaches, including connectionism, symbolic modeling, dynamical systems, Bayesian modeling, and cognitive architectures. Research topics can range from low-level perception to high-level reasoning. In 2019, ICCM was jointly held with MathPsych – the annual meeting of the Society for Mathematical Psychology at the Le Centre Sheraton hotel, in Montreal, Quebec, Canada, on July 19th – 22nd.

Acknowledgements

We would like to acknowledge the Society for Mathematical Psychology (SMP), Elsevier, and Springer, whose combined support kept the conference fees low and allowed us to fund student awards. We also would like to acknowledge the people who brought the MathPsych and ICCM conferences together for the first time (Andrew Heathcote, Amy Criss, Frank Ritter, and David Reitter), the hard work of the MathPsych Conference Chair (Joachim Vandekerckhove), and the officers of the SMP (Jennifer Trueblood, Scott Brown, and Leslie Blaha) for their logistical support. EasyChair was used to manage submissions and reviews.

Papers in this volume may be cited as:

Lastname, A., Lastname, B., & Lastname, C. (2019). Title of the paper. In Stewart, T.C. (Ed.). Proceedings of the 17th International Conference on Cognitive Modelling (pp. 6-12). Waterloo, Canada: University of Waterloo.

ISBN-13: 978-0-9985082-3-8, published by the Applied Cognitive Science Lab, Penn State.

Conference Committees

General and Program Chair

Terrence C. Stewart	National Research Council of Canada

Program Committee

Burcu Arslan	Educational Testing Service
Leslie Blaha	Air Force Research Laboratory
Jelmer Borst	University of Groningen
Edward Cranford	Carnegie Mellon University
Christopher Dancy	Bucknell University
Cvetomir Dimov	Carnegie Mellon University
Pooyan Doozandeh	The Pennsylvania State University
Olivia Guest	University College London
Stefan Huijser	University of Groningen
Preston Jiang	University of Washington
Wouter Kruijne	University of Groningen
Othalia Larue	Université du Québec à Montréal
Christian Lebiere	Carnegie Mellon University
Peter Lindes	University of Michigan
Konstantinos Mitsopoulos	Carnegie Mellon University
David Peebles	University of Huddersfield
Kai Preuss	TU Berlin
Roussel Rahman	Rensselaer Polytechnic Institute
Patrick Rice	University of Washington
Nele Russwinkel	TU Berlin
Dario Salvucci	Drexel University
Florian Sense	University of Groningen
Nick Sexton	University College London
Catherine Sibert	Rensselaer Polytechnic Institute
Sterling Somers	Carleton University
Andrea Stocco	University of Washington
Chris Street	University of Huddersfield
Niels Taatgen	University of Groningen
Farnaz Tehranchi	The Pennsylvania State University
Hedderik van Rijn	University of Groningen
Robert West	Carleton University
Yuxue Yang	University of Washington

Table of Contents

Cognitive Models as a Computational Correlate of Theory of Mind for Human-Machine Teaming

Leslie M. Blaha (leslie.blaha@us.af.mil)
Air Force Research Laboratory, Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract

I delve into an initial discussion on the nature of the theories of mind needed to support effective human-machine teaming. Effective human-machine teaming will require humans to have a theory of mind about machine intelligence and for machine intelligence to have a theory of mind about human teammates. The latter will require a machine to be able to make inferences about the cognitive states related to observable behaviors by the human and to predict future states and actions consistent with the human's beliefs, goals, and desires. This paper proposes that cognitive models can provide the computational correlates to enable a machine theory of mind to reason about its human counterparts.

Keywords: Human-machine teaming; Computational cognition; Cognitive models; Human information processing; Theory of mind

Introduction

The purpose of this paper is to spark an exploration around the nature of the theory of mind required to support human-machine intelligence teaming. I begin with the claim that a theory of mind is necessary for humans and machine intelligence to work together in collaborative teaming situations. These are situations in which a machine has autonomous capability, meaning it can act alone without human supervision or direct intervention, can take direction or feedback from a human, can give direction or feedback to a human teammate, and leverages some form of artificial intelligence to process information, learn and adapt to complete tasks and achieve the team's goals.[1]

Human-machine teaming of this type is predicated on the assumption that humans and machine intelligence understand each other. We can see this in claims that increasing transparency of automation will allow humans to properly calibrate their trust and reliance on the technology (Lee & See, 2004). Or it is similarly implied in the claims that artificial intelligence endowed with the ability to explain its decisions (so called explainable AI or XAI) will aid human users to reason about the correctness and sources of error in the machine's output (Hoffman, Klein, & Mueller, 2018). The push for real-time state assessment in humans is partially driven by the goal of representing the human in ways that can be interpreted and adapted to by machine systems (e.g., Borghetti & Rusnock, 2016). Across these research topics and engineering endeavors, there is a common theme of measuring, identifying, and representing the unobservable states of agents to make them understandable to the other team members, particularly between heterospecific team members.

We have been implicitly demanding a theory of mind to support effective human-machine teaming.

Theory of Mind Defined for Human-Machine Teams

Theory of mind (ToM) is the term ascribed to the processes an agent uses to impute the internal "mental" states of itself and other agents (c.f. Fodor, 1992; Mahy, Moses, & Pfeifer, 2014; Premack & Woodruff, 1978). Note that herein, I am using the term mental state both for humans and machines to refer to the internal information processing mechanisms and representations that are only indirectly observable by the other agent. In various social and developmental lines of ToM research, this inference process is usually considered conceptually from the perspective of an exemplar human or primate, the "subject" of the study. The social interactions, and therefore relevant ToM, is about the subject's ability to reason about itself and one or a small number of other agents, usually other humans.

One level of reasoning within ToM emphasizes the subject's ability to interpret observed actions of the other as goal-directed behaviors. That is, the ToM must support the interpretation of a sequence of actions as representing a trajectory through a state space toward a goal state. Any time the agent is seeking the same goal state, it is likely to exhibit similar sequences of behaviors. A subject could reason over these trajectories to abstract a degree of meaning about the goals driving the observed behaviors. However, ToM is usually invoked at a deeper level: the inferences by the subject should be representing the intentions, emotions, prior experiences, mental state, awareness, and goals of the other agent. That is, we hypothesize that a subject capable of full ToM is attempting to represent to him or herself the latent factors within another agent that contextualize the goal-oriented behaviors.

The dominant theories about ToM generally argue that either people rely on their own mental mechanisms to simulate the experiences of other agents (e.g., Scholl & Leslie, 1999), or they rely on their ability to reason over internal conceptual representations of cognitive mechanisms (e.g., Gopnik & Wellman, 1994). A key commonality across theories is the reliance on an internal representation of the mechanisms of mind. This brings us to the crux of the challenges in defining a ToM for human-machine teaming, which can be summarized in three questions:

1. What are the mechanisms of mind for machine intelli-

[1] At this junction, I am agnostic to whether that intelligence is embodied in a robotic form and to the specific nature of the interactions and communications between the human and machine intelligence. These details not change the present argument, though are critical for engineering actual systems.

gence?

2. How do we represent machine mechanisms of mind in humans to be reasoned over?

3. How do we represent human mechanisms of mind in machines to be computed about?

The nature of human-machine teaming and the fundamental differences between human cognition and computational processes require that we expand the concept of ToM to include multiple types of ToM models and mechanisms. In spite of our often-useful analogy of cognition as computation, the nature of the ToM for machines reasoning about machines, machines reasoning about humans, and humans reasoning about machines must be different than human ToM about other humans. Elucidating the nature of these new theories of mind is a hard problem. Indeed, I note that developing an artificial theory of mind to support human-robot interaction was listed as one of the top grand challenges in humanoid robots today (Yang et al., 2018).

The human ToM within a human-machine team will likely operate as a classical ToM: introspection about self and introspection about other people (particularly for multi-human, multi-agent team configurations) will continue to engage processes of simulating and theorizing about mental states based on our own experiences with self and interacting with other people. But now human ToM must also provide introspection about machine intelligence. Properly supporting such heterospecific introspection will require the development of appropriate mental models for machine intelligence capabilities. Deeper discussion about human mental models of machine intelligence is beyond the current scope.

Let us make the working assumption that a machine ToM parallels human ToM. It must enable a machine intelligence to "introspect" about itself.[2] It must enable a machine intelligence to introspect about other machine agents. In some cases, the other agents may employ similar artificial intelligence algorithms, but machine learning, which is sensitive to input data and conditions, may have produced deviating internal representations of the world. In other cases, other machine agents may have completely different algorithms, chip architectures, and system structure. It could potentially take a complex set of representations and savvy abstractions to enable machines to reason about other agents. Recently, Rabinowitz and colleagues (2018) have made headway in developing machine ToM that abstracts all agent behaviors into state-action trajectories and engages pattern recognition for inferences between agents (see also Winfield, 2018, for a candidate abstraction in robots).

Finally, a machine ToM for human-machine teaming must enable the machine to reason about human teammates. I argue that it will not be enough to abstract a human into a simple, observable state-action sequence for pattern recognition. Analogous to human ToM, the machine intelligence will need to make inferences about the mechanisms of mind, the emotions, intentions, beliefs, and goals of the humans. There may also be cases where the machine must make inferences about physical states and capabilities, too.

The reason we must go beyond simple state-action pattern recognition is that our intentions for human-machine teaming capabilities entail intelligent machines that anticipate and adapt to their human teammates in addition to adapting to dynamic task environments and data. This will require that machines can predict *future* human states and likely actions (and sometimes likely consequences).[3] For machine ToM about humans to achieve prediction or anticipation, it must incorporate a representation of the internal states, intention, beliefs, and goals of the human. It is not enough for the machine intelligence to be reactive to the behavior or action of the human, which may facilitate pattern recognition but not prediction of future actions contextualized by the mental state of the human teammate. It is here that cognitive models of the mental mental mechanisms and processes supporting the human states have a critical role to play.

Cognitive Models in the Machine ToM

We now come to a primary question for consideration by the cognitive modeling community: can cognitive models provide the algorithmic framework(s)—computational correlates, if you will—to enable machine intelligence to have a ToM about human teammates? A limitation of the few current artificial theories of mind is that they do not offer a human-specific representation that differentiates human teammates from other environment variables or computational agents, though the need for such representations to support effective interactions is recognized within social robotics at least (Yang et al., 2018). Winfield (2018) states that the artificial ToM for robots based on a consequence engine is most effective for conspecific agents; that is, reasoning about another agent is most effective when the agent is the same type as the robot. Scassellati (2002) had demonstrable success integrating models of fundamental perceptual skills into humanoid robots to encourage behaviors consistent with the emergence of higher level ToM-related behaviors (e.g., gaze tracking). While behavior consistent with a machine ToM about human teammates is promising, we can go further by not only leveraging models of elements of perception and cognition but leveraging models instantiating full decision-action processes and information processing systems or even full architectures of cognition *and* conceptualizing them as the machine's ToM about the human teammate. In this way, the cognitive models provide a computationally tractable representation of human mental mechanisms, states, beliefs, intentions, and goals—

[2] I use the term introspection here loosely and without proper definition at the present time. This definition will need to delve into the nature of computational inference and state assessment of computational algorithms, which is beyond the present scope.

[3] I note for completion that there is an analogous need for humans to predict the future states, likely actions, and likely consequences of machine activity in the human-machine team. This is related to the need to examine the nature of human mental models about machine intelligence and is left to future exploration.

all those elements critical for deeper introspection within a ToM. And because computational model implementations are in computational languages, they can be integrated into system architectures and intelligent processes.

We must ask then, if cognitive models are to be thought of as a correlate for machine ToM about humans, do they provide the same support to machines that neural correlates of human ToM provide to humans? Within their review of neural correlates of ToM from the social and developmental psychology perspectives, Mahy et al. (2014) offer some initial criteria we can use to evaluate conceptual consistency.

A correlate for ToM should support mental simulation. One key hypothesis for ToM is that people simulate themselves in novel situations and then project inferences about what will happen onto other people (Fodor, 1992; Scholl & Leslie, 1999). Such simulation relies on people having direct access to their own mental states and past experiences. Cognitive models, whether computational cognition formalized in cognitive architectures or mathematical models instantiated in computational algorithms, can simulate human behavior. While the "mental states" of a specific model depend on the mechanisms instantiated in it, generative cognitive models are theoretically grounded in known cognitive mechanisms. In this way, cognitive models might provide machine intelligence teammate direct access to the internal model/mechanism states. Traces of the model history or direct representations of memory, such as declarative memory in ACT-R, provide access to past experiences. The simulated representation of a human (or multiple simulations), can then be compared to observed human behavior to further inform the machine ToM.

A correlate for ToM should be modular in nature. Multiple theories of mind postulate the existence of dedicated, even innate, neural correlates and cognitive mechanisms supporting reasoning about self and others. Modularity of mechanisms is important for the reasoner to keep the inferences about self separate from inferences about others. In our case, then we want to construct human-machine teaming systems where the cognitive models constitute their own module that keeps the representation of human teammates unique from the representations of the task, environment, data or machine's own capabilities. It is not inconsistent to consider the cognitive models within the machine intelligence in a modular way. Designing machine intelligence-based systems in a modular way would enable the system to access its ToM about human teammates when operating with those teammates and to operate autonomously when the human teammates are not present. The representation of the human remains consistent even as the structure or mission of the team changes.

A correlate for ToM supports reasoning over multiple perspectives. A mature ToM is able to hold multiple perspectives in working memory and reason over them independently. This helps someone to differentiate inferences about themselves from inferences about each other individual. Cognitive models have been used as independent agent representations within larger systems. One example is the use of model to support human-robot interaction using ACT-R to simulate human predictions to inform robot planning (Lebiere, Jentsch, & Ososky, 2013). This system enables reasoning about potential human states together with computation about the robot itself. Another example is the development of cognitive-model based synthetic teammates for training (Ball et al., 2010) where the system tracks the synthetic agent and models human learning behavior simultaneously. As long as they are incorporated into systems with adequate processing resources, cognitive models are capable of being used in a modular way in parallel with all other relevant machine intelligence algorithms and artificial ToM about other machine agents.

A correlate for ToM should support theoretically grounded conceptual learning. Human ToM evolves over time, as people learn about themselves and others. They move from simpler to more complex conceptual representations. They evolve to account for observations about other that are inconsistent with currently held conceptions. It is argued that relevant conceptual knowledge must reside in theory-like structures that support the human ToM (Gopnik & Wellman, 1994).

Cognitive models are theoretically grounded in the mechanisms of cognition. As such, they can provide the theoretical structures needed for evolution of conceptual understanding about the human within the machine ToM. Cognitive models can further be equipped with human-like learning mechanisms that enable the model representations to evolve in human-like ways. Consistent with the assumptions of ToM development, this concept learning can be captured through experiential changes in the model and age-related changes in a model operating at a longitudinal scale. This is critical for the machine to have conceptual, or theoretically grounded, representations of how the human's mental state is or could be changing, even if the machine is not learning or reasoning in a human-like way.

Open Questions

Conceptually, cognitive models are capable of supporting machine ToM about human teammates. As we are early in the process of exploring ToM for human-machine teaming, there are a number of open questions that must be debated, including but not limited to:

- Do we need full computational cognitive architectures or unified theories of cognition instantiated in machine intelligence to make useful inferences?

- How detailed must a human's mental model of the machine be for useful inferences?

- What are the critical tests that a cognitive-model based machine ToM is, in fact, a full theory of mind?

- When would it disadvantage a human-machine team to require a full ToM in the system?

As we evolve our vision for the capabilities of human-machine intelligence teams, and as the evolution of such teams changes the way we even conceive of what be capable, the need for the human and machine to understand each other will remain a constant system requirement. It behooves us to consider now what it means for humans and machines to understand each other and how establishing human-machine teaming theories of mind will inform that understanding. Cognitive models have an important role to play in meeting the grand challenge of developing an artificial theory of mind and a critical role to play when those artificial minds interact with our own.

Acknowledgments

The author wishes to thank Christian Lebiere and John Anderson for stimulating discussions on this topic. This work was sponsored by a seedling grant from the 711th Human Performance Wing Chief Scientist. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

Ball, J., Myers, C., Heiberg, A., Cooke, N. J., Matessa, M., Freiman, M., et al. (2010). The synthetic teammate project. *Computational and Mathematical Organization Theory*, *16*(3), 271–299.

Borghetti, B. J., & Rusnock, C. F. (2016). Introduction to real-time state assessment. In *International conference on augmented cognition* (pp. 311–321). Springer.

Fodor, J. A. (1992). A theory of the child's theory of mind. *Cognition*, *44*(3), 283–296.

Gopnik, A., & Wellman, H. M. (1994). The theory theory. *Mapping the mind: Domain specificity in cognition and culture*, 257.

Hoffman, R. R., Klein, G., & Mueller, S. T. (2018). Explaining explanation for Explainable AI. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 62, pp. 197–201).

Lebiere, C., Jentsch, F., & Ososky, S. (2013). Cognitive models of decision making processes for human-robot interaction. In *International conference on virtual, augmented and mixed reality* (pp. 285–294).

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, *46*(1), 50–80.

Mahy, C. E., Moses, L. J., & Pfeifer, J. H. (2014). How and where: Theory-of-mind in the brain. *Developmental Cognitive Neuroscience*, *9*, 68–81.

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, *1*(4), 515–526.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S., & Botvinick, M. (2018). Machine theory of mind. *arXiv preprint arXiv:1802.07740*.

Scassellati, B. (2002). Theory of mind for a humanoid robot. *Autonomous Robots*, *12*(1), 13–24.

Scholl, B. J., & Leslie, A. M. (1999). Modularity, development and theory of mind. *Mind & Language*, *14*(1), 131–153.

Winfield, A. F. T. (2018). Experiments in artificial theory of mind: From safety to story-telling. *Frontiers in Robotics and AI*, *5*, 75.

Yang, G.-Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of science robotics. *Science Robotics*, *3*(14), eaar7650.

On the Matter of Aggregate Models for Syllogistic Reasoning: A Transitive Set-Based Account for Predicting the Population

Daniel Brand* (daniel.brand@cognition.uni-freiburg.de)
Nicolas Riesterer* (riestern@cs.uni-freiburg.de)
Marco Ragni (ragni@cs.uni-freiburg.de)
Cognitive Computation Lab, Georges-Khler-Allee 79
79110 Freiburg, Germany

Abstract

Recent work in modeling human syllogistic reasoning claimed that heuristic approaches perform worse in accounting for experimental data than more comprehensive representations of cognition. We show that this observation might have been due to a misconception of the goals heuristics are often developed for: representing a specific psychological phenomenon or reflecting individual inference strategies. To demonstrate the performance of heuristic models, we introduce a novel model for syllogistic reasoning fundamentally based on transitivity. By evaluating it based on predicting the most frequent answer, i.e., the response most often selected by participants, we show that this model is able to outperform the current state of the art, demonstrate the promising role of transitive inferences in syllogistic reasoning, and discuss its implications for modeling individual reasoners instead of populations.

Keywords: syllogistic reasoning; predictive modeling; heuristics; transitivity

Introduction

Syllogistic reasoning is, next to conditional and relation reasoning, one of the core domains of human reasoning research (Evans, 2002). Syllogisms are quantified statements of the form “All pilots are painters, Some painters are divers” consisting of two premises which are constructed by relating two terms A-B (i.e., pilots-painters), and B-C (i.e., painters-divers) via one quantifier out of “All, Some, No, Some ... not” (for additional background information see Khemlani & Johnson-Laird, 2012). Depending on the order of terms in the premises, syllogisms can be classified into four figures:

Figure 1	Figure 2	Figure 3	Figure 4
A-B B-C	B-A C-B	A-B C-B	B-A B-C

The goal of syllogistic reasoning tasks is to use the information of the premises which are related to each other via the middle term B in order to draw a conclusion about the end terms A, C by using one of the quantifiers mentioned above or infer “No Valid Conclusion” (NVC) if there is none. In total, by considering all combinations of quantifiers and figures, there are 64 distinct syllogistic problems with nine possible conclusions causing the domain to be well-defined and accessible for cognitive modeling endeavors. To increase readability of the syllogistic problems, quantifiers will be represented in accordance to their traditional latin abbreviations (originating from “affirmo” and “nego”) by an uppercase letter for the remainder of the article:

All	Some	No	Some ... not
A	I	E	O

Syllogistic problems are encoded by specifying these quantifier encodings as well as the figural identifier (e.g., AI1 for “All A are B; Some B are C”).

Research shows that human syllogistic inferences differ substantially from classical logics (Wetherick & Gilhooly, 1995). Over the course of the last decades, multiple statistical effects and psychological phenomena were identified and used to formulate hypotheses and theories about mental representations and inferential mechanisms used when reasoning over syllogisms. Traditionally, analyses of syllogistic models are based on aggregated data resulting in models being evaluated in terms of their capability to capture an “average” reasoner. As an example, the authors of a meta-analysis (Khemlani & Johnson-Laird, 2012) relied on hits, correct rejections, and correct predictions to quantify the match between model predictions and experimental data. Their results showed that no satisfactory ordering of model performances could be identified as all theories exhibited distinct strengths and weaknesses with respect to the evaluation metrics.

In this paper, we introduce a novel model for syllogistic reasoning — TransSet — which is based on a heuristic use of transitive inferences. We evaluate the model by focusing on the ability to predict the most frequently given answer (MFA) to a syllogism. This reflects the response given by the “average” reasoner, which lies at the center of population-based analyses. The model’s performance is discussed and compared to the state of the art models in cognitive modeling of syllogistic reasoning. Additionally, since group-level results do not necessarily generalize to the individual level (Molenaar, 2004; Fisher, Medaglia, & Jeronimus, 2018), we investigate the transferability of the results to the level of individuals.

The structure of the remainder of the article is as follows. First, we introduce related literature on cognitive modeling in the field of syllogistic reasoning as well as on statistical effects and psychological phenomena we base our model

*Both authors contributed equally to this manuscript.

on. Second, we give details about the model's computational principles along with an overview of the responses it is able to predict. Third, we perform the predictive analysis of the state of the art and our newly proposed model. Finally, the implications of the results are discussed and directions for future work are suggested.

Related Work

Developing accurate models to explain and predict human responses which differ greatly from classical logics (Wetherick & Gilhooly, 1995) has been a core focus of syllogistic reasoning research in the past decades. Currently, there exist at least twelve cognitive theories attempting to give explanations about the inferential mechanisms inherent to human cognition by relying on a multitude of different methodological foundations (Khemlani & Johnson-Laird, 2012). The authors of a recent meta-analysis (Khemlani & Johnson-Laird, 2012) proposed a classification of the existing theories into *heuristic theories* largely based on simple explanations for differences from classical logics, *formal rule theories* mainly proposing logic-based inference mechanisms, and *theories based on diagrams, sets, or models* which focus on mental representation of information and corresponding inferential operations.

Recently, an increasing effort has been made to turn abstract and often underspecified cognitive theories of syllogistic reasoning into computational models allowing for an assessment of predictions. In their meta-analysis, Khemlani and Johnson-Laird (2012) compiled prediction tables for most of the cognitive theories resulting in an analysis showing that the existing theories feature distinct predictive properties with respect to hits, correct rejections, and correct predictions. In consequence, no clear ranking of the models' predictive qualities could be determined.

One minor result of the meta-analysis was that heuristic models generally perform worse than more elaborate comprehensive accounts which try to give more detailed explanations about cognition by tying into mental representation, memory, or other components of the human mind (for an example, see the mental models theory, MMT, Johnson-Laird, 1983). However, as recent work shifting the focus of analysis to predicting responses could show, the poor performance of heuristics might have been due to a mismatch of modeling purpose and intent. Since heuristics do not aim at explaining the general population but attempt to formalize specific strategies which may be applied by certain individuals, caution needs to be exercised when analyzing comparative performance evaluations. Indeed, recent work combining heuristics to form a composite portfolio model demonstrated a substantial improvement in performance when leveraging strengths while avoiding weaknesses of specific heuristic accounts (Riesterer, Brand, & Ragni, 2018). A conclusion of this work is that heuristic models should not be underrated in general cognitive modeling. While potentially unsuitable as comprehensive accounts of human cognition, they might be able to reflect strategies and mechanisms employed by individuals. Because of this they can serve as promising test benches to investigate the role of the numerous statistical effects and psychological phenomena uncovered.

A fundamental concept of human reasoning that has been extensively investigated is transitivity. In the domain of reasoning in particular, the term *pseudo-transitive fallacy* was introduced to describe the phenomenon that human reasoners are prone to drawing transitive inferences even if logically unwarranted (Goodwin & Johnson-Laird, 2008). Some reasoners also assumed transitivity and symmetry when presented with a completely unknown relation (Tsal, 1977).

In the following, we rely on transitivity to develop a novel heuristic model of human syllogistic reasoning which is based on transitive chains of information. The idea to explain syllogistic reasoning based on transitive effects is not new. Guyote and Sternberg (1981) introduced a model which represents information as pairs and integrates set relations via rules applied to transitive chains of information. The difference to what we propose is that transitivity is used as the driving factor for reasoning. Our model assumes transitivity to serve a heuristic purpose allowing humans to avoid relying on higher-level reasoning processes.

A Transitive Model

A major part of the inferences that are drawn on a regular basis in daily life are transitive (e.g., A is bigger than B, B is bigger than C, therefore A is bigger than C). Usually, these kinds of inferences are easy for human reasoners to draw. On the other hand, tasks that look like transitive inference tasks at first glance, when in reality they are not, are prone to errors originating from an unwarranted use of transitivity. It can be assumed that the simplicity and familiarity of transitive tasks plays a major role for this kind of fallacy.

In the following we propose a heuristic model for syllogistic reasoning based on the principle of transitivity. The main assumption of the heuristic model is that some human reasoners try to circumvent a fully fleshed-out inference process by trying to apply simple rules for patterns they are familiar with from transitive inferences. Here lies the major difference to the transitive-chain theory (Guyote & Sternberg, 1981), which is a theory of the human reasoning process instead of a heuristic which might be used by some reasoners to avoid in-depth inference processes by applying shallow transformations to obtain familiar patterns.

The general process of TransSet is sketched in Figure 1. Its first step focuses on determining the direction of the syllogism by looking for a transitive pattern A-B-C or C-B-A. Such patterns can be found directly for syllogisms with figure 1 and 2, corresponding to a path from A to C and from C to A, respectively.

For Figure 3 and 4 this process fails, which leads to an NVC response in most cases. In some cases, however, a path can be constructed by changing the direction of one of the premises: Figure 3 syllogisms consist of two premises featur-

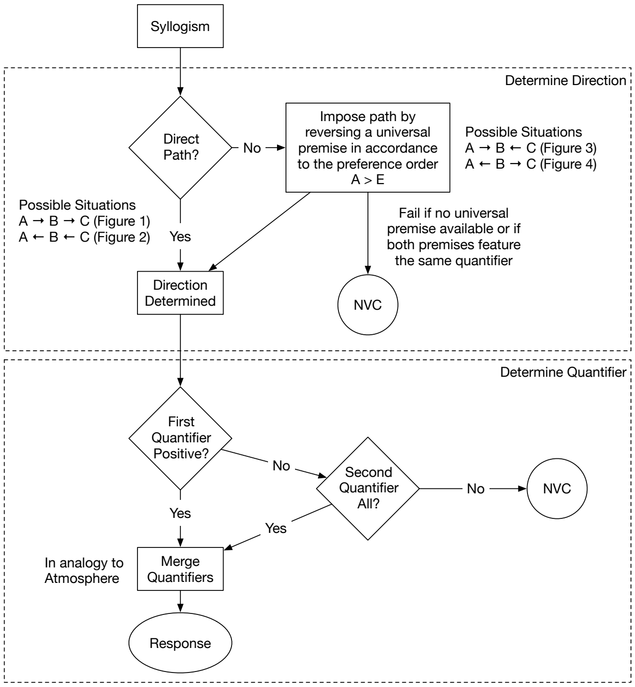


Figure 1: Flow of the TransSet model. First, the direction is determined by extracting a transitive path from the premise information. Second, the quantifier is determined by merging quantifiers. In case of failures resulting from insufficient information or disconnected premises, NVC is generated.

ing paths to B, while Figure 4 syllogisms yield paths starting from B. In both cases it has to be decided which categorical set (A or C) should be put in the place of B. At first glance, it might seem reasonable to choose the set of elements that has the most informative quantifier as it is able to "compensate" for the uncertainty introduced by changing the premise direction. We consider the universal quantifiers A and E as "informative", since they make statements about all elements in a set. The TransSet model therefore changes the direction of the premise with the most informative universal quantifier, if there is a single "most informative" universal quantifier (with an ordering of $A > E$). In case of ties or a lack of informative quantifiers, the process fails and returns NVC. Note, that the change of direction requires the assumption of symmetry, which is logically invalid for the quantifier A. The occurrence of this deviation from classical logic in human reasoning behavior is also a core concept of the conversion theory (Revlis, 1975).

As soon as a path is obtained, the task can be solved by propagating the starting set of elements along the path (while applying the quantifiers). For example, considering syllogism AI1 (All A are B, Some B are C), a set consisting of all A is propagated to B, where it is filtered by the second quantifier on its path from B to C, reducing the set to "Some A". Therefore, the conclusion would be "Some A are C", which is logically invalid. It is important to note that the process of set propagation yields the same conclusion quantifiers as the atmosphere theory (Wetherick & Gilhooly, 1995), but also predicts the direction of the answer: a path from A to C naturally corresponds to an answer with the direction $A \rightarrow C$. The resulting predictions are in line with the figural effect (Johnson-Laird, 1983).

The propagation, however, does not succeed in all cases. When the set obtained after filtering by the first quantifier is empty, traversing the transitive path is no longer possible. For example, when considering syllogism EI1, the set after the path $A \rightarrow B$ would be empty, as there are no elements from A that are also B. It is therefore not possible to integrate the second quantifier, as the set cannot be reduced any further. This leads to the NVC response, since the endpoint of the path cannot be reached. An exception to this can occur if the second quantifier is A: because A does not require any filtering, it corresponds to simply passing the set ahead, which prevents the path from breaking. Note, that this failure of the propagation induces an asymmetry regarding the quantifier which is not generally assumed in heuristic models: since it can only happen if the first processed quantifier leads to an empty set, syllogism EI1 and IE2 are affected but IE1 and EI2 are not.

The TransSet model is a heuristic model. As such, it only describes a single heuristic strategy assumed to be used by some human reasoners for syllogistic reasoning. Therefore, we used the heuristic in a strictly deterministic setup, where a single prediction for each syllogism was generated according to the procedure described above. The resulting predictions are shown in Table 1.

Analysis

The following analysis is based on the dataset and models reported by Khemlani and Johnson-Laird (2012). Additionally, we included a separate analysis on a dataset of 139 reasoners obtained from a web experiment conducted on Amazon Mechanical Turk which was published as part of the benchmarking framework CCOBRA[1]. This second dataset is not only included to extend the size of the evaluation dataset, but also because it contains unaggregated responses to syllogistic problems which can be used to assess a model's capability to account for individual reasoners. All files related to the following analyses are available on GitHub[2].

MFA Assessment

First, we investigate how accurately models are able to predict the MFA by comparing the set of possible predictions for a given syllogism with the most frequently selected response in the data.

Figure 2 depicts the results of this evaluation based on two different metrics. The left plot presents the proportion of syllogistic problems which feature an MFA response that is contained in the set of possible predictions by the respective model. The obtained values differ substantially be-

[1] https://github.com/CognitiveComputationLab/ccobra
[2] https://github.com/Shadownox/iccm-transset

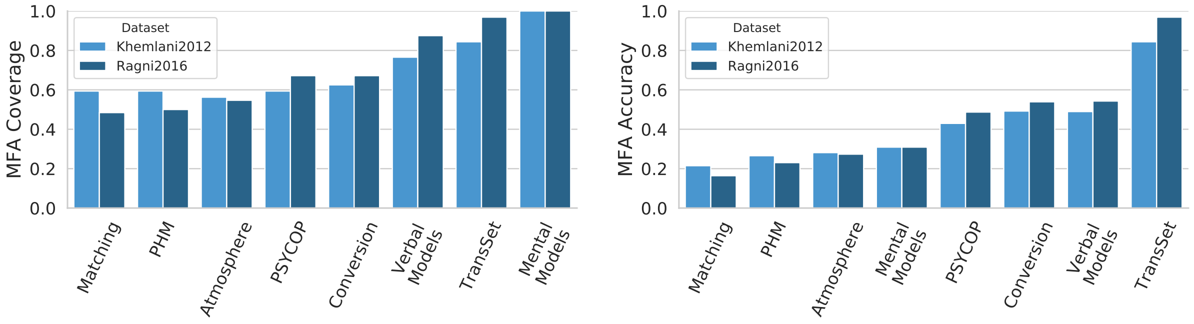


Figure 2: Analysis of predictive performances based on MFA comparison. Left plot depicts proportion of syllogistic problems where at least one of the possible model predictions matches the MFA. Right plot depicts accuracy of predicting MFA (discounted for multiple possible conclusions).

Table 1: Predictions of the TransSet model.

Syllogism	Prediction	Syllogism	Prediction
AA1	Aac	EA1	Eac
AA2	Aca	EA2	Eca
AA3	NVC	EA3	Eac
AA4	NVC	EA4	Eca
AI1	Iac	EI1	NVC
AI2	Ica	EI2	Oca
AI3	Ica	EI3	Oca
AI4	Iac	EI4	NVC
AE1	Eac	EE1	NVC
AE2	Eca	EE2	NVC
AE3	Eca	EE3	NVC
AE4	Eac	EE4	NVC
AO1	Oac	EO1	NVC
AO2	Oca	EO2	NVC
AO3	Oca	EO3	NVC
AO4	Oac	EO4	NVC
IA1	Iac	OA1	Oac
IA2	Ica	OA2	Oca
IA3	Iac	OA3	Oac
IA4	Ica	OA4	Oca
II1	Iac	OI1	NVC
II2	Ica	OI2	Oca
II3	NVC	OI3	NVC
II4	NVC	OI4	NVC
IE1	Oac	OE1	NVC
IE2	NVC	OE2	NVC
IE3	Oac	OE3	NVC
IE4	NVC	OE4	NVC
IO1	Oac	OO1	NVC
IO2	NVC	OO2	NVC
IO3	NVC	OO3	NVC
IO4	NVC	OO4	NVC

tween models. While heuristics such as Matching, the Probabilistic Heuristic Model (PHM), or Atmosphere only contain the MFA response in less than 60% of syllogistic problems, model-based approaches such as the Mental Models Theory (MMT) or Verbal Models are able to achieve above 80%. These observations are in line with the results obtained by Khemlani and Johnson-Laird (2012). However, despite its fundamentally heuristic principles, TransSet is capable to compete with the most performant state of the art models arriving at MFA coverage proportions of above 80% demonstrating that heuristic principles are not generally inferior to more comprehensive models.

A shortcoming of this type of coverage-based analysis is that it ignores the size of the sets of possible model predictions. However, since the more responses a model is allowed to include the higher the possibility is to cover the MFA, models need to be penalized for unnecessary predictions. This is presented in the right plot of Figure 2 which assigns a score of $1/|P_s|$ if the MFA is contained in the prediction set P_s thereby introducing a penalty factor linear in the number of possible predictions. As a result, a model is given a score of 1 if it does not include other responses apart from the MFA for all syllogisms and lower scores if unnecessary conclusions are predicted. For example, the mental models theory captures the MFA “Aac” for syllogism “AA1” in its prediction set $\{\mathrm{Aac}, \mathrm{Aca}, \mathrm{Ica}\}$. As a result it is assigned a score of $1/3$.

This plot draws a different picture of model performances. It shows that when discounting scores based on the number of predictions, performances drop considerably. MMT and Verbal Models which dominated the coverage analysis (left plot) drop substantially due to the fact that they include up to five of the nine possible conclusions in their prediction sets. TransSet on the other hand remains unchanged since it only allows a single prediction to each syllogistic problem.

Put together, both plots demonstrate that the high levels of accuracy achieved by some models (Mental Models, Verbal Models) are mainly due to their large numbers of predicted responses. When compared to TransSet, however, it becomes apparent that complex and potentially parameterized mod-

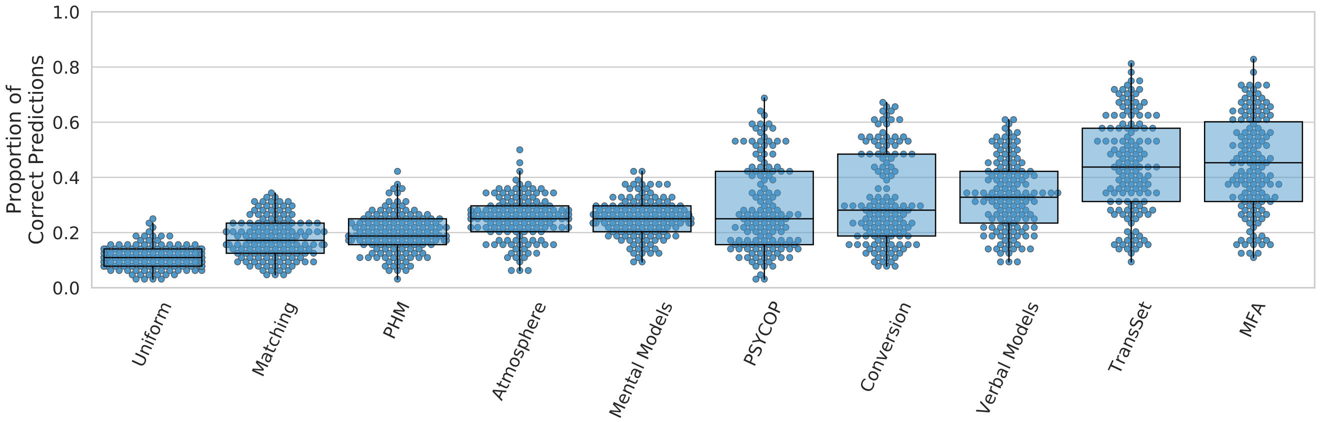


Figure 3: Evaluation of the predictive accuracies of syllogistic models. Boxplots denote medians, inter-quartile range (IQR) as well as whiskers extending to the last data point within a distance of 1.5 times the IQR from the edge of the boxes. Points indicate the accuracy for a specific individual reasoner. Accuracies on the CCOBRA 139 participant dataset of the state-of-the-art models computed from the predictions as reported by Khemlani and Johnson-Laird (2012) are depicted alongside TransSet and two baseline models: “Uniform” corresponds to using a uniform distribution to randomly select an answer and “MFA” reflects the most-frequent answer strategy.

els are unnecessary for predicting aggregated responses. The heuristic principles based on notions of transitivity resulting in a single response suffice to achieve state-of-the-art performance. Note that this analysis focusses on the ability of predicting responses, whereas some models might also be able to provide additional estimates (e.g., reaction times), which are out of scope for the present article.

Individual Match

The results show that TransSet is able to cover a majority of the most frequently given answers and is able to account for populations of reasoners. The pressing question is how relevant MFA is for capturing the variety of strategies that are employed by human reasoners. Put differently, it would be insightful to see whether reasoners differ with respect to their strategies and whether MFA, and accordingly TransSet as its approximation, is a suitable representation for the majority of reasoners.

The second analysis reported in the following therefore shifts the focus towards an assessment of the capability of models to account for the strategies employed by individual human reasoners. In particular, we investigate the match between model predictions and individual responses of the 139 participants contained in the CCOBRA dataset.

Figure 3 depicts the evaluation output obtained from the benchmarking framework CCOBRA. The image depicts the accuracy of individual models when predicting responses for individual reasoners (dot swarm). The box plots present an aggregated representation of these exact results. The image shows that models achieve low predictive accuracies across the board with TransSet surpassing the current state of the art. The swarm plots show that variances of accuracies differ greatly between models. While models on the lower end of the spectrum produce accuracies between 0% and 40%, TransSet is able to predict up to 80% of an individual’s responses correctly.

There are two sides to the results depicted here. On the one hand, it is interesting to see that some of the models are able to successfully predict most of the responses for at least a small part of the population. On the other hand, it shows that not even MFA is able to adequately cover the majority of people. This demonstrates that syllogistic model evaluation solely on aggregated data is severely limited and not necessarily generalizable to individuals. This puts the general goals of cognitive modeling into perspective. A model that claims to reflect cognitive processes or general phenomena of nature in a suitable manner should always be able to achieve high levels of predictive accuracy. If we assume reasoners to rely on a large number of independent strategies this would correspond to models being able to match certain individuals well while completely failing to capture others. This is often the case for heuristics, since the phenomena or cognitive fallacies they are constructed on are only applicable to a subset of individuals in the population. Models accounting for general principles, on the other hand, should generally show a smaller variance in coverage of individuals since the principles should be prevalent in all responses to some degree.

Discussion

In this article we introduced TransSet, a novel model for predicting human syllogistic reasoning. Drawing from the statistical effects and psychological phenomena of the recent literature, TransSet is capable of competing with state-of-the-art models by relying on deterministic and heuristic principles only. When discounted for the number of possible predictions a model generates for a syllogism, TransSet is able to achieve a coverage of MFA of above 80% resulting in an improvement of about 20% over the state of the art as reported by Khemlani and Johnson-Laird (2012).

The main conclusions of this article are twofold. First, we demonstrate that complex parameterized models are not required when aiming for predicting an "average" reasoner, i.e., aggregated data. TransSet, which generates a single deterministic response to each syllogism is not only competing with but outperforms the state of the art when discounting for the number of possible responses. Second, the evaluation of predictive accuracy on individuals highlights that no existing model is able to adequately reflect the reasoning strategy employed by the majority of participants. In order to not only account for a select few reasoners but for a wide variety of individuals, adaptive models tuned to the inferential mechanisms of specific reasoners are required. This, however, remains an open challenge for future work.

TransSet's performance is made possible because it incorporates effects and phenomena uncovered in empirical research. As such it is comprised of ideas found in other models (e.g., transitivity and illicit conversion) and as such can be understood as a superset of models. The fact that a simple model based on heuristic principles is able to outperform the state of the art illustrates the potential that remains in the field. Especially when moving beyond models for aggregated data, the adaptability of parameterized models to individual inferential mechanisms will allow for an even better understanding of cognition and consequently for the development of more accurate models.

Human syllogistic reasoning is far from being solved. In addition to outperforming the state of the art in the aggregate case, TransSet demonstrates a performance that suggests that its underlying concepts form a plausible reasoning strategy for at least some individuals. The heuristic use of transitivity has therefore proven to be a powerful mechanism for explaining human syllogistic reasoning performance and might suggest connections to related results from cognitive science indicating that humans are generally likely to draw transitive conclusions even when they are unjustified (Goodwin & Johnson-Laird, 2008). It remains to be seen if the model can be transferred to other domains featuring transitive properties successfully (e.g., spatial-relation or conditional reasoning). Currently, we only focus on a direct extraction of general output predictions from the model. Future work will focus on two directions: First, we will investigate possible parameterizations allowing the model to fine-tune itself to individual human reasoners. Second, we will investigate further properties of the reasoning process such as reaction times or its connection to the psychological phenomena of syllogistic reasoning.

Acknowledgements

This paper was supported by DFG grants RA 1934/3-1, RA 1934/2-1 and RA 1934/4-1 to MR.

References

Evans, J. S. B. T. (2002). Logic and human reasoning: An assessment of the deduction paradigm. *Psychological Bulletin*, *128*(6), 978–996.

Fisher, A. J., Medaglia, J. D., & Jeronimus, B. F. (2018). Lack of group-to-individual generalizability is a threat to human subjects research. *Proceedings of the National Academy of Sciences*, *115*(27), E6106–E6115.

Goodwin, G. P., & Johnson-Laird, P. (2008). Transitive and pseudo-transitive inferences. *Cognition*, *108*(2), 320–352.

Guyote, M. J., & Sternberg, R. J. (1981). A transitive-chain theory of syllogistic reasoning. *Cognitive Psychology*, *13*(4), 461–525.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA, USA: Harvard University Press.

Khemlani, S., & Johnson-Laird, P. N. (2012). Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, *138*(3), 427–457.

Molenaar, P. C. M. (2004). A manifesto on psychology as idiographic science: Bringing the person back into scientific psychology, this time forever. *Measurement: Interdisciplinary Research & Perspective*, *2*(4), 201–218.

Revlis, R. (1975). Two models of syllogistic reasoning: Feature selection and conversion. *Journal of Verbal Learning and Verbal Behavior*, *14*(2), 180–195.

Riesterer, N., Brand, D., & Ragni, M. (2018). The predictive power of heuristic portfolios in human syllogistic reasoning. In F. Trollmann & A.-Y. Turhan (Eds.), *Proceedings of the 41st German Conference on AI* (pp. 415–421). Berlin, Germany: Springer.

Tsal, Y. (1977). Symmetry and transitivity assumptions about a nonspecified logical relation. *Quarterly Journal of Experimental Psychology*, *29*(4), 677–684.

Wetherick, N. E., & Gilhooly, K. J. (1995). 'Atmosphere', matching, and logic in syllogistic reasoning. *Current Psychology*, *14*(3), 169–178.

A cognitively plausible algorithm for causal inference

Gordon Briggs and Sangeet Khemlani
{gordon.briggs, sangeet.khemlani}@nrl.navy.mil
Navy Center for Applied Research in Artificial Intelligence
US Naval Research Laboratory, Washington, DC 20375 USA

Abstract

People without any advanced training can make deductions about abstract causal relations. For instance, suppose you learn that *habituation causes seriation*, and that *seriation prevents methylation*. The vast majority of reasoners infer that *habituation prevents methylation*. Cognitive scientists disagree on the mechanisms that underlie causal reasoning, but many argue that people can mentally simulate causal interactions. We describe a novel algorithm that makes domain-general causal inferences. The algorithm constructs small-scale iconic simulations of causal relations, and so it implements the "model" theory of causal reasoning (Goldvarg & Johnson-Laird, 2001; Johnson-Laird & Khemlani, 2017). It distinguishes between three different causal relations: causes, enabling conditions, and preventions. And, it can draw inferences about both orthodox relations (*habituation prevents methylation)* and omissive causes (*the failure to habituate prevents methylation*). To test the algorithm, we subjected participants to a large battery of causal reasoning problems and compared their performance to what the algorithm predicted. We found a close match between human causal reasoning and the patterns predicted by the algorithm.

Keywords: causation; mental models; reasoning; simulation

Introduction

People routinely make inferences about complex causal matters. For instance, consider the following description about a particular farm:

1. Flourishing weeds will cause a lack of nutrients.
   A lack of nutrients will prevent the vegetables from growing.
   The lack of vegetables will enable an early harvest.

What is the relation between the growth of weeds and an early harvest? Reasoners needn't have a background in botany to infer a possible causal relation between the two events, such as in (2):

2. Flourishing weeds will cause an early harvest.

People's inferences are systematic, and at least some errors are obvious, i.e., anyone who infers (3) from the information in the description above is grossly mistaken:

3. Flourishing weeds will prevent an early harvest.

How do people infer causal relations between events? Sometimes, perceptual cues may drive people to infer a causal connection between one event and another: if you observe that when a man flips a switch, a particular light goes off, it seems reasonable to infer a causal relation between the switch and the light. Indeed, the temporal contiguity of two events can be sufficient to imply causality (e.g., Lagnado & Sloman, 2006; Rottman & Keil, 2012). But the preceding farming example demonstrates that people can infer causal relations from descriptions, not just observations, and that they can do so in the absence of any explicit temporal information.

How do people make causal inferences? A popular approach in artificial intelligence simulates human causal reasoning using causal Bayes nets and a calculus developed by Pearl (2009). It allows precise calculations of conditional probabilities, e.g., the probability of an early harvest given flourishing weeds, P(*early harvest* | *flourishing weeds*), provided that relevant causal relations are translated into the notation of a graphical network. While the approach can distinguish between causes and mere associations, Pearl's calculus cannot explain how reasoners infer *novel* causal relations where none had existed before, i.e., it cannot explain how people infer (2) from (1).

Cognitive scientists disagree on the mechanisms and representations that underlie causal reasoning (Ahn & Bailenson, 1996; Cheng, 1997; Sloman, 2005; White, 2014; Wolff & Barbey, 2015). Mental simulation is central to many psychological accounts of the process: theorists agree that people construct small-scale simulations to predict outcomes (Kahneman & Tversky, 1982), to understand mechanistic relations (Hegarty, 2004), to comprehend physical scenes (Battaglia, Hamrick, & Tenenbaum, 2013), to resolve inconsistent and contradictory information (Khemlani & Johnson-Laird, 2011), to deduce the consequences of sequences of events (Khemlani, Mackiewicz, Bucciarelli, & Johnson-Laird, 2013), and to make counterfactual inferences (Byrne, 2005; Galinsky & Moskowitz, 2000).

Recent approaches to modeling causal reasoning in AI and cognitive science face two overarching challenges: first, people distinguish between causal relations such as *cause*, *enable*, and *prevent*. They understand, for instance, that (4a) and (4b) mean different things:

4a. A lack of vegetables will *cause* an early harvest.
  b. A lack of vegetables will *enable* an early harvest.

Graphical networks have difficulty capturing the difference between the two relations. Various psychological theories have invoked the transmission of causal forces (Wolff, 2007), causal model structures (Sloman et al., 2009), and mental simulations of possibilities (Goldvarg & Johnson-Laird, 2001) to explain what different causal relations mean (for a review, see Khemlani, Barbey, & Johnson-Laird, 2014). But there exists no robust computational model that predicts what causal relations people generate from descriptions such as (1) above.

Second, most theories of causal reasoning cannot explain reasoning about *omissive* causal relations, such as in (5):

5. A lack of nutrients will cause the vegetables to die.

The assertion is distinct from (2) above because it describes how the absence of an element can bring about some outcome. Philosophers, psychologists, and computer scientists have so much difficulty coping with omissive causation that some philosophers deny it as a meaningful concept (e.g., Beebee, 2004; Dowe, 2001; Hall, 2004). In recent years, psychologists advanced theories to account for omissive causation: some theorists treat omissive causes as an arrangement of causal forces (Wolff, Barbey, & Hausknecht, 2010) or as a set of counterfactual contrasts (Stephan, Willemsen, & Gerstenberg, 2017). But, counterfactuals cannot explain how people reason about future causal relations, such as in (1) above, because the counterfactuals are retrospective by definition. And, forces do not explain why reasoners appear to distinguish omissive causes from omissive enabling conditions and omissive preventions (see, e.g., Khemlani, Wasylyshyn, Briggs, & Bello, 2018).

Hence, students of causal reasoning remain bereft of a feasible, adequate process model of how humans infer causal relations. Our goal in the present article is to specify such an algorithm and to describe its computational implementation. The algorithm is based on the notion that people build iconic simulations of possibilities when they reason, and that they mentally scan those possibilities to infer novel conclusions. Since the goal of the algorithm is to account for human intuitions, we describe an experiment used to benchmark the algorithm, and we show how its implementation matches the performance of human reasoners. We also describe a set of simulations used to validate the parameters in the implementation. We conclude by evaluating the results in the context of contemporary accounts of causal reasoning.

Mental models and causal reasoning

The algorithm for causal inference we present in this paper is based on the tenets of mental model theory – the "model" theory for short. The model theory argues that reasoning depends on the mental simulation of sets of possibilities. The theory is based on three fundamental principles:

- **Mental models represent possibilities.** When people reason about relations, causal or otherwise, they construct one or more possibilities – situations describing finite alternatives – consistent with those relations (Johnson-Laird, 2006; Khemlani, Byrne, & Johnson-Laird, 2018).
- **Mental models are iconic.** The structure of a mental model mirrors the structure of what it represents as far as possible (Peirce, 1931-1958, Vol. 4). An iconic simulation of a causal relation, e.g., *A causes B,* concerns sets of events, *A* and *B*, in a temporal order. Models can also include abstract symbols, e.g., the symbol for negation (Khemlani, Orenes, & Johnson-Laird, 2012) and they can represent sequences of events as they unfold in time (Khemlani et al., 2013).
- **Intuitions depend on one model; deliberations depend on multiple models.** Human reasoning is based on two interacting sets of processes: people form rapid, intuitive inferences by constructing and scanning a single "mental" model, but those intuitive inferences lead individuals to make errors (Khemlani & Johnson-Laird, 2017). Mistakes can be corrected by deliberation, which requires reasoners to consider multiple models by searching for counterexamples to intuitive conclusions (Khemlani & Johnson-Laird, 2013; Khemlani et al., 2015).

The model theory explains why people distinguish between *causes, enables*, and *prevents:* each relation refers to a distinct set of possibilities (Goldvarg & Johnson-Laird, 2001), known as *fully explicit models*. Table 2 shows the fully explicit models for the three relations. For instance, a causal assertion such as (2) above refers to a conjunction of three separate models of possibilities, depicted in this schematic diagram:

weeds	**early-harvest**
¬ weeds	early-harvest
¬ weeds	¬ early-harvest

Each row in the diagram represents a distinct temporally ordered possibility, e.g., the first row represents the possibility in which weeds flourish and then an early harvest occurs. Any possibility missing from the diagram is inconsistent with (2): hence, the situation in which weeds occur and an early harvest does not is incompatible with (2), and so too is any possibility in which an early harvest occurs before the weeds flourish. In contrast, an enabling condition, such as in (6):

6. Flourishing weeds will enable aphids to thrive.

refers to a different conjunction of possibilities:

weeds	**aphids**
weeds	¬ aphids
¬ weeds	¬ aphids

Unlike causes, enabling conditions permit the situation in which the antecedent occurs but the consequent doesn't, e.g., (6) allows for the possibility in which weeds occur but aphids don't thrive. Typically, enabling conditions rule out the possibility in which aphids thrive in the absence of weeds. As Goldvarg and Johnson-Laird (2001) showed, reasoners list these possibilities for assertions such as (2) and (6). Reasoning about causal relations, however, requires significantly more processing than interpreting causal statements, and so when people have to reason, they often do not consider the full list of possibilities – instead, they draw conclusions from just a single possibility, referred to as the *mental model*. The mental models for causes and enabling conditions are bolded in the diagrams above. They're identical, and as a result, individuals often fail to distinguish enabling from causing when they reason (see Experiment 5 in Goldvarg & Johnson-Laird, 2001). Preventions are akin to causes with a negated consequent (see Table 1).

A recent development of the model theory shows that it can explain omissions: the theory treats them as negated antecedents (Khemlani et al., 2018). Hence, the fully explicit models for (5) above are:

```
¬ nutrients     dying-vegetables
  nutrients     dying-vegetables
  nutrients   ¬ dying-vegetables
```

Analogous changes explain omissive enabling conditions and omissive preventions (Table 1). The theory accordingly uses a unified representation for both omissions and orthodox causes.

The model theory explains how people represent causal relations, and various empirical assessments validate the theory's central predictions (Johnson-Laird & Khemlani, 2017). We turn next to describe a novel algorithm and its computational implementation, and we show how to compute inferences from models of possibilities.

Computing with mental models

The algorithm used to infer causal relations relies on three separate subroutines, each of which depends on the representational conventions described in the previous section. First, the algorithm needs to build integrated models of multiple causal relations, e.g., it needs to combine the three sentences in (1) above into a set of models. Second, since reasoners are unlikely to construct models deterministically, the algorithm needs to specify a stochastic system that can mimic the distribution of possible interpretations that humans tend to make. Third, the algorithm needs to explain how people scan models to generate novel relations. We review each subroutine in turn.

Building integrated models

To construct an integrated model from a set of premises, the algorithm adopts a mechanism developed for previous model-theoretic computational implementations (see, e.g., Johnson-Laird & Byrne, 1991): the algorithm takes the Cartesian product of two models with the proviso that a model of an event cannot be combined with its negation. An example will illustrate the process. Consider the premises in (7), both of which concern omissions:

7. A lack of sunlight will prevent the vegetables from growing.
   The lack of vegetables will enable an early harvest.

The mental model of the first premise is:

```
¬ sunlight    ¬ vegetables
```

and the model of the second is:

```
              ¬ vegetables      early-harvest
```

So, a Cartesian product of the two models identifies that the middle event is shared in both models, and it combines the two to create an integrated model:

```
¬ sunlight    ¬ vegetables      early-harvest
```

Suppose instead that people build fully explicit models of the premises, not mental models. The fully explicit model of the first premise is:

```
¬ sunlight    ¬ vegetables
  sunlight    ¬ vegetables
  sunlight      vegetables
```

and the fully explicit model of the second premise is:

```
              ¬ vegetables        early-harvest
              ¬ vegetables      ¬ early-harvest
                vegetables      ¬ early-harvest
```

A procedure implementing the Cartesian product starts by combining the first model of the first premise with the three models of the second premise to yield:

```
¬ sunlight    ¬ vegetables        early-harvest
¬ sunlight    ¬ vegetables      ¬ early-harvest
```

The last model of the second premise is a situation in which vegetables grow, and so it cannot be combined with the first model of the first premise. The same procedure applies to the second and third models of the first premise, and so the full Cartesian product of the two sets of fully explicit models is:

```
8. ¬ sunlight    ¬ vegetables        early-harvest
   ¬ sunlight    ¬ vegetables      ¬ early-harvest
     sunlight    ¬ vegetables        early-harvest
     sunlight    ¬ vegetables      ¬ early-harvest
     sunlight      vegetables      ¬ early-harvest
```

Reasoners are likely to vary in their tendency to interpret causal assertions using mental models or fully explicit models, and so the algorithm implementing the theory uses a stochastic parameter to govern the process: the ε-parameter determines the probability that the algorithm will construct only the mental model or whether it will construct fully explicit models (see, e.g., Johnson-Laird, Khemlani, & Goodwin, 2015; Khemlani & Johnson-Laird, 2013, 2016; Khemlani et al., 2015; for applications of this methodology to quantificational reasoning). The parameter accordingly varies from 0.0 to 1.0 such that when $\varepsilon = 0.0$, the algorithm always produces mental models, and when $\varepsilon = 1.0$, the algorithm produces fully explicit models. Hence, the ε parameter varies the contents of the models.

Varying the size of models

Another parameter, the λ-parameter, controls the number of possibilities that the algorithm yields as it constructs an integrated model. It therefore controls size of the models. This parameter corresponds to the λ-parameter of a Poisson distribution. Consider how the parameter might apply to interpreting the premises in (7). On any given run of the the algorithm, the size of a set of models is governed by $n_{\text{Premise 1}} + n_{\text{Premise 2}}$, both of which are established by two samples drawn from a Poisson distribution of parameter λ. Once the two ns are determined, possibilities are sampled from the fully explicit models and their Cartesian product is taken to yield an integrated mental model. Hence, if $n_{\text{Premise 1}} = 2$, the algorithm would sample 2 separate possibilities from the

	Conjunctions of possibilities yielding different causal relations					
	A causes B	*A enables B*	*A prevents B*	*Not A causes B*	*Not A enables B*	*Not A prevents B*
Fully explicit models	A B	A B	A ¬B	¬A B	¬A B	¬A ¬B
	¬A B	A ¬B	¬A ¬B	A B	¬A ¬B	A ¬B
	¬A ¬B	¬A ¬B	¬A B	A ¬B	A ¬B	A B
Mental model	A B	A B	A ¬B	¬A B	¬A B	¬A ¬B

Table 1. The possibilities consistent with various causal relations in the model theory. Reasoners distinguish between the meanings of relations based on the distinct sets of possibilities – the *fully explicit models* – to which they refer. But, when they make inferences, people often consider just one of the possibilities consistent with the meaning of a relation – the *mental model*. Background knowledge can block the construction of certain models, e.g., *alcohol causes inebriation* is true, and since only alcohol causes inebriation, people should not consider the situation in which inebriation occurs in the absence of alcohol, i.e., the ¬A B model in the first column. A more thorough discussion of strong and weak interpretations is provided in Johnson-Laird and Khemlani (2017).

3 consistent with *not-A prevents B*, which corresponds to the first premise of (7). The same procedure would be used for the second premise. Their Cartesian product would be taken, and since the product concerns sets of fewer possibilities, the resulting integrated model would be a subset of the models in (8) above, e.g.,

```
9. ¬ sunlight   ¬ vegetables     early-harvest
   ¬ sunlight   ¬ vegetables   ¬ early-harvest
     sunlight     vegetables   ¬ early-harvest
```

The algorithm provides two distinct methods of sampling from the possibilities to which the relations refer: the first method samples *n* separate possibilities uniformly; the second samples the possibilities in the order specified by Table 1. Previous empirical results suggest that reasoners list certain possibilities more frequently than others in a manner predicted by the model theory (Bello, Wasylyshyn, Briggs, & Khemlani, 2017). A simulation analysis presented below tests whether random sampling or preferential sampling produces a better to human data.

Generating causal inferences

To generate causal inferences from, e.g., an integrated model such as (9) above, the algorithm reduces the integrated model to a model of its end terms, discarding redundant models where relevant. The reduction process for (9) yields the model in (10) below:

```
10.¬ sunlight                    early-harvest
   ¬ sunlight                  ¬ early-harvest
     sunlight                  ¬ early-harvest
```

The algorithm attempts to match this reduced set of possibilities with all combinations of possibilities in Table 1. If one or more matches can be found in Table 1, the algorithm can form a response by choosing randomly from the corresponding matching relations. In the case of (10), matching relations include*: sunlight prevents an early harvest* and *a lack of sunlight enables an early harvest.*

A more sophisticated response heuristic integrated into the algorithm assesses the first premise of the problem to check whether the antecedent it describes concerns omissive or orthodox causation. For (7), the antecedent – "a lack of sunlight" – concerns omission, the only candidate response is: *a lack of sunlight enables an early harvest.*

To assess whether the algorithm we describe matches human causal reasoning responses, we collected data from participants and compared their responses to those generated by the algorithm.

Experiment and simulations

We conducted an experiment to test the algorithm specified in the previous section. The experiment replicated a design developed by Wolff and Barbey (2015, Experiment 3), in which the authors provided participants with 32 causal reasoning problems of the following form:

X prevents Y.
Y prevents Z.
What, if anything, follows?

In their original study, participants carried out a multiple-choice task in which they selected which responses followed of necessity from 10 possible options. Multiple-choice tasks are limited in their ecological validity – the task encourages participants to select multiple responses, and the order in which they select those responses is subject to carry-over effects. To address the limitation, we replicated their design but used a fill-in-the-blank task to test participants' natural responses to causal reasoning problems. Participants in our study registered their responses by using a series of dropdown menus to formulate a conclusion that relates *X* and *Z*:

[X/¬X] [causes/enables/prevents] [Z/¬Z]

Participants provided one response to each problem.

Method

Participants. 50 participants were recruited through Amazon Mechanical Turk (28 male, mean age = 34.6). 15 participants reported some formal logic or advanced training in mathematics, and all but 1 of the participants were native English speakers.

Design, procedure, and materials. Each participant was presented with 32 two-premise causal inference problems taken from Wolff and Barbey (2015). The causes and effects in each premise were populated from a set of fictional conditions (e.g., "valmork temperaments", "kandersa moods"). Orthodox and omissive antecedents were created using the phrases "having" and "not having," respectively, and so some participants received the following problem:

> Having valmork temperaments prevents kandersa disease.
> Having kandersa disease prevents rempust fever.

The order in which the participants carried out the 32 problems was randomized, as was the assignment of the contents of the premises. Data, materials, experimental code, and computational modeling code are available at https://osf.io/5yqfx.

Results and simulations

Figure 1 (top panel) shows the data from the experiment. As the figure shows, different problems yielded markedly different patterns of response, e.g., participants generated the response "Not X causes Z" for only one of the 32 problems. For brevity, we omit further analyses of the experimental data in favor of using the dataset to benchmark a series of simulation analyses.

Four separate versions of the algorithm were implemented. The four versions reflected the two strategies for model constructed described above (*random sampling* or *preference sampling*) and the two sorts of response policy (*random selection* or *heuristic selection*). A separate parameter search was conducted for each of the four versions of the algorithm.

Sampling method	Response selection	Best fitting ε value	Best fitting λ value	Goodness of fit (*r*)
Random	Random	0.8	1.0	.65
Random	Heuristic	0.8	1.3	.71
Preferential	Random	0.9	0.9	.71
Preferential	**Heuristic**	**1.0**	**0.8**	**.75**

Table 2. The model-fitting results of simulation analyses conducted for each of the four versions of a model-based causal reasoning algorithm. The version of the algorithm that used preferential sampling and heuristic response generation yielded the best fit to the data.

For each parameter search, the parameters ε and λ varied in 0.1 increments such that the ε ranged from 0.0 to 1.0 and the λ parameter ranged from 0.0 to 3.0, which produced 300 separate parameter configurations. For each parameter configuration, the algorithm was run 100 times on each of the 32 separate causal reasoning problems.

Table 2 compares the overall results of each of the four versions of the algorithm. The table shows that the version of the algorithm that used preferential sampling to construct integrated models as well as a heuristic response strategy performed better than the other three versions of the algorithm. Figure 1 (bottom panel) shows the data generated by the best fitting simulation amongst the four versions of the algorithm.

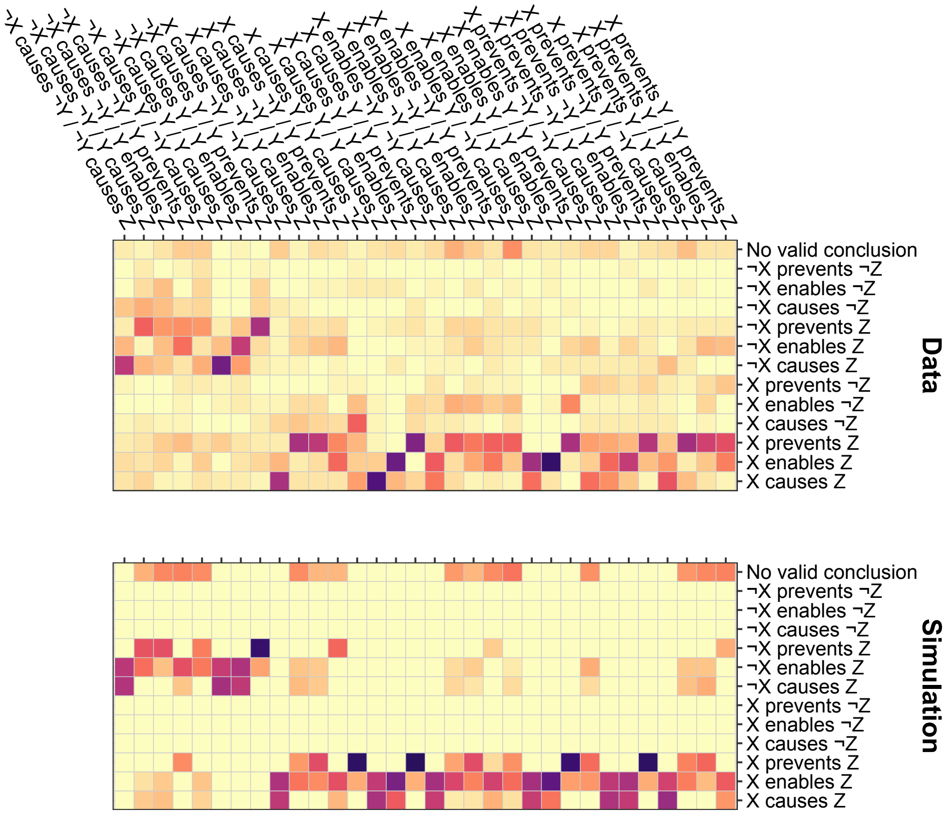


Figure 1. The proportions of participants' responses to 32 different causal reasoning problems (top panel) and from the best fitting simulation from the algorithm that computes causal inferences (bottom panel). The color in each cell indicates the proportion of corresponding conclusions such that the darker the cell, the higher the proportion. Hence, nearly 100% of participants responded "X enables Z" when responding to the problem: "X enables Y / Y enables Z". The version of the algorithm that yielded the best fit implemented a preferential sampling and a heuristic response selection policy.

To assess the necessity of the algorithm's two parameters, we carried out parameter lesioning tests for the version of the algorithm that used preferential sampling and heuristic response selection. Specifically, we ran the algorithm in two lesioned conditions: one in which ε was set to 0, while λ was permitted to vary, and another in which λ was set to 4.0 while ε was permitted to vary. If either condition performed as well as the optimal fit, then it suggests that one of the parameters was redundant. But, neither lesioned condition produced a better fit to the data: the best fitting simulation when λ was permitted to vary yielded a lower goodness-of-fit ($r = .64$) and likewise for the best fitting simulation when ε was permitted to vary ($r = .44$). We conclude that the algorithm that incorporated preferential sampling and heuristic response generation produced the closest match to participants' inferences ($r = .75$).

General discussion

We introduced a novel algorithm for computing causal inferences from sets of causal premises. The algorithm mimics human inference because it is based on a cognitive theory of reasoning, the model theory (Khemlani et al., 2014). It generates causal conclusions by following three procedures: first, the system stochastically constructs mental models from the meanings of causal relations. Second, it combines models from multiple premises using a procedure akin to taking the Cartesian product of a set of possibilities. Third, the algorithm reduces the model and checks it against models of the causal relations specified by the model theory. If an adequate match is found, the system generates the corresponding causal relation as a conclusion.

No prior computational cognitive theory explains how people infer causal relations from sets of causal premises. But, the algorithm can be improved further. As Figure 1 shows, many discrepancies exist between the algorithm's predictions and human participants' tendency to make certain causal inferences. For instance, the algorithm predicts that humans should frequently infer that *X prevents Z* from the following premises: *X causes Y* and *Y causes* $\neg Z$. But people seldom ever make such a response. Perhaps they operate on a different sort of inferential heuristic, or perhaps they deliberate on their initial inferences and consider multiple models consistent with the premises (see, e.g., Khemlani & Johnson-Laird, 2016). The present algorithm can serve as a foundation for causal reasoning systems that take such deliberations into account.

References

Ahn, W. K., & Bailenson, J. (1996). Causal attribution as a search for underlying mechanisms: An explanation of the conjunction fallacy and the discounting principle. *Cognitive Psychology*, 31.

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110.

Beebee, H. (2004). Causing and nothingness. In J. Collins, N. Hall, and L. A. Paul (Eds.), *Causation and Counterfactuals*, Cambridge, MA: The MIT Press.

Bello, P., & Khemlani, S. (2015). A model-based theory of omissive causation. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Bello, P., Wasylyshyn, C., Briggs, G., & Khemlani, S. (2017). Contrasts in reasoning about omissions. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Bernstein, S. (2014). Omissions as possibilities. *Philosophical Studies*, 167.

Byrne, R. (2005). *The rational imagination: How people create alternatives to reality*. Cambridge, MA: The MIT Press.

Cheng, P. W. (1997). From covariation to causation: a causal power theory. *Psychological Review*, 104.

Cummins, D. D., Lubart, T., Alksnis, O., & Rist, R. (1991). Conditional reasoning and causation. *Memory & Cognition*, 19.

Dowe, P. (2001). A counterfactual theory of prevention and'causation'by omission. *Australasian Journal of Philosophy*, 79.

Goldvarg, E. & Johnson-Laird, P. (2001). Naïve causality: A mental model theory of causal meaning and reasoning. *Cognitive Science*, 25.

Goodwin, G.P., & Johnson-Laird, P.N. (2005). Reasoning about relations. *Psychological Review*, *112*.

Hall, N. (2004). Two concepts of causation. *Causation and counterfactuals*.

Hegarty, M. (2004). Mechanical reasoning by mental simulation. *Trends in Cognitive Sciences*, *8*, 280-285.

Johnson-Laird, P.N. (2006). *How we reason*. NY: Oxford University Press.

Johnson-Laird, P.N., & Byrne, R.M.J. (1991). *Deduction*. Hillsdale, NJ: Erlbaum.

Johnson-Laird, P.N., & Khemlani, S. (2017). Mental models and causation. *Oxford Handbook of Causal Reasoning*, 1-42.

Johnson-Laird, P. N., Khemlani, S., & Goodwin, G.P. (2015). Logic, probability, and human reasoning. *Trends in Cognitive Sciences*, 19.

Kahneman, D., & Tversky, A. (1982). The simulation heuristic. In D. Kahneman & A. Tversky (Eds.), Judgment under uncertainty: Heuristics and biases (pp. 201-208). Cambridge, England: Cambridge University Press.

Khemlani, S., Barbey, A., & Johnson-Laird, P.N. (2014). Causal reasoning with mental models. *Frontiers in Human Neuroscience,* 8.

Khemlani, S., Byrne, R. M., & Johnson-Laird, P. N. (2018). Facts and Possibilities: A Model-Based Theory of Sentential Reasoning. *Cognitive Science*, *42*, 1887-1924.

Khemlani, S., & Johnson-Laird, P. N. (2011). The need to explain. *Quarterly Journal of Experimental Psychology*, 64.

Khemlani, S., & Johnson-Laird, P.N. (2013). The processes of inference. *Argument & Computation*, 4, 1-20.

Khemlani, S., & Johnson-Laird, P.N. (2016). How people differ in syllogistic reasoning. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Khemlani, S., & Johnson-Laird, P. N. (2017). Illusions in reasoning. *Minds and Machines*, 27.

Khemlani, S., Lotstein, M., Trafton, J.G., & Johnson-Laird, P.N. (2015). Immediate inferences from quantified assertions. *Quarterly Journal of Experimental Psychology*, 68.

Khemlani, S., Mackiewicz, R., Bucciarelli, M., & Johnson-Laird, P.N. (2013). Kinematic mental simulations in abduction and deduction. *Proceedings of the National Academy of Sciences*, 110.

Khemlani, S., Orenes, I., & Johnson-Laird, P.N. (2012). Negation: a theory of its meaning, representation, and use. *Journal of Cognitive Psychology*, 24.

Khemlani, S., Wasylyshyn, C., Briggs, G., & Bello, P. (2018). Mental models of omissive causation. *Memory & Cognition*, 46.

Lagnado, D. A., & Sloman, S. A. (2006). Time as a guide to cause. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *32*.

Pearl, J. (2009). *Causality*. Cambridge University Press.

Peirce, C.S. (1931-1958). *Collected papers of Charles Sanders Peirce*. 8 vols. C. Hartshorne, P. Weiss, and A. Burks, (Eds.). Cambridge, MA: Harvard University Press.

Rottman, B. M., & Keil, F. C. (2012). Causal structure learning over time: Observations and interventions. *Cognitive psychology*, 64.

Sloman, S. (2005). *Causal models: How people think about the world and its alternatives*. Oxford University Press.

Stephan, S., Willemsen, P., & Gerstenberg, T. (2017). Marbles in Inaction: Counterfactual Simulation and Causation by Omission. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society.* London, UK: Cognitive Science Society.

White, P. A. (2014). Singular clues to causality and their use in human causal judgment. *Cognitive Science*, 38.

Wolff P., Barbey A., Hausknecht M. (2010). For want of a nail: how absences cause events. *Journal of Experimental Psychology: General*, 139.

Wolff, P., & Barbey, A. K. (2015). Causal reasoning with forces. *Frontiers in Human Neuroscience*, *9*, 1.

Memory of relative magnitude judgments informs absolute identification

Adithya Narayan Chandrasekaran
NIT Calicut

Narayanan Srinivasan
University of Allahabad

Nisheeth Srivastava
IIT Kanpur

Abstract

We characterize difficulties with both absolute and relative accounts of magnitude representation in the absolute identification paradigm and present a resolution for these difficulties. We postulate that people store neither long-term internal referents for stimuli nor operate simply using binary comparisons of size between successive stimuli. Rather, they obtain probabilistic judgments of size differences between successive stimuli and encode these for future use, within the course of identification trials. We set up a Bayesian ideal observer model for the absolute identification task using this memory-based representation of magnitude and propose a memory-sampling algorithm for solving it. Simulations suggest that this model captures complex human behavior patterns in absolute identification. Specifically, it reproduces empirically documented crossover effects, practice effects, effects from the use of overlapping stimuli and stimuli with uneven spacing.

Keywords: absolute identification; relative judgment; mental representations; memory; Bayesian learning

Introduction

While it is possible for humans to make fine-grained perceptual judgments about magnitudes, it is not yet clear at what granularity judgments about magnitudes experienced previously are stored. Theoretical opinion currently lies on a spectrum conceptually defined by two strongly divergent positions: one camp assumes that people have direct psychophysical access to the magnitude of entities in the world (how big was this stimulus on the scale I'm interested in?) (Brown, Marley, Donkin, & Heathcote, 2008); the other claims that people store only the relative results of comparative evaluations (which stimulus was bigger?) (Stewart, Brown, & Chater, 2005).

A classic problem for the absolute magnitude camp is that of absolute identification. Across a range of sensory modalities like line lengths, sound frequency, and sound loudness, observers are quicker and more accurate in identifying stimuli at the extremes of the presented stimulus set than those in the middle (Lacouture & Marley, 2004). In addition to identification, it is also possible to ask participants to categorize perceptual stimuli into one of two groups, in which case a similar pattern of results is seen to hold. The same 'bow-tie' seen in identification experiments is also seen in perceptual categorization experiments, with extreme stimuli within the stimuli set categorized more accurately and rapidly (Lacouture & Marley, 2004; Ratcliff & Rouder, 1998). Why should the range of stimuli presented in a set affect observers' responses to individual stimuli, if each stimulus has its own independent internal magnitude representation?

A classic challenge posed to the relative magnitude camp is the distance effect seen in closely related experiments. Distance effects are seen when people are asked to identify which of two presented stimuli is larger (Ratcliff & Rouder, 1998). Participants were more accurate and quicker to respond when pairs of presented stimuli were far apart in actual brightness in a brightness discrimination experiment (Ratcliff & Rouder, 1998). The distance effect in perceptual choice finds an exact counterpart in the distance effect observed in economic experiments, where participants are more inconsistent and late in responding when the values of competing options are close (Dickhaut, Smith, Xin, & Rustichini, 2013). If people are not storing absolute magnitude information, why do they find stimuli farther apart easier to categorize and differentiate?

While explanations for subsets of these phenomena have been previously proposed, the ubiquity of these effects in perception and cognition demands a universal explanation, one equally applicable to both simple perceptual identification and cognitive preference judgment tasks. Sophisticated models of absolute identification place the source of these effects in the process by which observers map their internal representations of perceived stimuli magnitude onto discrete number labels. For instance, Lacouture & Marley showed that treating the magnitude-label mapping problem as an *encoder problem*, to be solved by a feed-forward network, yields mappings for response strengths quadratic in the stimulus order, immediately yielding the bowtie effect (Lacouture & Marley, 1995) when coupled with a DDM (Brown et al., 2008).

Assuming that long-term absolute internal representations of stimuli magnitude are noisy, the efficient encoding hypothesis holds that when confronted with a specific stimulus set, humans will respond to the specific task challenge of mapping stimuli to labels by comparing the presented stimulus to all available internal referents. The strength of the evidence for the mapping is information-theoretically stronger for stimuli corresponding to fewer overlapping internal referents, thus privileging points closer to the extremes, since they will have less interference from stimuli representations from one side of the scale.

A prominent empirical challenge to such accounts comes from the finding that stimuli of the same length are responded to differently when they are members of stimuli sets of different lengths, even within the same subject. If long-term stimulus magnitude representations exist, then they should be indifferent to the impact of adding more stimuli to an existing stimuli set, and the pattern of response should not change for the side of the stimulus order where new stimuli are not added. However, empirical evidence shows that it does (Sims, 2016). One solution to this problem is to adjust the noise levels in the internal stimuli representations 'adaptively' as a function of the set of stimuli to be represented (Sims, 2016). Such solutions, while mathematically feasible, call into question exactly how long-term the internal representations are, if they are to be so responsive to extraneous context.

Adopting a representation of stimuli that stores only local comparisons, it has been argued that observers, once given feedback about the previous trial, and comparing the current stimulus to the immediately previous one, can restrict the range of possible responses by using the previous stimulus as an upper or lower bound for the new one (Stewart et al., 2005). This range restriction naturally proves to be more informative for stimuli closer to the edge of the stimulus set range, making responses to these stimuli more accurate. Thus, a convex relationship between response strength and stimulus order, specific to the presented stimuli set, is obtained.

Prominent challenges to such relative comparison-based accounts include the fact that they do not provide easy explanations for differences in response patterns induced as a function of unequal distances between stimuli in absolute identification tasks. When a large gap was included in the middle of an otherwise linear in log space stimuli range, people find stimuli surrounding this gap easier to identify. However, relative judgment models find it hard to even fit such data without detracting from predictive performance for the other stimuli (Brown, Marley, Dodds, & Heathcote, 2009). The core problem is that the model in question, the relative judgment model (RJM) uses a hard threshold in inter-stimulus distance to determine if a stimulus is larger or smaller than its predecessor, and fits this threshold as a parameter (Stewart et al., 2005). Changes in spacing end up compromising the quality of the model fit.

It is intriguing to note that what is hard to explain using one family of models is easy using the other. Relative judgment models would have no difficulty explaining the effect of multiple stimuli sets on the response pattern, since there are no long-term response strength mappings to expect consistent responses from. Absolute magnitude models would find it straightforward to explain heightened accuracy across large gaps - assuming the same variance for each internal representation, shifts in the mean by adding a gap increases the discriminability of neighboring stimuli, increasing the response strength for the corresponding stimuli.

Finally, both classes of models find it hard to explain practice effects in absolute identification - the fact that participants in these experiments actually get better at the task given practice (Rouder, Morey, Cowan, & Pealtz, 2004). Since neither class of model posits any form of learning mechanism for observers, they fail to explain the actual learning curves seen in real experimental subjects (Dodds, Donkin, Brown, & Heathcote, 2011).

Judgments are formed from memories

The striking complementarity of the strengths and weaknesses of absolute and relative models of absolute identification suggest an opportunity to formulate an intermediate account that bridges this theoretical divide. We make an effort to do so in this paper.

We make three assumptions about the process by which observers perform absolute identification and related tasks.

- First, we assume that the mental representation actually used by people in such tasks is a judgment of *relative* magnitude made using comparison to the immediately preceding stimulus during the experiment.
- Second, we assume that observers learn the stimulus-label mapping via a process well-described as an approximately Bayesian learning algorithm that explicitly samples memory engrams corresponding to the internal representations of stimulus magnitude learned during earlier trials of the experiment.
- Finally, we assume that this memory sampling self-terminates according to an information-gain criterion during each trial, and that the learned distribution of stimuli ranks at the time of termination is what the observer uses to emit an overt label response.

The relative magnitude representation. We use the same relative judgment assumption as Stewart's RJM model (Stewart et al., 2005), that observers calculate a relative magnitude judgment comparing the immediate stimulus to the one immediately preceding it. This probabilistic representation of the pairwise difference between successive stimuli may, in principle, contain more information than a simple binary judgment. For any pair of successive observations $\{x_{t-1}, x_t\}$, we denote this probabilistic container of relative magnitude $p(r|x, o = \{x_{t-1}, x_t\})$, where r takes on the interpretation of magnitude. For all the demonstrations in this paper, we use binary judgments.

Bayesian stimulus-label mapping. Given this assumption about the nature of the long-term internal referent, an observer's goal in absolute identification is to extract a relative magnitude judgment across stimuli in the stimulus set given access to a history of pairwise relative magnitude observations, and to do so using their own history of stimulus exposure within the task. We model the stimulus-label mapping process in the absolute identification task as Bayesian marginalization over relative magnitude judgments seen in pairwise comparisons (Srivastava, Vul, & Schrater, 2014). The mathematical machinery of sequential Bayesian updating allows us to formalize this learning process sequentially on a trial-by-trial, instead of treating the stimulus-label mapping and experimental responding as separate events as is classically done.

The relative magnitude of each stimulus, as we describe above, takes on a probabilistic interpretation formally expressed as $p(r|x, o)$, where r is the relative magnitude judgment, x is the currently visible stimulus, and $o = \{x_{t-1}, x_t\}$ is the relevant comparison *observation*. The ideal Bayesian observer learns $p(r|x, o)$ by combining comparison information from all previously observed comparisons. Thus, this quantity is obtained by marginalizing over the set of previously seen unique observations in memory $\mathcal{C} = \mathcal{P}(\mathcal{X}), s.t. \forall c \in$

$\mathcal{C}, |c| = 2$ which we denote the memorized *comparisons*. Then,

$$D(x) = p(r|x,o) = \frac{\sum_c^{\mathcal{C}} p(r|x,c)p(x|c)p(c|o)}{\sum_c^{\mathcal{C}} p(x|c)p(c|o)}, \qquad (1)$$

where it is understood that the *comparison* probability $p(c|o) = p(c|\{o_1, o_2, \cdots, o_{t-1}\})$ is a distribution on the set of all comparisons available from observation history. Here, $p(r|x,c)$ encodes the probability that the item x was found to be larger in the comparison c, $p(x|c)$ encodes the probability that the item x was present in the context c and $p(c)$ encodes the frequency with which the observer encounters these comparisons during the experiment. This frequency is updated via recursive Bayesian estimation,

$$p(c^{(t)}|o^{(1:t)}) = \frac{p(o^{(t)}|c)p(c|o^{(1:t-1)})}{\sum_c^{\mathcal{C}} p(o^{(t)}|c)p(c|o^{(1:t-1)})}. \qquad (2)$$

This completes the computational description of the task an ideal Bayesian observer would perform in service of absolute identification, given access to local relative magnitude judgments. The practical approximation arises when we explicitly model the act of accessing previous relative magnitude judgments as memory sampling.

Self-terminating memory sampling. Evidence accumulation influences the shape of the distribution $p(c|o)$ via memory sampling. We model the process of memory recall as the activation of a subset $\mathcal{Q}$ of decision-relevant memory engrams. Using this notation, a general memory accumulation model could be expressed as,

$$p(c) = \sum_{q \in \mathcal{Q}} p(c|q)p(q), \qquad (3)$$

where $c \in \mathcal{C}$ are stimuli comparisons available in memory and $q \in \mathcal{Q}$ are memory engrams corresponding to past relative magnitude judgments. Here, the probability distribution $p(q)$ - which we call the *memory prior* - encodes the likelihood of recalling the memory of experience q, while the distribution $p(c|q)$ encodes the knowledge of having seen c and its corresponding relative magnitude judgment stored in the memory engram q. For simplicity, we assume a trivial bijective mapping between c and q - each memory engram is assumed to be associated with a unique stimuli pair.

This memory-sampling variant of $p(c|o)$ plugs directly as the prior in the Bayesian comparison probability update for $p(c|o)$ in Equation 2, which then itself plugs into the two computations in Equations 1 and 2 that define the ideal observer model. This replacement is facilitated by one additional assumption: that the comparison-specific memories recalled are episodic, and therefore convey all comparison-relevant information once the comparison episode itself has been activated in memory[1].

Finally, we formalize our information-theoretic criterion for terminating memory sampling and emitting an identification response. We assume that observers continue to sample memory engrams until the rate at which these provide new information subsides below a threshold. Additional information gained by adding an additional engram q_n to the existing set can be expressed as,

$$IG(q_n) = \sum_i p(c_i|q_{1:n-1}) \log \frac{p(c_i|q_{1:n-1})}{p(c_i|q_{1:n})}, \qquad (4)$$

so our sampling termination rule is,

$$\arg\min_n IG(q_n) < T, \qquad (5)$$

where T is the termination threshold, potentially informed by exogenous influences.

Stimulus rank decoding. At each time step t the model uses the differential internal representation between the current and one-back stimulus $(D_t - D_{t-1})$, and the previous rank obtained from post-trial feedback to estimate the current stimulus' rank according to the formula:

$$RANK_t = RANK_{t-1} + \left(\frac{D_t - D_{t-1}}{\alpha_t}\right) \qquad (6)$$

The parameter α in turn is updated at each time step t as:

$$\alpha_t = \frac{1}{t}\sum_t \left[\frac{D_t - D_{t-1}}{RANK_t - RANK_{t-1}}\right] \qquad (7)$$

The observer's choice is determined from the relative magnitude judgments across all x available at the time memory sampling is terminated. We count instances where the observer's decision variable predicts the correct rank of the stimulus introduced on individual trials as accurate responses. Samples to termination are directly interpreted as linearly scaled response times. Notice that the parameter α controlling the rank-magnitude mapping relies entirely on local one-back comparisons between magnitude judgments D, as in the RJM (Stewart et al., 2005).

Simulation Results

Modelling our *in silico* experiment design after the design reported in (Lacouture & Marley, 2004), we showed 20 instances of the model 40 copies each of N stimuli, asking them to assign number labels $1 \cdots N$ to them. On each trial, agents updated their estimates for $p(r|x_t, o = \{x_t, x_{t-1}\})$ following the model described above. Since we assumed equal spacing on a log scale for stimuli as in the original experiment, we kept the relative magnitude judgments as 1 for simplicity, and used a threshold value $T = 10e-7$ across all our experiments unless specified otherwise.

Our model reproduces the absolute identification results of (Lacouture & Marley, 2004), which are the baseline benchmark for absolute identification models (Brown & Heathcote, 2008). Accuracy exhibits a convex relationship with stimulus

[1] This assumption simplifies our analysis by ignoring the memory dependence of our other intermediate probability terms. While it is likely that such dependence exists, its effects will work in the same direction as the basic results of our approach, since it would further impoverish the preference representation we are already imposing sampling constraints on.

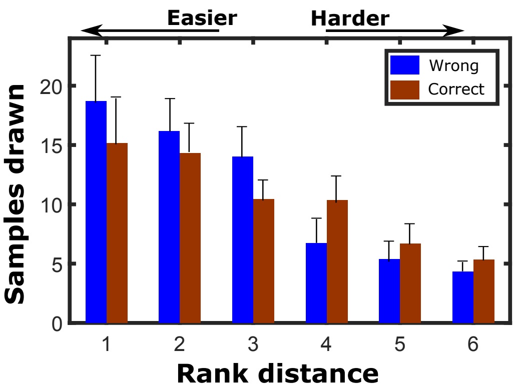


Figure 1: Replication of the crossover effect in perceptual choice.The x-axis plots the rank distance between compared stimuli on a given trial, and the y-axis plots the average number of samples drawn before responding during 20 model runs. Error bars represent ±1 s.e.m. across these model runs.

order, and the response time distribution is concave, matching the profiles observed by (Lacouture & Marley, 2004). We do not explicitly demonstrate these results in the interest of succinctness, focusing on demonstrating more complex patterns of behavior.

Then, we show how it can replicate a harder pattern of behavior - the crossover effect in RT (Luce, Nosofsky, Green, & Smith, 1982; Brown & Heathcote, 2008).

Reproducing the crossover effect

The crossover effect describes a complicated pattern of behavior typically seen in perceptual choice experiments. When choice is easy and speed is emphasized, incorrect responses are quicker than correct responses; when choices are harder and accuracy is emphasized, the opposite is true (Brown & Heathcote, 2008). This pattern of response time (RT) behavior has proved very challenging for several models of choice RT to fit, and is a challenging benchmark for models in this field.

Perceptual choice fits into our framework without affecting the formalism in the slightest. The only difference is that the observations o now represent two stimuli seen together instead of sequentially. All the other interpretations remain identical to those in the identification setting. We conducted *in silico* experiments using the same simulation setup as above. As Figure 1 illustrates, our model displays a crossover effect even ignoring the effect of the speed-accuracy tradeoff.

Further, our model offers a straightforward parameter-free explanation for the crossover effect. Simple choices correspond to situations where most samples in memory point in the same direction for a particular stimulus. In such cases, the only way the model could fail to produce the correct response is if the sampling was terminated prematurely. Thus, incorrect responses for simple choices have to be fast. Given sufficient time for integration, it would be impossible for the model to be incorrect. Hard choices correspond to situations where both options have memory samples supporting their case for being bigger. In such cases, the model is biased towards terminating when the marginal information gain is low. Thus, the model will fail to terminate when memory sampling fails to resolve to a modal response, which is more likely when the sampling has failed to discover the true mode of the relative magnitude judgment distribution, resulting in bigger response times for errors.

Reproducing practice effects

By varying the number of history samples, i.e. the samples that the model is exposed to before the start of the trials, our model can reproduce the differential conditions observed in experiments documenting practice effects in absolute identification (Dodds et al., 2011). We ran 5 iterations of the model for practice/no practice conditions with number of stimuli $N = 6$. For the no practice case, the model was exposed to 30 history samples, whereas in the practice case it was exposed to 300 history samples before we started taking the model's predictions into account. The information threshold was kept at $T = 10E-4$.The results of the simulation are shown in Figure 2 alongside data from (Dodds et al., 2011). A clear qualitative reproduction of the pattern of results seen in the experiment is seen - accuracy for end-points starts out high, while responses for stimuli in between start out with greater error, and then improve. The explanation is intuitive: fewer unique samples are needed to clearly differentiate the rank order of endpoint stimuli.

Reproducing overlapping stimuli effects

The overlapping stimuli effect in which the same stimulus elicits different responses when presented as part of different stimulus sets poses a challenge to absolute accounts of magnitude representation. We ran 30 iterations of the model, for each of the cases with number of stimuli $N = 5$, $N = 8$, and $N = 11$. To work around the large compute times necessitated by the combinatorial explosion in the number of contexts to be sampled with increase in N, the empirical data presented in (Sims, 2016) for the cases $N = 13$, $N = 20$, and $N = 30$ were down sampled to $N = 5$, $N = 8$, and $N = 11$ respectively. The down-sampling was done by taking every 3rd data point and extrapolating the last point, if necessary. We observe a strong qualitative and quantitative reproduction of the empirical effect with a single parameter fit across all conditions in the experiment.

Reproducing the uneven spacing effect

When a large central gap is introduced into the stimuli set, the accuracy profile significantly deviates from the bowtie curve, with the stimuli near the gap having higher accuracy compared to the ones away from the gap. The uneven spacing effect presents a major challenge to relative accounts of magnitude representation, including ours. To capture this effect, our model requires an additional augmentation - we assume

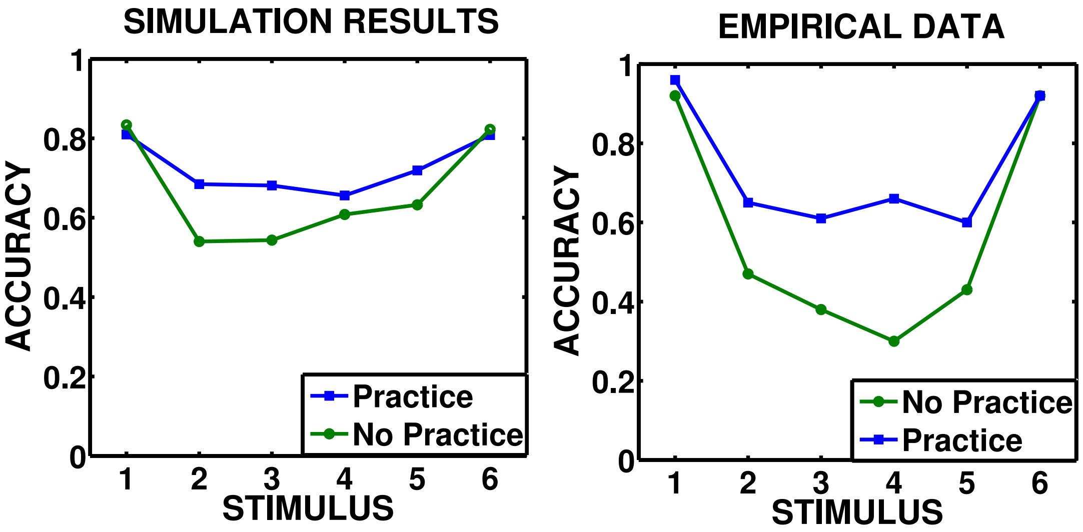


Figure 2: Demonstrating practice effects in absolute identification. *(Left)* Simulation results run by varying the number of history samples to which the model is initially exposed. *(Right)* Empirical data re-plotted from data in article (Dodds et al., 2011).

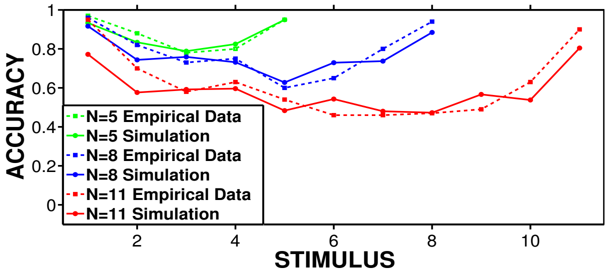


Figure 3: Reproducing the overlapping stimuli effect. The x-axis plots the stimulus length and the y-axis plots the accuracy of the model. Empirical data re-plotted from (Sims, 2016)

that the observer first seeks to identify when a stimulus from one side of the gap is presented versus the other, and then tries to identify its rank.

Rank decoding across gaps. Our model's estimate of α as a simple running average makes sense when it expects perceptually even spacing between successive stimuli. Where spacing is uneven, ranks estimated using such an estimate would be faulty. To accommodate the effect of uneven spacing, after each trial, the decoder in our model calculates and stores the ordinal difference between the perceived rank, as estimated using α, and the real rank, obtained from feedback post trial. This rank difference is denoted by RD and is updated at every trial involving a jump across the gap.

$$RD_t = (PerceivedRank)_t - (RealRank)_t \qquad (8)$$

Rank prediction occurs as follows here

$$RANK_t = RANK_{t-1} + \left(\frac{D_t - D_{t-1}}{\alpha}\right) - RD_t, \qquad (9)$$

such that the evenly spaced stimulus set decoding (as specified in Equation 6) arises as a special case of the unevenly spaced stimuli. If the gap tracks the constant inter-stimulus interval, RD goes to 0 in Equation 9, yielding Equation 6.

We ran 30 iterations of the model with number of stimuli N=10 with a large central gap, 9 times the size of the even spacing gap, introduced between stimuli 5 and 6. The threshold parameter was held at the same value as in the other demonstrations. The model's results (Figure 4) match the 'w' shaped accuracy profile observed in the empirical data.

Discussion

In this paper, we have presented a model of absolute identification based on three basic principles: one, that observers store 1-back relative magnitude judgments in memory; two, that observers solve the computational problem implicit in absolute identification (stimulus-rank mapping) using an approximately Bayesian calculation that can be stylized as sampling engrams from memory; three, that this memory sampling procedure terminates using an information-gain criterion.

Our model's capacity to identify absolute stimuli arises from differences in the informativeness of memory samples corresponding to various stimuli. Because the evidence from comparisons involving extreme stimuli consistently points the same way, the marginal information gain from sampling

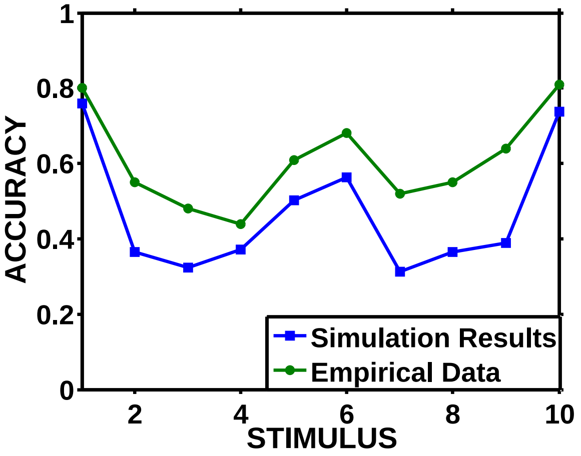


Figure 4: Reproducing the uneven spacing effect. The x-axis plots the stimulus length and the y-axis plots the accuracy of the model. Empirical data replotted from (Brown et al., 2009)

saturates rapidly and the model terminates memory retrieval sooner, leading to faster and accurate responses. On the other hand, for stimuli closer to the middle, samples will be split between comparisons where the stimulus is larger and ones where it is smaller, resulting in greater decision variable volatility, and hence, more sampling. This interaction then manifests summarily as slower and noisier responses.

The representational flexibility provided by our encoding stimulus-rank mapping information in memory, in conjunction with the fact that we model the process by which the representation is actually learned trial-by-trial from stimulus observations, allows our model to reproduce practice effects in absolute identification (Dodds et al., 2011), as well as reproduce both the shift in response patterns as a function of stimulus set (Rouder et al., 2004) and the heightened response to unequal spacing (Brown et al., 2009) without committing to long-term storage of magnitude estimates for arbitrary stimuli, and using only two explicit free parameters. Our account also predicts that the uneven spacing effect should also propagate to the RT distribution - an easily testable prediction.

In addition to these direct results, sequentially modelling the mapping process, in conjunction with the use of an information-based stopping criterion, also sheds new light on the relationship between the psychophysical bowtie effect (Lacouture & Marley, 2004) and the economic distance effect (Dickhaut et al., 2013). Extreme choice valence (distance in utility) appears to be correlated with lower error rate, response times and interestingly, levels of neuronal activation as measured by fMRI (Dickhaut et al., 2013). According to our model, constructing a decision variable using conflicting evidence requires more samples to breach the information-based threshold, resulting in greater effort, which is correlated with higher RT and brain activation for both perceptual and economic choices with greater mutual confusability, as determined by their history of pairwise comparisons.

In summary, the model we have proposed appears to have robust empirical fits to challenging data within the absolute identification literature, and interesting theoretical connections with other strands in the literature on choice process modeling.

Acknowledgments

NiS acknowledges financial support from the Research I Foundation.

References

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: linear ballistic accumulation. *Cognitive psychology*, *57*(3), 153–178.

Brown, S. D., Marley, A., Dodds, P., & Heathcote, A. (2009). Purely relative models cannot provide a general account of absolute identification. *Psychonomic Bulletin & Review*, *16*(3), 583–593.

Brown, S. D., Marley, A., Donkin, C., & Heathcote, A. (2008). An integrated model of choices and response times in absolute identification. *Psychological review*, *115*(2), 396.

Dickhaut, J., Smith, V., Xin, B., & Rustichini, A. (2013). Human economic choice as costly information processing. *Journal of Economic Behavior & Organization*, *94*, 206–221.

Dodds, P., Donkin, C., Brown, S. D., & Heathcote, A. (2011). Increasing capacity: Practice effects in absolute identification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *37*(2), 477.

Lacouture, Y., & Marley, A. (1995). A mapping model of bow effects in absolute identification. *Journal of Mathematical Psychology*, *39*(4), 383–395.

Lacouture, Y., & Marley, A. (2004). Choice and response time processes in the identification and categorization of unidimensional stimuli. *Perception & Psychophysics*, *66*(7), 1206–1226.

Luce, R. D., Nosofsky, R. M., Green, D. M., & Smith, A. F. (1982). The bow and sequential effects in absolute identification. *Attention, Perception, & Psychophysics*, *32*(5), 397–408.

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356.

Rouder, J. N., Morey, R. D., Cowan, N., & Pealtz, M. (2004). Learning in a unidimensional absolute identification task. *Psychonomic Bulletin & Review*, *11*(5), 938–944.

Sims, C. R. (2016). Rate–distortion theory and human perception. *Cognition*, *152*, 181–198.

Srivastava, N., Vul, E., & Schrater, P. R. (2014). Magnitude-sensitive preference formation. In *Advances in neural information processing systems* (pp. 1080–1088).

Stewart, N., Brown, G. D., & Chater, N. (2005). Absolute identification by relative judgment. *Psychological review*, *112*(4), 881.

Understanding the Learning Effect of Approximate Arithmetic Training: What was Actually Learned?

Sizhu Cheng[1] **(scheng72@stanford.edu)**
Institute of Computational Mathematical Engineering, Stanford University, Stanford, CA 94305 USA

Arianna Yuan[1] **(xfyuan@stanford.edu)**
Department of Psychology, Department of Computer Science, Stanford University, Stanford, CA 94305 USA

Abstract

Previous studies have documented the learning effect of approximate arithmetic training on symbolic arithmetic skills in both preschoolers and adults. Particularly, Park and Brannon (2013, 2014) have trained participants to add two non-symbolic quantities (dot arrays) and showed that such training improved the participants' symbolic arithmetic skills. They argued that this finding suggested that training subjects to mentally manipulate non-symbolic quantities and practice nonverbal addition would result in enhanced symbolic arithmetic skills. However, we would like to propose an alternative explanation to account for their findings: instead of mentally manipulating visual dot arrays, participants might simply estimate the numerosities represented by the dot arrays and practice exact symbolic arithmetic during the seemingly non-symbolic arithmetic training. In that case, it would also lead to a better performance in subsequent symbolic arithmetic test. To verify our hypothesis, we implemented a neural network model to simulate their experiments. Our simulations confirmed that our explanation was sufficient to reproduce their psychological findings. The current work invites re-interpretation of the benefit of approximate arithmetic training and pushed us to think what cognitive component was improved exactly in approximate arithmetic training and what the relationship between approximate number processing and symbolic number processing really is.

Keywords: neural network model; approximate arithmetic; symbolic arithmetic; cognitive training

Introduction

Many researchers believe that infants exhibit a number sense without any knowledge of mathematical symbols (Feigenson, Dehaene, & Spelke, 2004). This non-symbolic representation of numbers, known as the Approximate Number System (ANS), is believed to be widely present in different species (Feigenson et al., 2004). Previous researchers have documented positive correlations between non-symbolic number processing and symbolic mathematical ability (Halberda, Mazzocco, & Feigenson, 2008; Libertus, Feigenson, & Halberda, 2011).

Particularly, Park and Brannon (2014) investigated the effect of approximate arithmetic training on symbolic arithmetic performance. Approximate arithmetic refers to a process of applying numerical manipulation, such as addition and subtraction, to non-symbolic quantities without counting (Park & Brannon, 2013). It turns out that such nonverbal arithmetic training did improve the exact symbolic arithmetic performance for adults and this result cannot be explained by just better approximate number comparison or a boost of short-term memory (Park & Brannon, 2014). Similar results were found in a recent study by Szkudlarek and Brannon (2018) in which participants were preschoolers.

In Park and Brannon (2013, 2014), participants were first presented with two images of dot array. They were asked to mentally add the two numerical quantities represented by the two dot arrays. Participants were then presented either (1) with a third image of dots and asked to report whether the numerical quantity represented in the third image is more than the sum of the numerical quantities in the first two images (More-or-Less comparison) or (2) with two images of dots and asked to select the one that matched the sum of the numerical quantities in the first two images (Same-or-Different comparison). Note that the participants were not given enough time to count the number of dots and they were not required to report the sum of the numerical quantities in the first two images. Park and Brannon (2013, 2014) found that training participants to perform these tasks lead to significant improvement in symbolic addition and subtraction. The authors took the findings as evidence for a causal link between non-symbolic number processing skills and symbolic number process skills. Many other researchers shared this view as well (for review, see Szkudlarek and Brannon (2017)). Particularly, they argued that nonverbal addition and subtraction was trained in such approximate arithmetic training, which led to improvement in symbolic addition and subtraction. However, we would like to propose an alternative explanation for such learning transfer: participants may simply estimate the numerosities of the first two dot arrays, perform symbolic arithmetic on the extracted numerosities, and compare the sum with the numerosity of the third dot array. In other words, the participants may practice symbolic arithmetic in the seemingly non-symbolic, arithmetic training (Julie & Silke, 2012; Gobel, Watson, Lervag, & Hulme, 2014; Mazzocco, Feigenson, & Halberda, 2011a, 2011b; Kolkman, Kroesbergen, & Leseman, 2013). Under that assumption, it would also lead to a better performance in later symbolic arithmetic test.

In this paper, we explore this alternative explanation by implementing this idea in a deep learning model. In our modeling framework, we simulated a similar approximate arithmetic training experiment as in Park and Brannon (2013, 2014), and evaluated our trained model on symbolic arithmetic problems. For simplicity, we only report our simulations on addition tasks. We compare the model

[1] equal contribution

performance on symbolic addition tasks before and after approximate arithmetic training and showed that the training resulted in significant learning outcome. Our simulations confirmed our hypothesis that an alternative explanation for the transfer learning effect of approximate arithmetic training is plausible, and future work is needed to understand what cognitive component is actually being trained in approximate arithmetic training and what the relationship between approximate number processing and symbolic number processing might be.

Method

Data For non-symbolic number stimuli, we randomly generated 10000 training images of dot arrays (see **Figure 1**, top). The number of dots per image ranged from 0 to 9, and each numerosity class contained 1000 images. We also generated a separate test dataset of 2000 dot array images. For symbolic numbers, we used the training digit images and the test digit images from the MNIST database (LeCun & Cortes, 2010), a dataset of handwritten digits with labels. In both cases, the training dataset and the evaluation dataset had no overlapping images.

For the comparison tasks, we train the model to compare numbers that ranged from 0 to 9. For More-or-Less comparison, the two comparands always differ by one, and the base rates of responding "More" and responding "Less" are the same (except for 0 and 9). In Same-or-Different comparison, every number n is compared with itself, and also with either $n+1$ or $n-1$, so that the base rates of responding "Same" and responding "Different" are the same.

Model Architecture

Approximate Arithmetic

We proposed a neural network architecture to perform the non-symbolic arithmetic task. As mentioned in the **Introduction**, we proposed an alternative account for how participants actually solved the approximate arithmetic problems in Park and Brannon (2013, 2014) experiments. Consistent with our hypothesis, in our model, the approximate arithmetic task was decomposed into a "recognition" part, an "addition" part and a "comparison" part. In our previous work, we have shown that decomposing numeric processing tasks into sub-tasks and building several neural networks that collectively work on the tasks could explain learning transfers in a situated math game environment (Yuan & McClelland, 2019). Inspired by those results, in the current paper our model also consists of several neural networks (**Figure 1**). First, we built an Image Classifier Network to extract the numerosity of the dot arrays. The Image Classifier Network learned to output a one-hot vector which represented the number of dots in the given input image. We built an Addition Network to add two numbers in the form of one-hot vectors/probability vectors to obtain the sum (also represented as a vector). Finally, we had a Comparison Network that learned to compare two numbers. During the approximate arithmetic training, the Addition Network is combined with the Comparison Network to form a so-called Combined Network that could perform the Same-or-Different/More-or-Less comparison. We next describe these networks in detail (**Figure 1**).

Image Classifier Network The Image Classifier Network was trained on dot array images to output a one-hot vector which represented the numerosities (from 0 to 9) of the dot array in the image. The Image Classifier Network processed the input images with a convolutional neural network (CNN). It contained four blocks of (a convolution layer with 3×3 filters and 20 output channels, a ReLU activation layer, a 2×2 max pooling layer), followed by two fully connected layers of size 120 and 84 with ReLU activation function. The activation of the last fully connect layers were then used to predict the one-hot vectors of numerosities. Here the ReLU non-linear activation function is used to restrict the magnitude output to be non-negative, i.e., $\mathrm{ReLU}(x)=\max(0,x)$.

Addition Network The Addition Network was a fully connected neural network of two hidden layers of size 81 (with ReLU activation function), which was trained to output a correct response for the addition task given two addends. Since we are only interested in modeling the relationship between non-symbolic number processing and symbolic number processing, but not multi-column addition, we restricted our scope to single digit processing. Therefore, we only considered numbers that ranged from 0 to 9, and the number pairs whose sum fell into the interval between 0 and 9.

Same-or-Different (SD) Comparison To perform the Same-or-Different comparison, we built a model that learned to differentiate very similar numerosities. The model learned to decide whether two input numbers were the same or different. Our SD Comparison Network is a fully connected layer of two fully connected layers of size 81 (with ReLU activation function) and its output layer contained two nodes, indicating the "Same" and the "Different" responses.

More-or-Less (ML) Comparison We also simulated the More-of-Less comparison. Instead of deciding whether two numerosities were the same or not, the More or Less (ML) Comparison Network needed to decide whether the first image had more dots than the second image or less dots. We trained a different comparison network. The architecture of the ML Comparison Network is exactly the same as the SD Comparison Network, except that the two output nodes now represented the "More" and the "Less" response, rather than the "Same" and the "Different" response.

Combined Network During the approximate arithmetic training, three dot array images were first processed by the Image Classifier Network to obtain three 10-dimensional probability vectors. The vectors of the first two images were then fed to the Addition network. The output of the Addition Network was again a probability vector representing the distribution over the possible sums. This output was concatenated with a third vector and was fed into the comparison network. This was to simulate the experimental procedure in Park and Brannon (2013, 2014). Regarding the output of the Combined Network, in Same-or-Different (SD) Comparison, the output layer had two nodes, indicating whether the number of all dots in the first two images was equal to the number of dots in the third image. In the More-or-Less (ML) Comparison, the output layer also had two nodes, indicating whether the number of all dots in the first two images was greater than the number of dots in the third image. The architect of the model and the information flow during the training process is illustrated in **Figure 1**.

Symbolic Arithmetic Network

When people solve symbolic arithmetic problems, it is reasonable to assume that humans first recognize the digits and then perform the actual addition or subtraction operation. To simulate this process, we decomposed the symbolic addition task into a "recognition" part and an "addition" part. We use the same Image Classifier Network and the same Addition Network as used in the approximate arithmetic training simulation to solve the symbolic arithmetic problems. For symbolic arithmetic task, we trained our Image Classifier Network on the digit images to output a 10-dimensional one-hot vector which represented the digit classes from 0 to 9.

Training Procedure

In this task, we first pre-trained our Image Classifier Network on both digit images and dot array images so that it achieved near-perfect accuracy. Also, it is reasonable to assume that participants in Park and Brannon (2013, 2014) already had some ability to add single-digit numbers. Therefore, we pre-trained our Addition Network to reach a decent level of accuracy. Particularly, the addition network for symbolic arithmetic is pre-trained for 60 epochs to reach ∼60% accuracy. We deliberately leave our Addition Network some room for improvement so that we could explore the role of non-symbolic arithmetic training and avoided a ceiling effect. Similarly, we also assumed that the participants were already equipped with some knowledge about comparison before they underwent the approximate arithmetic training, so we also pre-trained our SD and ML comparison network 7 epochs to reach ∼90% accuracy.

When simulating the comparison tasks, the probability vectors over the possible numerosities are fed to the Comparison Network. Since we pre-trained the Image Classifier to near-perfect performance, the probability vectors approximated 10-dimensional one-hot vectors, where the node standing for the correct label had activation of 1.0 and the rest of the nodes had activation of 0.0.

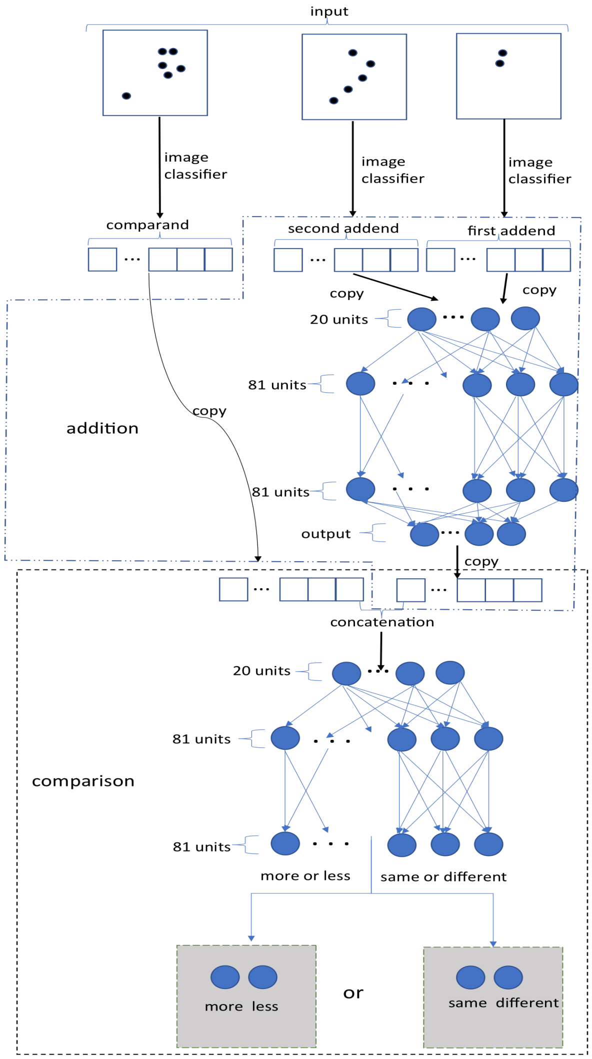


Figure 1: Model Architecture

In the joint-training phase, the Comparison Network is joint trained with the Addition Network. The first comparand was the probability vector obtained from the output layer of the downstream Addition Network, and the second comparand (the referent) was the probability vector obtained from the output layer of the downstream Image Classifier Network when applied to the image of the third dot array. The Combined Network (the Addition Network + the Comparison Network) were trained for 200 epochs, during which time period the network only had access to the "Same/Different" or "More/Less" labels. The model was tested on exact symbolic addition both before and after non-symbolic arithmetic training. The overall experiments were repeated for 10 runs with different random seeds. For both SD comparison and ML comparison, we used cross entropy loss and we ran backpropagation to train the model (Rumelhart, Hinton, Williams, et al., 1988). The learning rate for both SD and ML experiments were the same, as shown in **Table 1**. For evaluation, the accuracy was calculated as the percentage of correct examples among all predictions.

Results

For the sake of clarity, the learning curves of the Addition Network during the single-task training and the joint-training (when the Combined network was trained) are colored differently in the figures in this paper. Moreover, the horizontal axis indicates the cumulative epochs, i.e., the number of epochs that one network has been trained. In other words, since the Addition Network has been trained for 60 epochs before it was recruited in Combined Network for joint-training, then the epoch index of its joint training starts from 60 rather than 0. The arrows and the associated texts in the figures indicate the time point when the training condition changed.

Table 1: Learning Rates in different tasks

Model	Learning Rate
Pre-trained Addition	0.001
Pre-trained Comparison (SD/ML)	0.01
“Combined” Addition	0.0012
“Combined” Comparison	0.0001

Approximate Arithmetic Training with Same-or-Different Comparison

The learning curves of the networks, averaged from 10 runs, are shown in **Figure 2** and **Figure 3**. The error bars indicate standard deviations that are calculated from 10 runs and are plotted every 5 epochs.

It can be seen from **Figure 2** that the addition test loss reached the minimum around epoch 200, i.e. when the combined network was trained for ~140 epochs. The prediction accuracy and the cross entropy loss were calculated both before the Combined Network was trained (i.e. pre-approximate arithmetic training) and after the Combined Network was trained (i.e. post-approximate arithmetic training). We ran a paired t-test and found that there was a significant improvement in test accuracy of the symbolic arithmetic problems after the approximate arithmetic training, $t = 18.427, p = 0.000$. Both pre-training and post-training test accuracies are shown in **Table 2**.

The learning curves of the Combined Network during approximate arithmetic training are shown in **Figure 4**, with the loss and the accuracy of the Addition Network and the whole Combined Network. We could see that as the training proceeded, both the test accuracies of the Addition Network and the whole Combined Network increased (**Figure 4**, bottom).

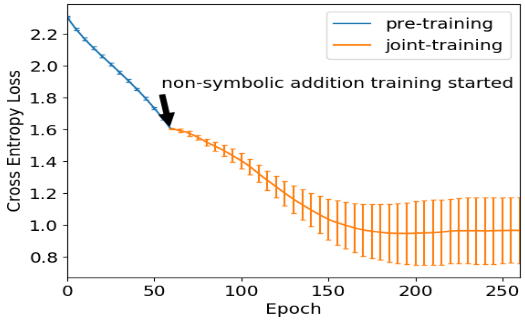


Figure 2: Test loss of the “Addition Network” in Same-or-Different comparison.

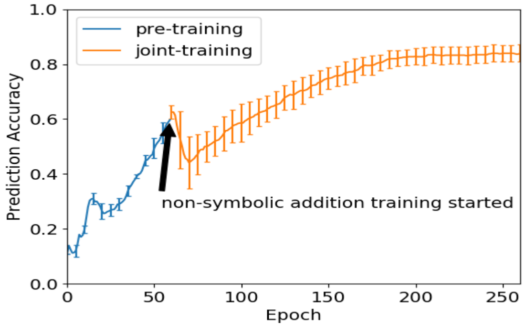


Figure 3: Test accuracy of the “Addition Network” in Same-or-Different comparison.

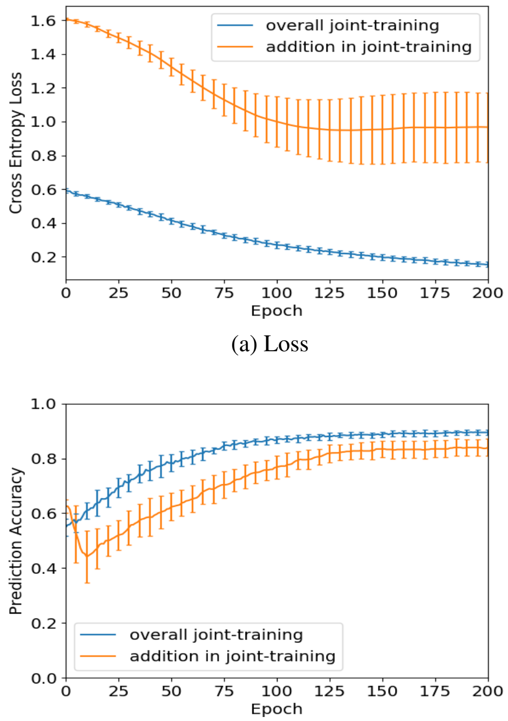
Approximate Arithmetic Training with More-or-Less Comparison

Similarly, in the More-or-Less Comparison case, the overall experiments were repeated 10 times with different random seeds. The learning curves of the networks, averaged from 10 runs, are shown in **Figure 5** and **Figure 6**. **Figure 7** shows the learning curves of the Addition Network and the overall Combined Network during the approximate arithmetic training.

From **Figure 5** we can see that the addition test loss reaches the minimum around epoch 260, i.e. when the combined network was trained for ~200 epochs. We also ran a paired t-test and found that there was a significant improvement in test accuracy of the symbolic arithmetic problems after the approximate arithmetic training, $t = 17.660, p = 0.000$. Both pre-training and post-training test accuracies are shown in **Table 2**. Though the learning curves demonstrated more variance across different runs in the More-or-Less comparison than in the Same-or-Different comparison, the model's addition ability did improve in both cases.

Table 2: Symbolic addition performance pre- and post-approximate arithmetic training

Approximate Arithmetic Training	Addition Loss: Pre-training		Addition Loss: Post-training		Pre- vs. Post- *t*-statistics	Pre- vs. Post- *p*-value	Addition Accuracy: Pre-training		Addition Accuracy: Post-training		Pre- vs. Post- *t*-statistics	Pre- vs. Post- *p*-value
	mean	std	mean	std			mean	std	mean	std		
Same-or-Different Comparison	1.612	0.005	0.967	0.206	-9.983	0.000	0.597	0.030	0.837	0.033	18.427	0.000
More-or-Less comparison	1.613	0.007	0.962	0.212	-9.637	0.000	0.607	0.022	0.876	0.038	17.660	0.000



(a) Loss

(b) Accuracy

Figure 4: The test losses and the test accuracies of the Addition Network (orange) and the whole Combined Network (blue) in Same-or-Different comparison.

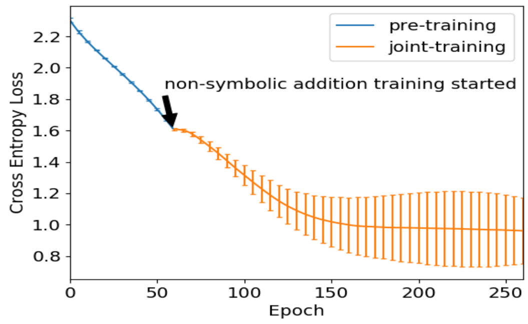


Figure 5: Test loss of the Addition Network in More-or-Less comparison.

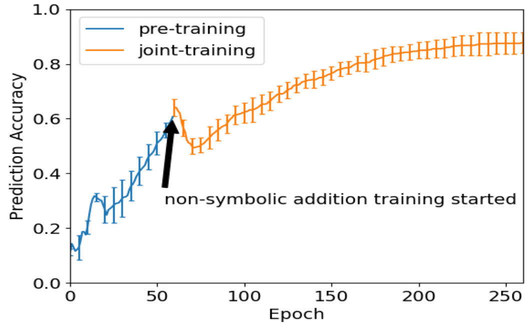
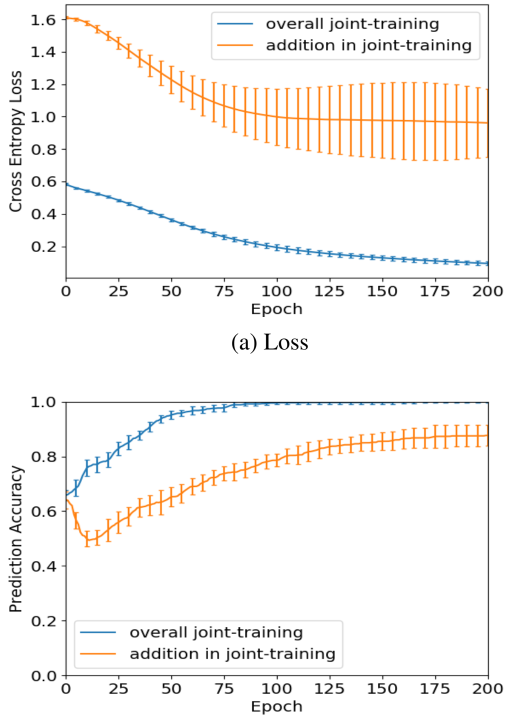


Figure 6: Test accuracy of the Addition Network in More-or-Less comparison.

Discussion

Our model demonstrated improved addition accuracies in both the Same-of-Different comparison and the More-or-Less comparison, which is qualitatively aligned with the psychological findings in Park and Brannon (2013, 2014). This confirmed our hypothesis that if participants were doing exact symbolic addition rather than nonverbal addition during the approximate arithmetic training, their symbolic arithmetic skill would still be improved. This alternative explanation forces us to question the original interpretation of the findings in Park and Brannon (2013, 2014). For instance, how did participants mentally added the two dot arrays exactly? Did they combine two dot arrays and compare it with the third dot array, or extract the numerosities first and perform exact symbolic addition? Also, what was being compared in the non-symbolic arithmetic training? Were the non-symbolic combination of dots being compared or the symbolic sums? All these questions call for future investigation.

One of the limitations in the current study is that number distribution in our simulations did not match the one in Park and Brannon (2013, 2014). They used arithmetic problems that involved multi-digit numbers, but in our current study we only limited our scope on single digit addition. It is reasonable to believe that our conclusions should still hold when extended to multi-digit addition, because multi-digit addition does involve single digit addition. Future work is needed to confirm this intuition. Another limitation is that we



(a) Loss

(b) Accuracy

Figure 7: The test losses and the test accuracies of the Addition Network (orange) and the whole Combined Network (blue) in More-or-Less comparison.

did not simulate subtraction in our modeling work. Although computationally subtraction would be analogous to addition for the neural networks and our simulation results won't change if we replace addition with subtraction, in reality the behavioral patterns of human children on addition and subtraction are different. Therefore, it still worth simulating subtraction in future work to see if our model reproduces the learning effect on symbolic subtraction as well.

Finally, we want to note that despite the alternative explanation we proposed for the learning effect in Park and Brannon (2013, 2014), we did not mean to reject the strong connection between non-symbolic number processing and symbolic processing. We only want to advocate for a more cautious interpretation for any learning effect observed in cognitive training and invite other researchers to join our effort in understanding what exactly was learned in different intervention studies.

Acknowledgments

We thank James McClelland for insightful suggestions. A. Y. is funded by the Stanford Interdisciplinary Graduate Fellowship.

References

Feigenson, L., Dehaene, S., & Spelke, E. (2004). Core systems of number. *Trends in Cognitive Sciences*, *8*, 307–314.

Gobel, M. S., Watson, S. E., Lervag, A., & Hulme, C. (2014). Childrens arithmetic development: It is number knowledge, not the approximate number sense, that counts. *Psychological Science*, *25*(3), 789–798. doi: https://doi.org/10.1177/0956797613516471

Halberda, J., Mazzocco, M. M., & Feigenson, L. (2008). Individual differences in non-verbal number acuity correlate with maths achievement. *Nature*, *455*(7213), 665.

Julie, C., & Silke, G. M. (2012). Impact of high mathematics education on the number sense. *PLoS ONE*, *7*(4:e33832). doi: https://doi.org/10.1371/journal.pone.0033832

Kolkman, M. E., Kroesbergen, E. H., & Leseman, P. P. M. (2013). Early numerical development and the role of non-symbolic and symbolic skills. learning and instruction. *Learning and Instruction*, *25*, 95-103.

LeCun, Y., & Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. Retrieved 2016-01-14 14:24:11, from `http://yann.lecun.com/exdb/mnist/`

Libertus, M. E., Feigenson, L., & Halberda, J. (2011). Preschool acuity of the approximate number system correlates with school math ability. *Developmental Science*, *14*(6), 1292–1300.

Mazzocco, M. M. M., Feigenson, L., & Halberda, J. (2011a). Impaired acuity of the approximate number system underlies mathematical learning disability (dyscalculia). *Child Development*, *82*(4), 1224-1237.

Mazzocco, M. M. M., Feigenson, L., & Halberda, J. (2011b). Preschoolers precision of the approximate number system predicts later school mathematics performance. *PLoS ONE*, *6*(9), 1-8.

Park, J., & Brannon, E. M. (2013). Training the approximate number system improves math proficiency. *Psychological Science*, *24*.

Park, J., & Brannon, E. M. (2014). Improving arithmetic performance with number sense training: An investigation of underlying mechanism. *Cognition*, *133*, 188–200.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, *5*(3), 1.

Szkudlarek, E., & Brannon, E. M. (2017). Does the approximate number system serve as a foundation for symbolic mathematics? *Language Learning and Development*, *13*(2), 171–190.

Szkudlarek, E., & Brannon, E. M. (2018). Approximate arithmetic training improves informal math performance in low achieving preschoolers. *Frontiers in Psychology*, 1–12.

Yuan, A., & McClelland, J. (2019). Modeling cross-task transfer by integrating verbal, visual, and grounded action components of natural number. In *Proceedings of the Annual Meeting of the Cognitive Science Society.*

Simulating Problem Difficulty in Arithmetic Cognition Through Dynamic Connectionist Models

Sungjae Cho[1] **(sj.cho@snu.ac.kr), Jaeseo Lim**[1] **(jaeseolim@snu.ac.kr),**
Chris Hickey[1] **(chris.hickey@ucdconnect.ie), Jung Ae Park**[2] **(lydia120@snu.ac.kr),**
Byoung-Tak Zhang[1,3] **(btzhang@bi.snu.ac.kr)**
[1] Interdisciplinary Program in Cognitive Science, Seoul National University,
[2] Department of Psychology, Seoul National University,
[3] Department of Computer Science and Engineering, Seoul National University,
1, Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea

Abstract

The present study aims to investigate similarities between how humans and connectionist models experience difficulty in arithmetic problems. Problem difficulty was operationalized by the number of carries involved in solving a given problem. Problem difficulty was measured in humans by response time, and in models by computational steps. The present study found that both humans and connectionist models experience difficulty similarly when solving binary addition and subtraction. Specifically, both agents found difficulty to be strictly increasing with respect to the number of carries. Furthermore, the models mimicked the increasing standard deviation of response time seen in humans. Another notable similarity is that problem difficulty increases more steeply in subtraction than in addition, for both humans and connectionist models. Further investigation on two model hyperparameters — confidence threshold and hidden dimension — shows higher confidence thresholds cause the model to take more computational steps to arrive at the correct answer. Likewise, larger hidden dimensions cause the model to take more computational steps to correctly answer arithmetic problems; however, this effect by hidden dimensions is negligible.

Keywords: arithmetic cognition; problem difficulty; response time; connectionist model; recurrent neural network; Jordan network; answer step

Introduction

Do connectionist models experience difficulty on arithmetic problems like humans? Although connectionist models consist of abstract biological neurons, similar behaviors between humans and these models are not guaranteed. However, developing model simulations to discover such similarities can bridge this knowledge gap between humans and models, and deepen our understanding of the micro-structures involved in cognition (Rumelhart & McClelland, 1986; McClelland, 1988). Therefore, finding such similarities is a foundational step in understanding human cognition through connectionist models. This connectionist approach recently has been used in the domain of mathematical cognition (McClelland, Mickey, Hansen, Yuan, & Lu, 2016; Mickey & McClelland, 2014; Saxton, Grefenstette, Hill, & Kohli, 2019).

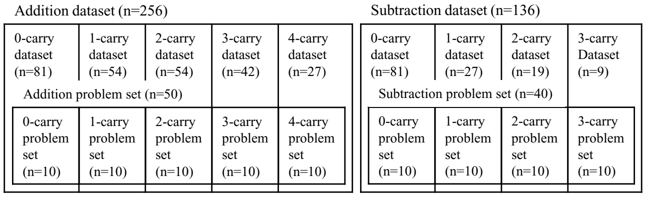
Cognitive arithmetic (Ashcraft, 1992), the study of the mental representation of arithmetic, conceptualizes *problem difficulty*. Problem difficulty can be measured by *response time* (RT) from the time a participant sees an arithmetic problem to the time the participant answers the problem (Imbo, Vandierendonck, & Vergauwe, 2007).

There are three criteria that affect problem difficulty (Ashcraft, 1992; Imbo et al., 2007): (a) operand magnitude (e.g., 1 + 1 vs. 8 + 8); (b) number of digits in the operands (e.g., 3 + 7 vs. 34 + 78); and (c) the number of carry[1] operations (e.g., 15 + 31 vs. 19 + 37). The present study uses a similar experimental approach to that suggested by Cho, Lim, Hickey, and Zhang (2019). This design employs the binary numeral system to control for familiarity with the decimal system and the two criteria (a) and (b). As such, the present study considers the number of carries as the only independent variable involved in problem difficulty.

Recurrent neural networks (Elman, 1990; Jordan, 1997) can model sequential decisions through time. These networks perform sequential nonlinear computations. Owing to the principle that many nonlinear computational steps are required to learn complex mappings (LeCun, Bengio, & Hinton, 2015), parallels can be drawn between human RT and model computational steps in response to problems of varying difficulty level.

Two experiments were conducted in the present study: one on human participants and the other on connectionist models. Both experiments had *learning* and *solving* phases. In the learning phase of the human experiment, participants were taught a method for solving binary arithmetic problems by following guiding examples. In the solving phase, participants began the experiment in earnest, solving arithmetic problems under experimental conditions and having their RTs recorded as a measure of problem difficulty. In the learning phase of the model experiment, connectionist models were trained until they achieved 100% accuracy across all problems. We consider this to be roughly equivalent to how participants were taught to solve arithmetic problems in the learning phase of the human experiment. In the solving phase, all problems were solved again and the number of computational steps taken to solve each problem were recorded as a measure of problem difficulty. Following both experiments, results were analyzed in order to investigate whether any similarities could be observed in how both agents underwent problem difficulty with respect to the number of carries. We then investigated how major model configurations affect model behavior.

[1] A *carry* in binary addition is the leading digit 1 shifted from one column to a more significant column when the sum of the less significant column exceeds a single digit. A *borrow* in binary subtraction is the digit 1 shifted to a less significant column in order to obtain a positive difference in that column. We refer to borrows as carries.



Addition dataset (n=256)

0-carry dataset (n=81)	1-carry dataset (n=54)	2-carry dataset (n=54)	3-carry dataset (n=42)	4-carry dataset (n=27)

Addition problem set (n=50)

0-carry problem set (n=10)	1-carry problem set (n=10)	2-carry problem set (n=10)	3-carry problem set (n=10)	4-carry problem set (n=10)

Subtraction dataset (n=136)

0-carry dataset (n=81)	1-carry dataset (n=27)	2-carry dataset (n=19)	3-carry Dataset (n=9)

Subtraction problem set (n=40)

0-carry problem set (n=10)	1-carry problem set (n=10)	2-carry problem set (n=10)	3-carry problem set (n=10)

Figure 1: Problem sets. The addition and subtraction datasets were assigned to connectionist models. The addition and subtraction problem sets were assigned to participants. *n* refers to the number of operations in a given dataset/problem set.

Problem Sets

Operation datasets For addition and subtraction, we constructed separate *operation datasets*, containing all possible operations between two 4-digit binary nonnegative integers that generate nonnegative results. The addition dataset has 256 operations, and the subtraction dataset has 136 operations (Figure 1). Operation datasets consist of $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is an 8-dimensional input vector that is a concatenation of two binary operands, and $\mathbf{y}$ is an output vector that is the result of computing these operands. $\mathbf{y}$ is 5-dimensional for addition and 4-dimensional for subtraction.

Carry datasets Operation datasets were further subdivided into carry datasets. A *carry dataset* refers to the total set of operations in which a specific number of carries is required for a given operator. The addition dataset was divided into 5 carry datasets, and the subtraction dataset was divided into 4 carry datasets (Figure 1). For example, in Figure 2, the addition guiding examples (a) and (b) are in 2-carry[2] and 4-carry datasets, respectively; the subtraction guiding examples (c) and (d) are in 2-carry and 3-carry datasets, respectively.

Experiment 1: Humans

Experiment 1 investigated whether human RT in problem solving increases as a function of the number of carries involved in a problem.

Participants

90 undergraduate and graduate students (48 men, 42 women) from various departments completed the experiment. The average age of participants was 23.6 ($SD = 3.3$).

Materials

Participants were given two types of problem sets: addition and subtraction. The addition problem set was constructed as follows: 10 different problems were sampled from each carry dataset without replacement[3]. These sampled problems were shuffled together to make the addition problem set. This addition problem set was comprised of 50 unique problems evenly distributed across 5 carry datasets (Figure 1). Likewise, the subtraction problem sets consisted of 40 problems evenly distributed across 4 carry datasets (Figure 1). The problems were newly sampled for each participant.

$$\begin{array}{rl} 10100 & \text{Carry} \\ 1011 & \\ +\ 1010 & \\ \hline 10101 & \\ (a) & \end{array} \qquad \begin{array}{rl} 11110 & \text{Carry} \\ 1111 & \\ +\ 1011 & \\ \hline 11010 & \\ (b) & \end{array} \qquad \begin{array}{rl} 0120 & \text{Carry} \\ 1001 & \\ -\ 0010 & \\ \hline 0111 & \\ (c) & \end{array} \qquad \begin{array}{rl} 0112 & \text{Carry} \\ 1000 & \\ -\ 0101 & \\ \hline 0011 & \\ (d) & \end{array}$$

Figure 2: Guiding examples

In any given problem, two operands were presented in a fixed 4-digit format in order to control for possible extraneous influences on problem difficulty as outlined by criterion (b). The experiment was designed in such a way that participants were required to fill out all digits when answering questions (e.g. if the answer was 1, participants were forced to respond with 0001 as opposed to just 1). This is to ensure RT is not affected by the number of answer digits.

Procedure

Participants were shown calculation guidelines containing two guiding examples for addition (Figure 2a, 2b). Participants were explicitly requested to solve problems by using carry operations outlined in the examples. Participants then began to solve each problem from their addition problem set. After solving all addition problems, participants repeated the previous procedure for their subtraction problem set with two subtraction guiding examples (Figure 2c, 2d). Participants were prohibited from using any writing apparatus in order to force participants to solve problems mentally.

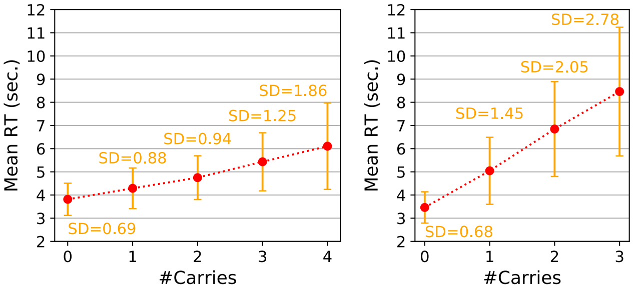
Results

Analysis of variance (ANOVA) was used to investigate differences in mean RTs of participants across carry problem sets. If there were significant differences between all the mean RTs, post hoc analysis was applied. If a participant provided a wrong answer, it was reasonable to assume that this participant made some cognitive error when solving the problem. As such, only RTs for correct answers were included in analysis. We removed the outlying RTs of each carry problem set for each participant since unusually short RTs may be due to memory retrieval and excessively long RTs may be caused by distraction or anxiety during problem solving. The RTs in the range $[Q_1 - 1.5 \cdot \text{IQR}, Q_3 + 1.5 \cdot \text{IQR}]$ were considered outliers, where Q_1 and Q_3 were the first and third quantiles of the RTs for a carry problem set, and $\text{IQR} = Q_3 - Q_1$.

Addition There were significant differences in mean RTs between all carry problem sets, as determined by ANOVA $[F(4,445) = 51.84, p < .001, \eta^2 = .32]$. Post hoc comparisons using the Games-Howell test indicated that mean RTs between any two carry problem sets showed a significant dif-

[2] Let us simply refer to the carry dataset involving *n* carries as the *n*-carry dataset, and problems from the *n*-carry dataset as *n*-carry problems.

[3] This only occurred when sampling 3-carry problems ($n = 10$) from the 3-carry subtraction dataset ($n = 9$). This required one random problem to be duplicated and shown twice in the 3-carry problem set.



(a) Addition

(b) Subtraction

Figure 3: Mean RT by carries. The error bars are ±1*SD*.

ference [3-carry and 4-carry problem sets: $p = .040$; other pairs: $p < .01$]. Therefore, the mean RT was strictly increasing [4] with respect to the number of carries (Figure 3a).

Subtraction There were significant differences in mean RTs between all carry problem sets, as determined by ANOVA [$F(3,356) = 117.41$, $\eta^2 = .50$]. Post hoc comparisons using the Games-Howell test indicated that mean RTs between any two carry problem sets showed a significant difference [$p < .001$]. Therefore, the mean RT was strictly increasing with respect to the number of carries (Figure 3b).

Experiment 2: Connectionist Models

Experiment 2 investigated whether computational steps required by connectionist models in problem solving increase as a function of the number of carries involved in a problem. Moreover, this experiment intended to examine how the central model hyperparameters — confidence threshold and hidden dimension — affect the simulated RT. The hidden dimension, denoted by d_h, refers to the number of units in the hidden layer.

Model

Imagine the human cognitive process while performing addition and subtraction. Humans predict answer digits one by one while mentally referencing two operands and previously predicted digits. Therefore, we aimed to simulate this human cognitive process by using the Jordan network (Jordan, 1997). The Jordan network is a recurrent neural network whose hidden layer gets its inputs from an input at the current step and from the output at the previous step (Figure 4).

The Jordan network solves problems as follows: An 8-dimensional input vector composed of two concatenated 4-digit operands is fed into the network (Figure 4a). At the same time, its hidden layer with ReLU gets its previous probability outputs. The network predicts step-by-step the probabilities of answer digits up to a maximum of 30 steps (Figure 4b). At the initial step, all digit predictions are initialized as 0.5, which mimics the initial uncertainty humans experience when solving problems. The output layer has sigmoid activation. Each output unit predicts each output digit. The network outputs 5-dimensional and 4-dimensional vectors for addition and subtraction problems respectively.

The network learned arithmetic by minimizing the sum of the losses at all steps: $\sum_t H(\mathbf{z}^{(t)}, \mathbf{p}^{(t)})$. At each step t, a loss is defined as the cross-entropy H between the true answer $\mathbf{z}^{(t)}$ and the output probability vector $\mathbf{p}^{(t)}$ where $\mathbf{x}^{(t)}$ is an input vector: $H(\mathbf{z}^{(t)}, \mathbf{p}^{(t)}) = -\mathbf{z}^{(t)} \cdot \log \mathbf{p}^{(t)} - (1 - \mathbf{z}^{(t)}) \cdot [1 - \log \mathbf{p}^{(t)}]$.

At each time step, the network predicts the probability of every answer digit. When problem solving, humans only decide on an answer digit when they are sufficiently confident that it is correct. Likewise, the network decides each digit only when its predicted probability p_i is higher than some threshold. We call this threshold the *confidence threshold*, denoted by θ_c. Suppose $\theta_c = 0.9$. If a predicted probability p_i is in the range $[0.1, 0.9]$, the model is *uncertain* about the digit. Otherwise, it is *confident* about the digit: if $p_i \in [0, 0.1)$, it predicts the digit is 0; if $p_i \in (0.9, 1]$, it predicts the digit is 1. The network is designed to give an *answer* when it is first confident about all answer digits (Figure 4b). The network in Figure 4b answers at step 1 because this is the first state where the model is confident about all digits. At this answer step, the answer is marked as either correct or incorrect. No answer is given if 30 steps are exceeded.

Measures

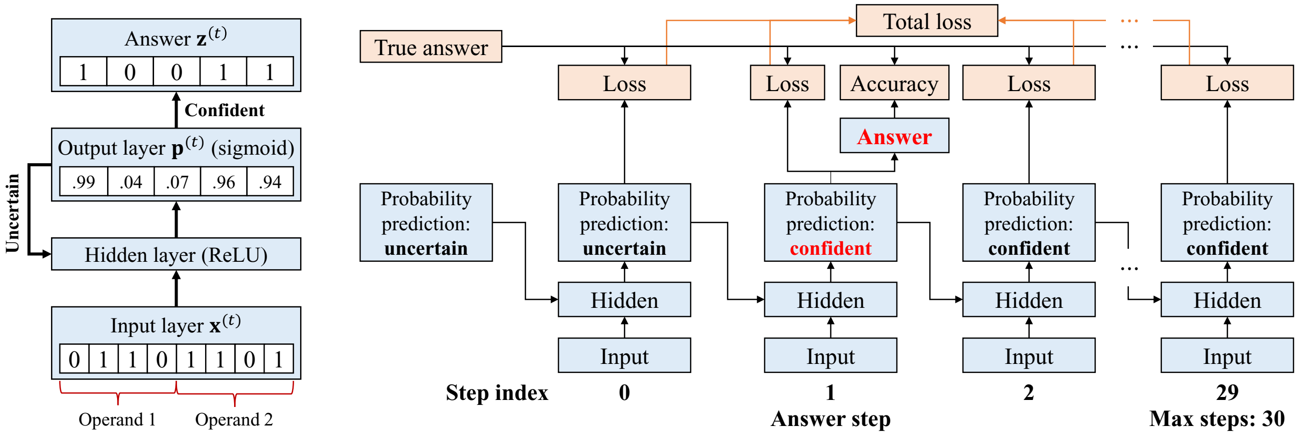
Accuracy *Accuracy* was measured by dividing the number of correct answers by the total number of problems. Model accuracy was used to measure how successfully the model learned arithmetic and to determine when to stop training. No answer after 30 time steps was considered a wrong answer.

Answer step *Answer step* was defined as the index of a certain time step where the network outputs an answer. Answer step is roughly equivalent to human RT. It refers to the number of computational steps required for the network to solve an arithmetic problem. Answer step ranges from 0 to 29.

Training Settings

The network learned arithmetic operations using backpropagation through time (Werbos, 1990) and a stocbohastic gradient method (Bottou, 1998) called Adam optimization (Kingma & Ba, 2015) with settings ($\alpha = .001$, $\beta_1 = .9$, $\beta_2 = .999$, $\varepsilon = 10^{-8}$). For each epoch, 32-sized mini-batches were randomly sampled without replacement (Shamir, 2016) from the total operation dataset. The weight matrix $W^{[l]}$ in layer l was initialized to samples from the truncated normal distribution ranging $[-1/\sqrt{n^{[l-1]}}, 1/\sqrt{n^{[l-1]}}]$ where $n^{[l]}$ was the number of units in the l-th layer; All bias vectors $b^{[l]}$ were initialized to 0. After training each epoch, accuracy was evaluated on the operation dataset (Figure 1). When the network attained 100% accuracy for the entirety of the operation dataset, training was stopped. 300 Jordan networks were trained for each model configuration in order to draw statistically meaningful results. Furthermore, to inves-

[4] For every x and x' such that $x < x'$, if $f(x) < f(x')$, then we say f is *strictly increasing*.


Answer $\mathbf{z}^{(t)}$
1 0 0 1 1
Confident
Output layer $\mathbf{p}^{(t)}$ (sigmoid)
.99 .04 .07 .96 .94
Uncertain
Hidden layer (ReLU)
Input layer $\mathbf{x}^{(t)}$
0 1 1 0 1 1 0 1
Operand 1
Operand 2
True answer
Total loss
Loss
Loss
Accuracy
Loss
Loss
Answer
Probability prediction: uncertain
Probability prediction: uncertain
Probability prediction: confident
Probability prediction: confident
Probability prediction: confident
Hidden
Input
Step index
0
1
Answer step
2
29
Max steps: 30

(a) The Jordan network for addition

(b) The Jordan network unrolled through time steps.

Figure 4: The Jordan network used in the present study. (a) The network is predicting the answer of $110+1101$ to be 10011. In this example, the confidence threshold is 0.9. At the current state t, $\mathbf{x}^{(t)}=(0,1,1,0,1,1,0,1)$, $\mathbf{p}^{(t)}=(.99,.04,.07,.96,.94)$, and $\mathbf{z}^{(t)}=(1,0,0,1,1)$. (b) The network is constrained to compute at most 30 steps. The initial probabilities of answer digits are 0.5, meaning the network is uncertain about all digits. The network repeatedly computes the probabilities of answer digits until it becomes confident about all answer digits; in this figure, it answers at step 1. In the learning phase, the network learns from the total loss from all steps. Accuracy is computed by comparing predicted answers to true answers.

tigate if any statistically significant relationship held for various model configurations, we reanalyzed the models with the confidence thresholds $\theta_c \in \{.7,.8,.9\}$ and hidden dimensions $d_h \in \{24,48,72\}$. 9 types of networks were trained for both addition and subtraction, respectively; a total of 5400 networks were trained in this experiment.

Results

Our proposed model successfully learned all possible addition and subtraction operations between 4-digit binary numbers. The model required 4000 epochs on average (58 minutes[5]) to learn addition, and 1080 epochs on average (13 minutes) to learn subtraction. When training was completed, we examined: (1) statistical differences in mean answer steps between carry datasets across all model configurations; (2) statistical differences in mean answer steps for operation datasets between different confidence thresholds and hidden dimensions.

Addition The first analysis was conducted on mean answer steps per carry dataset. For every model configuration, ANOVA found significant differences in mean answer steps between all carry datasets (Table 1). Post hoc Games-Howell testing found that for 8 of the 9 model configurations, mean answer step was strictly increasing with respect to the number of carries (Table 1, Figure 5a); the remaining model configuration ($\theta_c = 0.7$, $d_h = 24$) showed a monotonically[6] increasing relationship between mean answer step and the number of carries (Table 1).

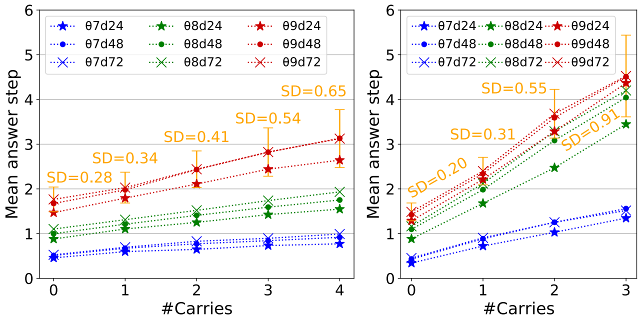
The second analyses were conducted on mean answer steps for the addition dataset. For every hidden dimension, ANOVA found significant differences in mean answer steps between all confidence thresholds $^\forall\theta_c \in \{.7,.8,.9\}$ (Table 2). Post hoc Games-Howell testing found that for all models, mean answer step was strictly increasing with respect to confidence threshold (Table 2, Figure 6a). For every confidence threshold, ANOVA found significant differences in mean answer steps between all hidden dimensions $^\forall d_h \in \{24,48,72\}$ (Table 3). Post hoc Games-Howell testing found that with $\theta_c = 0.7$, mean answer step was monotonically increasing with respect to hidden dimension. For both other confidence thresholds, mean answer step was strictly increasing with respect to hidden dimension (Table 3, Figure 7a). We should note however that while significant, the effect of hidden dimension on mean answer step was small.

Subtraction The first analysis was conducted on mean answer steps per carry dataset. For every model configuration, ANOVA found significant differences in mean answer steps between all carry datasets (Table 1). Post hoc Games-Howell testing found that for all model types, mean answer step was strictly increasing with respect to the number of carries (Table 1, Figure 5b).

The second analyses were conducted on mean answer steps for the subtraction dataset. For every hidden dimension, ANOVA found significant differences in mean answer steps between all confidence thresholds $^\forall\theta_c \in \{.7,.8,.9\}$ (Table 2). Post hoc Games-Howell testing found that for all models, mean answer step was strictly increasing with respect to confidence threshold (Table 2, Figure 6b). For every confidence threshold, ANOVA found significant differences in mean answer steps between all hidden dimensions $^\forall d_h \in \{24,48,72\}$

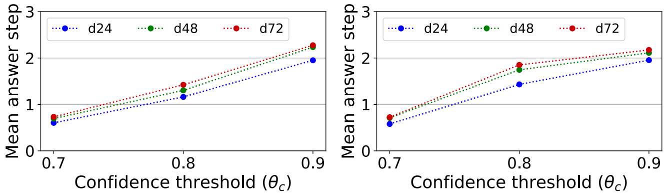
[5] Two Intel(R) Xeon(R) CPU E5-2695 v4 and five TITAN Xp were used. Training networks in parallel is vital in this experiment.

[6] For every x and x' such that $x < x'$, if $f(x) \leq f(x')$, then we say f is *monotonically increasing*.



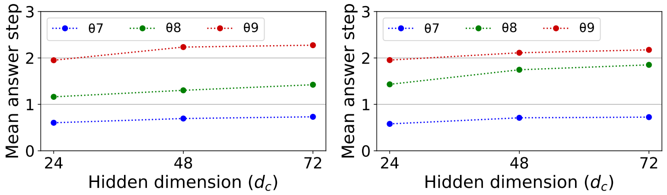
(a) Addition (b) Subtraction

Figure 5: Mean answer step by carries (for carry datasets). θ9d72 denotes models with $\theta_c = 0.9$ and $d_h = 72$. The error bars are $\pm 1 SD$ and belong to θ9d72.



(a) Addition (b) Subtraction

Figure 6: Mean answer step by confidence threshold (for operation datasets)



(a) Addition (b) Subtraction

Figure 7: Mean answer step by hidden dimension (for operation datasets)

(Table 3). Post hoc Games-Howell testing found that with $\theta_c = 0.9$, mean answer step was monotonically increasing with respect to hidden dimension. For both other confidence thresholds, mean answer step was strictly increasing with respect to hidden dimension (Table 3, Figure 7a). We should note however that while significant, the effect of hidden dimension on mean answer step was small (Figure 7a).

Discussion and Conclusion

Experiment 1 Experiment 1 has improved the previous study (Cho et al., 2019) as follows: Firstly, participants were forced to solve problems using solely mental arithmetic. This allows for more valid comparisons to be drawn between humans and models. Secondly, larger data samples allowed the present study to find more statistically significant results. Specifically, mean RT for addition problems were found to be strictly increasing with respect to the number of carries.

Table 1: The results of ANOVA and post hoc analysis on differences in mean answer steps between all carry datasets. The model configuration varies along two axes: confidence threshold and hidden dimension. 300 mean answer steps per carry dataset from 300 trained networks were analyzed for each model configuration. F is the F-test statistic and η^2 is the effect size from ANOVA; in addition, there were 4 degrees of freedom between carry datasets and 1495 within carry datasets: $\mathrm{df}_b^+ = 4$, $\mathrm{df}_w^+ = 1495$; in subtraction, $\mathrm{df}_b^- = 3$, $\mathrm{df}_w^- = 1196$. The mean answer step columns describe the results of post hoc analysis. The inequality ($<$) denotes a significant difference at the $p < .05$ level. Equality ($=$) denotes the opposite. The numbers in these columns refer to the number of carries of a carry dataset. * $p < .05$. ** $p < .01$. *** $p < .001$.

		Addition			Subtraction		
θ_c	d_h	F	η^2	Mean answer step	F	η^2	Mean answer step
.7	24	72^{***}	.16	$0 < 1 = 2 < 3 = 4^{***}$	499^{***}	.56	$0 < 1 < 2 < 3^{***}$
.7	48	206^{***}	.36	$0 < 1 < 2 < 3 < 4^{***}$	765^{***}	.66	$0 < 1 < 2 < 3^{***}$
.7	72	294^{***}	.44	$0 < 1 < 2 < 3 < 4^{***}$	716^{***}	.64	$0 < 1 < 2 < 3^{***}$
.8	24	129^{***}	.26	$0 < 1 < 2 < 3 < 4^{**}$	390^{***}	.49	$0 < 1 < 2 < 3^{***}$
.8	48	198^{***}	.35	$0 < 1 < 2 < 3 < 4^{***}$	571^{***}	.59	$0 < 1 < 2 < 3^{***}$
.8	72	142^{***}	.28	$0 < 1 < 2 < 3 < 4^{**}$	674^{***}	.63	$0 < 1 < 2 < 3^{***}$
.9	24	208^{***}	.36	$0 < 1 < 2 < 3 < 4^{**}$	970^{***}	.71	$0 < 1 < 2 < 3^{***}$
.9	48	421^{***}	.53	$0 < 1 < 2 < 3 < 4^{***}$	1769^{***}	.82	$0 < 1 < 2 < 3^{***}$
.9	72	432^{***}	.54	$0 < 1 < 2 < 3 < 4^{***}$	1718^{***}	.81	$0 < 1 < 2 < 3^{***}$

Table 2: The results of ANOVA and post hoc analysis on differences in mean answer steps between confidence thresholds. $\mathrm{df}_b^+ = \mathrm{df}_b^- = 2$. $\mathrm{df}_w^+ = \mathrm{df}_w^- = 897$. In the mean answer step columns, the numbers refer to confidence thresholds.

	Addition			Subtraction		
d_h	F	η^2	Mean answer step	F	η^2	Mean answer step
24	1032^{***}	.70	$.7 < .8 < .9^{***}$	1163^{***}	.72	$.7 < .8 < .9^{***}$
48	2002^{***}	.82	$.7 < .8 < .9^{***}$	1736^{***}	.79	$.7 < .8 < .9^{***}$
72	1735^{***}	.79	$.7 < .8 < .9^{***}$	1963^{***}	.81	$.7 < .8 < .9^{***}$

Table 3: The results of ANOVA and post hoc analysis on differences in mean answer steps between hidden dimensions. $\mathrm{df}_b^+ = \mathrm{df}_b^- = 2$. $\mathrm{df}_w^+ = \mathrm{df}_w^- = 897$. In the mean answer step columns, the numbers refer to hidden dimension.

	Addition			Subtraction		
θ_c	F	η^2	Mean answer step	F	η^2	Mean answer step
.7	58^{***}	.08	$24 < 48 = 72^{***}$	46^{***}	.10	$24 < 48 < 72^{**}$
.8	38^{***}	.08	$24 < 48 < 72^{***}$	77^{***}	.15	$24 < 48 < 72^{**}$
.9	37^{***}	.12	$24 < 48 < 72^{*}$	51^{***}	.09	$24 < 48 = 72^{***}$

Experiment 2 In Experiment 2, the two hyperparameters — confidence threshold and hidden dimension — were chosen since we expected these hyperparameters to correspond to humans' uncertainty and memory capacity, respectively. We

further expected that increasing confidence threshold and decreasing hidden dimension would increase answer step. This expectation subsequently arose for confidence threshold; confidence threshold had an augmenting effect on answer step. However, our expectation was not born out for hidden dimension. In order to observe clear differences in mean answer steps with respect to problem difficulty, high confidence thresholds are recommended. Hidden dimension should be fixed to the extent that the model can learn an entire dataset.

Experiments 1 & 2 The preceding results show three notable similarities between humans and our connectionist models: Firstly, both agents experienced increased levels of difficulty as more carries were involved in arithmetic problems. Secondly, the Jordan networks with the model configuration ($\theta_c = 0.9$, $d_h = 72$) successfully mimicked the increasing standard deviation of human RT with respect to the number of carries (Figure 3, 5). This phenomenon could not be achieved by a rule-based system performing the standard algorithm, although such a system would be able to simulate increasing RT as a function of the number of carries. Lastly, another similarity found between both humans and models is that the difficulty slope for subtraction is steeper than for addition (Figure 3, 5). This implies that the augmenting effect of carries on problem difficulty is stronger in subtraction than in addition.

Contributions The present study makes two major contributions to the literature: Firstly, our models successfully simulated humans' RT in terms of these three similarities: increasing latency, increasing standard deviation of latency, and relative steepness of increasing latency. The similarities may suggest that some cognitive process, equivalent to the nonlinear computational process used in the Jordan network, could be involved in human cognitive arithmetic. Secondly, the present study demonstrated that fitting our model to arithmetic data induced human-like latency to emerge in the connectionist models (McClelland et al., 2010). In other words, human RTs to arithmetic problems were successfully learned in an unsupervised way. This contrasts with previous studies that focus on learning arithmetic tasks in a supervised way.

Future Study The present study focuses solely on analyzing mean answer steps between arithmetic problem sets of varying difficulty levels. Therefore, future studies could aim to better understand what dynamic processes our model uses when solving individual problems: Specifically, it might be interesting to observe how our model predicts individual digits through each time step when solving problems. Furthermore, similarities between both the model's sequentially predictive answering process and the human answering process could be investigated. This comparison would give us a better understanding of both our model and human mathematical cognition (McClelland et al., 2016).

Our model is designed not just for arithmetic cognition, but also for sequential predictions that based on a constant input and a previous prediction, which result in a single answer. In this regard, this model has the potential to be applied to other cognitive processes involving sequential processing and RT as a measure of cognitive difficulty. Therefore, future studies could consider extending our model to other domains of cognition. For example, well known character image and word classification datasets can be subdivided into datasets of varying difficulty levels, similar to our carry datasets. Mean answer steps for classifying these data sets could be analyzed using a similar model to that outlined in the present study.

Acknowledgments

We thank Seho Park, Seung Hee Yang, Chung-Yeon Lee, Gi-Cheon Kang, and Paula Higgins for useful discussion and writing comments. This work was supported by a grant to Biomimetic Robot Research Center funded by Defense Acquisition Program Administration and Agency for Defense Development (UD160027ID).

References

Ashcraft, M. H. (1992). Cognitive arithmetic: A review of data and theory. *Cognition*, *44*(1-2), 75–106.

Bottou, L. (1998). *Online algorithms and stochastic approximations.* Cambridge University Press.

Cho, S., Lim, J., Hickey, C., & Zhang, B.-T. (2019). Problem difffculty in arithmetic cognition: Humans and connectionist models. In *Proceedings of the 41st annual conference of the cognitive science society.*

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Imbo, I., Vandierendonck, A., & Vergauwe, E. (2007). The role of working memory in carrying and borrowing. *Psychological Research*, *71*(4), 467–483.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology* (Vol. 121, pp. 471–495).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations.*

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

McClelland, J. L. (1988). Connectionist models and psychological evidence. *Journal of Memory and Language*, *27*(2), 107–123.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends in cognitive sciences*, *14*(8), 348–356.

McClelland, J. L., Mickey, K., Hansen, S., Yuan, A., & Lu, Q. (2016). A parallel-distributed processing approach to mathematical cognition. *Manuscript, Stanford University.*

Mickey, K. W., & McClelland, J. L. (2014). A neural network model of learning mathematical equivalence. In *Proceedings of the annual meeting of the cognitive science society.*

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing* (Vol. 1). MIT Press.

Saxton, D., Grefenstette, E., Hill, F., & Kohli, P. (2019). Analysing mathematical reasoning abilities of neural models. In *7th international conference on learning representations.*

Shamir, O. (2016). Without-replacement sampling for stochastic gradient methods. In *Advances in neural information processing systems 29: NIPS 2016* (pp. 46–54).

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Modeling Cognitive Dynamics in End-User Response to Phishing Emails

Edward A. Cranford (cranford@cmu.edu) and Christian Lebiere (cl@cmu.edu)
Department of Psychology, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Prashanth Rajivan (prajivan@uw.edu)
Department of Industrial & Systems Engineering, University of Washington
3900 E Stevens Way NE, Seattle, WA 98195 USA

Palvi Aggarwal (palvia@andrew.cmu.edu) and Cleotilde Gonzalez (coty@cmu.edu)
Dynamic Decision Making Laboratory, Social and Decision Sciences Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Abstract

Phishing attacks are a significant threat to cybersecurity, while current defense methods do not adequately address the human factor of this threat: the role of experiences and cognitive biases. To better understand human susceptibilities to phishing attacks, we developed an Instance-Based Learning (IBL) model for predicting end-user's behavior in a phishing email detection task. We present a phishing scenario that demonstrates that typically safe end-users can fall victims to phishing attacks in certain circumstances, and these situations are the result of cognitive mechanisms such as frequency and recency and similarities between memory events. We demonstrate the ability of an IBL model to predict human performance in a laboratory phishing detection task. While the results indicate that phishing detection was difficult for the model, it roughly reflects in the data the difficulty humans had. Future research is aimed at enhancing the IBL model to better predict end-user phishing detection, and to explore the ways in which this model can be used as a training tool and online aid for end-user detection of phishing attacks.

Keywords: phishing; cybersecurity; decision making; instance-based learning; cognitive model; ACT-R

Introduction

All it takes is one click in response to a phishing email to compromise the security posture of an entire organization, and as such phishing attacks pose the biggest threat for cybersecurity (Wombat Security report, 2018). Phishing aims to persuade end-users to share sensitive information using social engineering and psychological techniques (Jagatic et al., 2007). While phishing attacks exploit human weaknesses, defenders typically employ technological solutions to defend against them, such as machine learning filtering of phishing emails, email authentication tools, URL filtration, and blacklisting phishing URLs (Prakash et al., 2010; Marchal et al., 2014; Peng, Harris, & Sawa, 2018). Current methods of defense against phishing attacks are insufficient because they don't consider human cognitive biases and experience. Since the success of phishing attacks rely on the exploitation of end-user's cognitive and psychological weaknesses, it becomes essential to understand the detection capabilities, decision making, and cognitive biases of end users who respond to phishing emails (Canfield, Fischhoff, & Davis, 2016).

Considerable research has been devoted to investigating how to best train end-users to detect phishing emails (Kumaraguru et al. 2009, Jensen et al., 2017), yet even trained end-users can still fall victim to phishing attacks. Recent research examining the interaction between attackers and end-users revealed various strategies that attackers use to design phishing campaigns and their success on end-user's detection of phishing emails (Rajivan & Gonzalez, 2018; Curtis et al., 2018, Singh et al., 2019). In the current research, based on psychological theories of decisions from experience, and the insights of these recent phishing studies, we propose a cognitive model of end-user phishing email detection. Our insights suggest that phishing emails detection is influenced by the end-user's prior history of emails, their recent experiences, and their innate and learned cognitive biases.

In what follows, we first describe an example phishing scenario that reveals the process by which an end-user might fall trap to an attacker's social engineering strategies. We then formalize a cognitive model of end-user email classification, built in the ACT-R cognitive architecture (Anderson & Lebiere, 1998), using Instance-based Learning Theory (IBLT; Gonzalez, Lerch, & Lebiere, 2003). Using the data set from Rajivan and Gonzalez (2018), we demonstrate that cognitive models of end-user detection of phishing attacks can be useful for understanding how and when humans are most vulnerable to attacks, providing insights on how to best train people to detect phishing emails, and could potentially serve as a powerful decision support tool to prevent phishing attacks.

A Cognitive Model of Phishing Email Detection

In the example phishing scenario, depicted in Figure 1, Alice is a representative persona for a class of members of a fictional organization. The cyber-security division is assessing vulnerabilities of phishing attacks and sends Alice a number of emails, some of which are phishing emails. Her task is to decide whether to click a link within an email.

Alice represents a particularly savvy end-user, who usually recognizes malicious emails, and does not click on embedded links. In this scenario, Alice starts with a prior history of not clicking links from unknown senders (i.e., senders that she has not previously interacted with and whom she does not

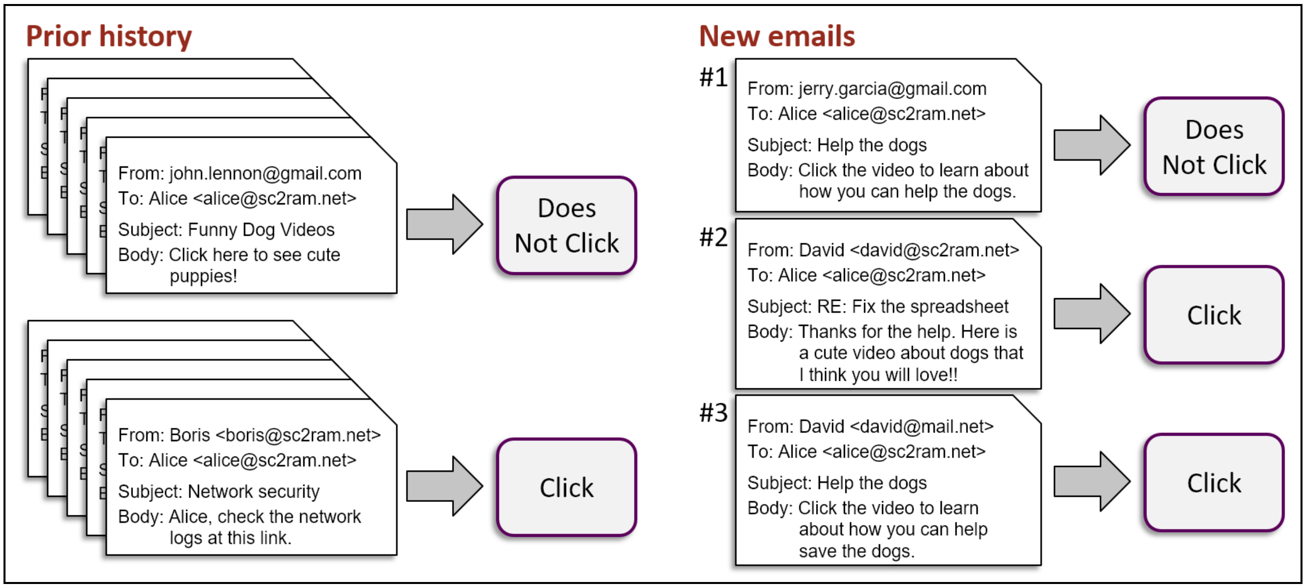

Prior history
From: john.lennon@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Funny Dog Videos
Body: Click here to see cute puppies!
Does Not Click
From: Boris <boris@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: Network security
Body: Alice, check the network logs at this link.
Click
New emails
#1
From: jerry.garcia@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help the dogs.
Does Not Click
#2
From: David <david@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: RE: Fix the spreadsheet
Body: Thanks for the help. Here is a cute video about dogs that I think you will love!!
Click
#3
From: David <david@mail.net>
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help save the dogs.
Click

Figure 1: The Alice scenario: an example phishing attack scenario.

recognize). She has a predisposition to click on links from trusted senders (e.g., coworkers and friends), particularly about topics that interest her (e.g., one of Alice's interests is in dogs). She is then presented with three new emails, one at a time. The first is from an unknown sender about dogs, for which she does not click the link. The second is from a trusted coworker that mentions dogs, for which she clicks the link. In the third email, after observing and/or inferring Alice's clicking behavior, the attacker spoofs the sender's source address, pretending to be a colleague of Alice, and baits her with a topic and domain name related to dogs. Alice clicks the link and self-compromises her system.

Alice's behavior can be described as emerging from the interaction between her learned behavior/tendencies and changes to the environment. The cognitive model, described next, captures underlying cognitive mechanisms such as priming, transfer, and recency bias that reflect the statistics and dynamics of the environment and give rise to Alice's behavior. As shown in Figure 2, Alice's prior history of emails may cluster on dimensions of email topic (work, dogs) and sender (known, unknown). Emails about dogs from unknown senders cluster together and embedded links are typically not clicked. Whereas, emails about work topics from known coworkers cluster together, and embedded links are typically clicked. The first email is similar to past emails for which she did not click on embedded links, and so she doesn't. The second email is from trusted coworkers, but mentions dogs, yet is more similar to past emails for which she clicked on links, and so she does. This expands the cluster of emails for which she previously clicked. Alice would typically not click on the link in the third email, because it is more similar to past emails for which she did not click embedded links. However, it is more similar to the recent second email, and so is pulled toward the cluster of emails for which she clicked links. Alice's normal behavior has changed as a result of her interactions with the environment over time.

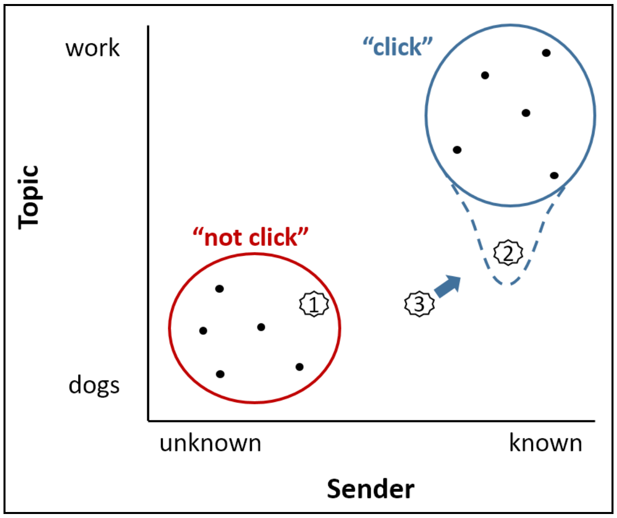


Figure 2: Representation of Alice's behavior.

An IBL Model of Phishing Detection

According to IBLT (Gonzalez et al., 2003), decisions are made by generalizing across past experiences, or instances, that are similar to the current situation. Typically, instances are encoded as chunks in declarative memory that represent the features of the decision: the context in which a decision is made, the action taken, and the outcome of that decision. For emails, there is usually a dissociation between the actions taken and feedback regarding whether the email was ultimately malicious. Therefore, for this task, only the context and the action are represented within each instance, but not the outcome. The context elements of an email include the sender's email address, the subject line, the body of the email, and the link. The action slot includes the action taken (either click or not click the link). Initial past instances include those represented in Figure 1 under *Prior History*: five emails from unknown senders about various topics, including puppies, for which Alice did *not* click on the embedded links, and five emails from trusted coworkers, about work-related topics, for which Alice clicked on the embedded links.

An IBL cognitive model was constructed in the ACT-R cognitive architecture (Anderson & Lebiere, 1998). For each new incoming email (see Figure 1, "New Emails"), the model takes as input the context of the email and generates an action by retrieving similar past instances. In ACT-R, the retrieval of past instances is based on the activation strength of the relevant chunk in memory and its similarity to each of the elements of the current context. The activation A_i of a chunk *i* is determined by the following equation:

$$A_i = \ln \sum_{j=1}^{n} t_j^{-d} + MP * \sum_{k} Sim(v_k, c_k) + \varepsilon_i$$

The first term provides the power law of practice and forgetting, where t_j is the time since the *j*th occurrence of chunk *i* and *d* is the decay rate of each occurrence which is set to the default ACT-R value of 0.5. The second term reflects a partial matching process, where $Sim(v_k,c_k)$ is the similarity between the actual memory value and the corresponding context element for chunk slot *k*, and is scaled by the mismatch penalty (MP) which was set to the default value of 1.0. The term ε_i represents transient noise, a random value from a logistic distribution with a mean of zero and variance parameter *s* of 0.25 (common ACT-R value, e.g. Lebiere, 1999), to introduce stochasticity in retrieval.

The probability of retrieving a particular instance is determined according to the softmax equation (i.e., the Boltzmann equation), reflecting the ratio of an instance's activation A_i and the temperature t (which was set to 1.0):

$$P_i = \frac{e^{A_i/t}}{\sum_j e^{A_j/t}}$$

The IBL model uses ACT-R's blending mechanism (Lebiere, 1999, Gonzalez et al., 2003) to generate an action, based on past instances. Blending is a memory retrieval mechanism that returns a consensus value across all memories with similar context elements, rather than from a specific memory, as computed by the following equation:

$$\underset{V}{\operatorname{argmin}} \sum_{i} P_i \times \left(1 - Sim(V, V_i)\right)^2$$

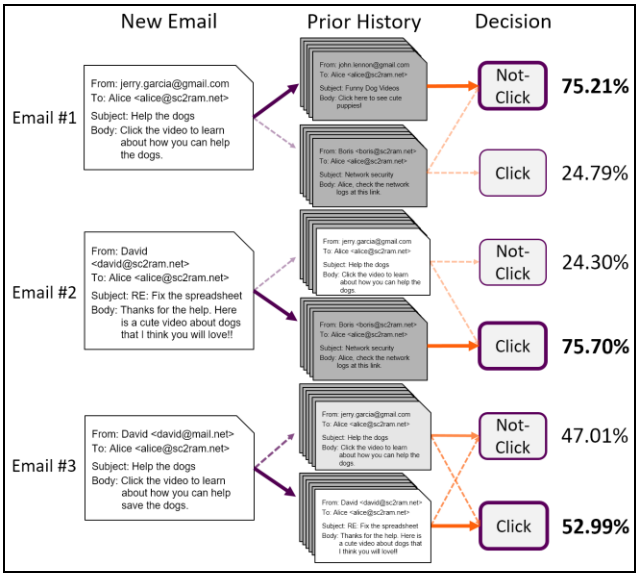
The value V is the one that best satisfies the constraints among actual values V_i in the matching chunks *i* weighted by their probability of retrieval P_i. Satisficing is defined as minimizing the dissimilarity between the consensus value V and the actual answer V_i contained in chunk *i*. In summary, the model matches memories to the current context and uses blending to generate the action. After generating an action, the experience (context plus action) is saved in declarative memory as a new instance, which affects future decisions.

An important feature of the model is how similarities are computed between slot values. Typically, similarities between numeric values are computed on a linear function scaled between 0 and -1.0, where 0 is a perfect match and -1.0 is maximally dissimilar. However, for non-numeric information, unless a value is specified for relation, they are either maximally similar or maximally different. For emails, the context is non-numeric, often several words to paragraphs in length. It is sensible then that two texts that are semantically similar should have higher similarity values (closer to 0) compared to texts that are semantically very dissimilar.

In order to compute similarities between slot contents involving textual information, we used the University of Maryland Baltimore County's semantic-textual-similarity tool (Han et al., 2013). The tool uses a combination of latent semantic analysis (LSA) and WordNet to produce semantic similarity values between two texts. The two input texts can be of any word-length and it produces a value between 0.0 and 1.0, with 1.0 being maximally similar in meaning. For example, the similarity between "happy dog" and "joyful puppy" is 0.65, whereas "happy dog" and "sad feline" is 0.34, and "happy dog" and "hot tea" is 0.0. We subtract one from this value to produce a dissimilarity value for use in blending. This technique has proven to be a useful methodology for producing meaningful similarity values for textual content.

Demonstration of the IBL Model Behavior

Figure 3 shows the model behavior during a typical run through the Alice scenario. The first column shows the new incoming emails. The second column shows Alice's prior history of emails stored in memory: the top stack shows emails for which Alice previously did not click on the embedded link, while the bottom stack shows emails for which she did click. For each new email, the model retrieves a decision based on its similarity to prior emails. The darker the email, the less recent it was experienced and encoded in memory. Darker, fuller arrows indicate greater activation strength (purple) or decision weighting (orange). The third column shows the blending values (i.e., the relative weighting given to each option based on activation and similarities) next to the two possible decisions (Click or Not-Click). The decision made is that with the greater blending value.



From: jerry.garcia@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help the dogs.

From: john.lennon@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Funny Dog Videos
Body: Click here to see cute puppies!

From: Boris <boris@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: Network security
Body: Alice, check the network logs at this link.

From: David <david@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: RE: Fix the spreadsheet
Body: Thanks for the help. Here is a cute video about dogs that I think you will love!!

From: jerry.garcia@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help the dogs.

From: Boris <boris@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: Network security
Body: Alice, check the network logs at this link.

From: David <david@mail.net>
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help save the dogs.

From: jerry.garcia@gmail.com
To: Alice <alice@sc2ram.net>
Subject: Help the dogs
Body: Click the video to learn about how you can help the dogs.

From: David <david@sc2ram.net>
To: Alice <alice@sc2ram.net>
Subject: RE: Fix the spreadsheet
Body: Thanks for the help. Here is a cute video about dogs that I think you will love!!

Figure 3: Example model behavior in the Alice scenario.

For example, for Email #1, its context is more similar to past emails from unknown senders than to those from trusted coworkers. The blending mechanism produces a distance metric from each possible decision, and since there are only two possible decisions, blending values can be translated directly into the weighted probability of making each decision. Therefore, for this particular run, the model decides to not click the link with a weighting of 75.21%. For Email #2, its context is more similar to emails from trusted coworkers, and the model decides to attack with a weighting of 75.7%. For the critical Email #3, although the contents are typically more similar to unknown senders, it also shares similarity with the most recent email from a trusted coworker giving more weight to the decision to click on the link (e.g., recency bias to click on a link about dogs from a David, because of the similarity to the contents of Email #2). On this run, the combination of activation strength and similarity across past instances result in a decision to click on the link with a weighting of 52.99%.

This example shows how under certain circumstances a relatively safe user could sometimes get caught performing an unsafe act. To generate stable predictions of human behavior, the model was run 1000 times to highlight its activation dynamics. For Email #1, the model decided to click the link on less than 1% of the 1000 runs, with a mean weighting in favor of not clicking of 66.3%. For Email #2, the model clicked the link on 98.8% of runs, with a mean weighing in favor of not clicking of 66.2%. For Email #3, the model clicked the link on 56.6% of runs, with a mean weighing in favor of clicking of 50.8%. Of course, that action itself will make further dangerous actions more likely.

The IBL model of the Alice phishing scenario shows how a user's response to phishing emails may be highly constrained by cognitive mechanisms, especially activation in declarative memory, which reflect the statistics and dynamics of the environment in the user's memory. Alice's behavior is a result of manipulating that environment in a way that can change well established behaviors. As demonstrated, it only takes a short history of human behavior, and their interests, to personalize a model to an individual user and make predictions about whether the user might perform an unsafe act when encountering a malicious email.

Validation of the IBL Model Against Humans

To assess performance of the IBL model described above, it was adapted to predict human behavior in a laboratory experiment, reported in Rajivan and Gonzalez (2018). Their data set includes 340 participants as end-users in an email management task. Participants were presented with 20 emails, one at a time; 10 were benign emails and 10 were phishing emails, randomly distributed. Their task was to assist a fictional office manager by examine each of her incoming emails and decide how to respond: 1) respond immediately; 2) flag the email for follow up; 3) leave the email in the inbox; 4) delete the email; or 5) delete the email and block the sender. An email rated as 1 can be viewed as more benign and important, while an email rated as 5 is more malicious.

For this task, the chunk definitions of the model were modified to represent the information available to participants. For these emails, there was no sender information available, but links were represented both as the HTML link as well as the observable text in the email. Therefore, the context slots include the subject, body, link, and link text. The decisions were recoded to be analogous to the conceptual model, with ratings of 1 and 2 recoded as "respond" (i.e., the equivalent of clicking a link) and ratings of 3 through 5 recoded as "do not respond" (i.e., the equivalent of not clicking a link). Therefore, for the model, the possible decisions are *respond* or *not-respond.* All parameters were left the same as for the conceptual model.

Results

The model was run 10 times for each participant and was presented the same stimuli experienced by the human. The first 10 emails experienced served as training instances for the model and were encoded as an initial declarative memory. The model then made a decision for each of the next 10 emails, and its predictive accuracy was evaluated.

The model performed better than chance (50%), accurately predicting the human's decision on 58.6% of benign emails and 63.4% of phishing emails, on average. The model was more accurate on phishing emails than benign emails, $F(1,9) = 10.12$, $p = 0.001$. There were no differences across trials and the interaction was not significant, both p's > 0.43.

The confusion matrices presented in Figure 4, show the percentage of trials in which the model and human agreed in their decisions to respond to the email or not, for phishing emails (top) and benign emails (bottom). D-prime for phishing emails is 0.60, while it is 0.43 for benign emails. Figure 4 also shows the phishing detection accuracy of humans and the model. For both phishing and benign emails, the model and humans decided to respond to ~40% of emails or more. As a result, the model more accurately predicts human decisions to *not* respond to an email than to respond.

Like humans, the model responded to a large proportion of phishing emails (39.7% and 39.0% respectively). Although, while humans responded to more benign emails (47.9%), the model responded to only 39.5% of benign emails – almost the same rate as phishing emails, indicating that distinguishing between ham and phishing emails was difficult.

Discussion

Humans were less cautious in the email management task than they might normally be in real-world circumstances, and the IBL model reflected this behavior, and responded to many phishing emails. Overall, the model was better than chance at predicting human performance, but the task proved difficult for both the humans and the model without rewards or feedback to aid learning. The model was trained on the first 10 trials of human data, and therefore reflects the overall tendencies to not respond. However, while the model is similarly as biased as humans to not respond to emails, it has a slightly more difficult time distinguishing a benign email from a phishing email than humans.

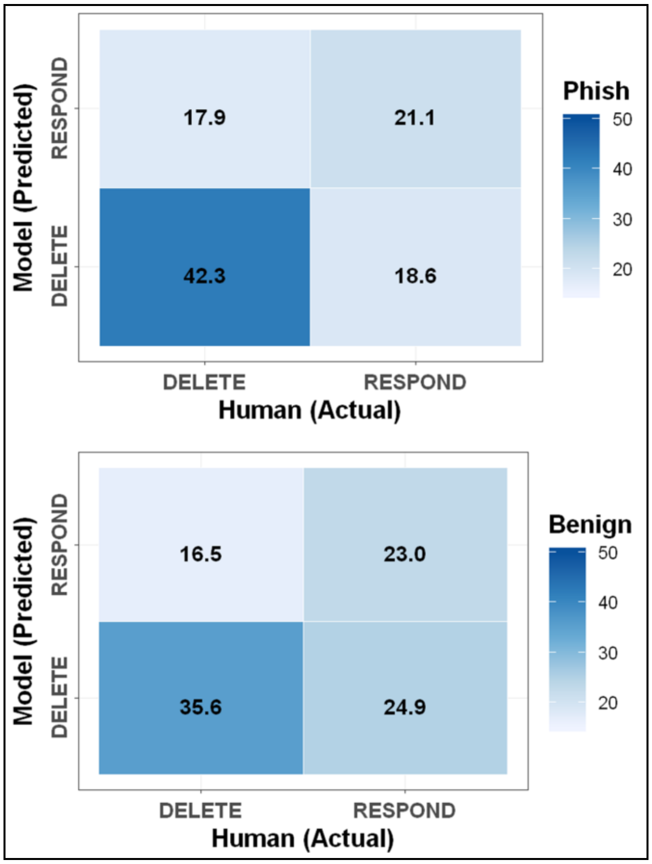


Figure 4: Confusion matrices comparing the model predictions of human decisions (in percentages) for phishing emails (top) and benign emails (bottom).

The benign emails in Rajivan and Gonzalez (2018) were ham emails that came from businesses and senders, and were about topics and accounts, that the end user could not know were relevant to their fictional office manager. Without context, the benign emails look and sound very similar to the phishing emails, making the detection of phishing emails difficult. In fact, when looking into the UMBC similarities within and between benign and phishing emails, the values are very close to each other. The mean similarities between benign emails are the highest, but still relatively low, at only 0.43. Meanwhile, the phishing emails are as dissimilar to each other (0.36) as they are to benign emails (0.39). The model accurately captures overall human tendencies, but has more difficulty than humans in classifying a benign email as safe.

Limitations and Future Directions

There is clear room for improvement for the IBL model. It is limited by its representation of the relevant features for detecting phishing emails. Research in human susceptibility to phishing scams has revealed important cues and indicators of phishing emails that end-users should be trained to detect (Vishwanath, Harrison, & Ng, 2018). While the sender, subject line, URL, and the email body are all important features to use for detection, representing only the semantic content limits the model's ability to discriminate. Some features that could be extracted from the email to enhance the representation include grammar and spelling ratings, emotional tone of the email, and sentiment. Future research is aimed at exploring and expanding the features that represent the context of an email. It is also unclear at this point what features the model relies on most to make decisions or if any do not affect decisions. The features need to be accurately represented in the context to accurately reflect the statistical dynamics of the environment. Representing user interests, as well as background knowledge of known senders, are additional features that would greatly improve the model's ability to predict a particular individual's behavior.

On a related note, while the UMBC semantic similarity tool proved useful, many of the similarity values between emails are in the range of 0.33-0.66. Adjusting these values so they fill the full range of 0.0-1.0 could help to increase the dissimilarity between the benign and phishing emails while increasing the similarity within email types. Additionally, the similarities are computed between entire email bodies. These bodies could be parsed into separate phrases to uncover more fine-grained features. Future research is aimed at exploring these possibilities.

Improving the cognitive model of phishing detection is an important goal for gaining a better understanding of end-user susceptibility to phishing emails. Additionally, there is a wide array of possible applications in cybersecurity, including using cognitive models to help train end users to detect phishing attacks. A cognitive model that can track a user's experience helps reveal instances when a user may be more susceptible to a phishing scam. The model can make the user aware of such instances to improve their detection. Predicting individual end user behavior is a challenging task but could be extremely helpful in aiding end users in online detection.

After improving the cognitive model, the model can be scaled up to larger applications. For example, cognitive models could also be used to estimate the risk of new phishing samples, or as part of a larger simulation testbed for cyber defense exercises, or to test tools. For applications such as these, scalability becomes an issue for computing semantic similarities. Tools like UMBC's similarity tool typically look up information from very large databases. If you only need to compute a few values per iteration, then computation costs are minimal. However, computation time increases exponentially as the number of instances in the model's declarative memory increases. One technique that proved useful for us was to build a hash-table that stores similarity values between two phrases, thereby eliminating the need to re-compute values for distinct pairs of phrases. If the corpus of emails is known, then these values can be computed prior to running the model. Otherwise, the model would only be able to reuse values after the first experience. Another approach is to use vector embeddings, then compute similarities as distances between vectors.

In the Rajivan and Gonzalez (2018) study, participants saw a large proportion of phishing emails compared to benign emails (50% precisely). Using the same dataset, Singh et al.

(2019) conducted another study to investigate how the frequency of experiencing phishing emails during training affected detection in a later testing phase. Participants completed three phases in a phishing detection task: pre-training, training, and post-training, where participants were trained on different frequencies of phishing emails (25%, 50%, or 75%) and tested before and after training with 20% frequency of phishing emails. The results showed that participants that saw a larger proportion of phishing emails during training had higher hit rates but also higher false alarm rates in detecting phishing emails. This, in addition to the similarity between the benign ham emails and the phishing emails, can explain the bias to not respond to emails in the Rajivan and Gonzalez task. In the future, we will adapt the cognitive model to the task performed in Singh et al. to test other predictions of the IBL model, given that frequency of instances is one of the driving cognitive factors that influence decision making. A similar line of research will explore the model's ability to predict end-user behavior in situations where the statistics of the environment are more similar to that in the real world (e.g., where a very small proportion of emails are phishing emails).

Conclusions

In this paper we demonstrated that a cognitive model of end-user detection of phishing emails can be useful for understanding human susceptibility to phishing attacks. As the Alice scenario showed, normally safe end-users can get caught performing unsafe actions under the right set of circumstances. Human decisions are constrained by cognitive mechanisms (e.g., memory, spreading activation, and pattern matching) that reflect the statistics and dynamics of the environment. By manipulating that environment, new patterns can arise that change well-established user behavior.

The IBL model developed here is a first attempt to model phishing detection using ACT-R, and captures the cognitive mechanisms and biases that could give rise to unsafe actions. It is also a first step toward developing a cognitive model that predicts human performance based on the similarity of emails confronted. According to IBLT (Gonzalez et al., 2003), decisions are based on the similarity of the current email to past emails for which the user clicked links, the recency of those past emails, and the frequency of phishing emails in comparison to benign emails. The model performed similarly to the actions taken by humans, neither the model nor humans were highly accurate in classifying phishing emails. The nature of the task made classification difficult for both. Future research will investigate the various cognitive aspects that influence classification decisions, and improve the context representation in the model to reflect the relevant features for phishing detection. A cognitive model that is highly accurate at predicting end-user susceptibility to phishing attacks can greatly enhance current cybersecurity practice.

Acknowledgments

This research was sponsored by the Army Research Office and accomplished under Grant Number W911NF-17-1-0370.

References

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Canfield, C. I., Fischhoff, B., & Davis, A. (2016). Quantifying phishing susceptibility for detection and behavior decisions. *Human factors, 58*(8), 1158-1172.

Curtis, S. R., Rajivan, P., Jones, D. N., & Gonzalez, C. (2018). Phishing attempts among the dark triad: Patterns of attack and vulnerability. *Computers in Human Behavior, 87*(2018), 174-182.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science, 27*(4), 591-635.

Han, L., Kashyap, A. L., Finin, T., Mayfield, J., & Weese, J. (2013). UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the 2nd JCLCS* (pp. 44-52). Atlanta, GA.

Jensen, M. L., Dinger, M., Wright, R. T., & Thatcher, J. B. (2017). Training to mitigate phishing attacks using mindfulness techniques. *Journal of Management Information Systems, 34*(2), 597–626.

Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM, 50*(10), 94–100

Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M. A., & Pham, T. (2009). School of phish: a real-world evaluation of anti-phishing training. In *Proceedings of the 5th Symposium on Usable Privacy and Security* (p. 3). Mountain View, CA.

Lebiere, C. (1999). A blending process for aggregate retrievals. In *Proceedings of the 6th ACT-R Workshop*. George Mason University, Fairfax, Va.

Marchal, S., François, J., State, R., & Engel, T. (2014). Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management, 11*(4), 458-471.

Peng, T., Harris, I., & Sawa, Y. (2018). Detecting phishing attacks using natural language processing and machine learning. In *Proceedings of the IEEE 12th international conference on semantic computing* (pp. 300–301).

Prakash, P., Kumar, M., Kompella, R. R., & Gupta, M. (2010). Phishnet: predictive blacklisting to detect phishing attacks. In *2010 Proceedings IEEE INFOCOM* (pp. 1-5). San Diego, CA.

Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez C. (2019). Training to detect phishing emails: Effect of the frequency of experienced phishing emails. In *Proceeding of the 63rd International Annual Meeting of the HFES.* Seattle, WA.

Rajivan, P., & Gonzalez, C. (2018). Creative Persuasion: A Study on Adversarial Behaviors and Strategies in Phishing Attacks. *Frontiers in Psychology, 9*(135), 1-14.

Wombat Security (2018). *Beyond the Phish Report.* Retrieved from https://www.wombatsecurity.com/beyond-the-phish-2018. Proofpoint Inc.

Vishwanath, A., Harrison, B., & Ng, Y. J. (2018). Suspicion, Cognition, and Automaticity Model of Phishing Susceptibility. *Comm. Research, 45*(8), 1146–1166.

Towards Personalized Deceptive Signaling for Cyber Defense Using Cognitive Models

Edward A. Cranford (cranford@cmu.edu)[1], Cleotilde Gonzalez (coty@cmu.edu)[2], Palvi Aggarwal (palvia@andrew.cmu.edu)[2], Sarah Cooney (cooneys@usc.edu)[3], Milind Tambe (tambe@usc.edu)[3], and Christian Lebiere (cl@cmu.edu)[1]

[1]Department of Psychology, Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA
[2]Dynamic Decision Making Laboratory, Social and Decision Sciences Department
Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA
[3]USC Center for AI in Society, University of Southern California, 941 Bloom Walk
Los Angeles, CA 90089 USA

Abstract

Recent research in cybersecurity has begun to develop active defense strategies using game-theoretic optimization of the allocation of limited defenses combined with deceptive signaling. While effective, the algorithms are optimized against perfectly rational adversaries. In a laboratory experiment, we pit humans against the defense algorithm in an online game designed to simulate an insider attack scenario. Humans attack far more often than predicted under perfect rationality. Optimizing against human bounded rationality is vitally important. We propose a cognitive model based on instance-based learning theory and built in ACT-R that accurately predicts human performance and biases in the game. We show that the algorithm does not defend well, largely due to its static nature and lack of adaptation to the particular individual's actions. Thus, we propose an adaptive method of signaling that uses the cognitive model to trace an individual's experience in real time, in order to optimize defenses. We discuss the results and implications of personalized defense.

Keywords: cyber deception; cognitive models; instance-based learning; knowledge-tracing; model-tracing

Introduction

Cybersecurity often involves passive defense strategies which fail to discover a threat before major damage is done to a network. However, recent work within the domain of cybersecurity has focused on developing active defense strategies based on cognitive principles of deception (Al-Shaer et al., 2019; Cooney et al., 2019; Cranford et al., 2018). Deception is a form of persuasion where one intentionally misleads an agent into a false belief, in order to gain an advantage over the agent and achieve one's goals (Rowe & Rushi, 2016). In this line of research, the goal for security is to assist human administrators defend networks from cyber-attacks (Gonzalez et al., 2014). Limited defense resources cannot simultaneously protect all targets. In the event of an attack, truthful signals that divulge the protection status of a target can deter some attacks on protected targets. However, defenders can use a combination of truthful and deceptive signals to improve protection of the unprotected resources.

Game-theoretic principles have been employed to optimize the allocation of limited defense resources and determine how often to send a deceptive signal before it loses its effectiveness (Xu et al., 2015). While deception may reduce attacks on uncovered targets compared to no deception, the algorithms are static and tailored to an entire population. They fail to take into account the individual and their particular set of knowledge, experiences, and biases. The goal of this paper is to develop a personalized signaling strategy that can outperform traditional static methods.

Cranford et al. (2018) developed an instance-based learning (IBL) cognitive model (Gonzalez, Lerch & Lebiere, 2003) of attackers that accurately predicts human decision making from experience. We propose that such a model can be used to trace an individual's knowledge and experiences, and exploit their biases, to determine on-the-fly the best signal given the situation, to further reduce attacks.

The following section presents a line of research on game-theoretic models that have proven to optimize deceptive signaling for perfectly rational adversaries, and initial efforts toward optimizing for boundedly rational adversaries. We then describe an online game developed to investigate attacker behavior against deceptive signaling algorithms and a cognitive model that accurately predicts human behavior. Next, we describe a method for deceptive signaling that uses the cognitive model to drive adaptive signaling, personalized to the individual attacker. We highlight its applicability for optimizing defense by tracking human knowledge, experience, and biases. Finally, we discuss the implications of this line of research and avenues for future research.

Deceptive Signaling for Cybersecurity

Research on Stackelberg Security Games (SSGs) led to the development of algorithms that have greatly improved physical security systems (e.g., protecting ports, scheduling air marshals, and mitigating poachers) through the optimal allocation of limited defense resources (Pita et al., 2008; Shieh et al., 2012; Sinha et al., 2018; Tambe, 2011). Xu et al. (2015) extended these models by incorporating elements of *signaling*, in which a defender (sender) strategically reveals information about their strategy to the attacker (receiver) in order to influence the attacker's decision making (Battigalli, 2006; Cho & Kreps, 1987). Their solution, the Strong Stackelberg Equilibrium with Persuasion (peSSE), improves defender utility against a perfectly rational attacker compared to strategies that do not use signaling. For a given target, the peSSE finds the optimal combination of bluffing (sending a deceptive message that the target is covered when it is not) and truth-telling (sending a truthful message that the target is covered) so the attacker continues to believe the bluff.

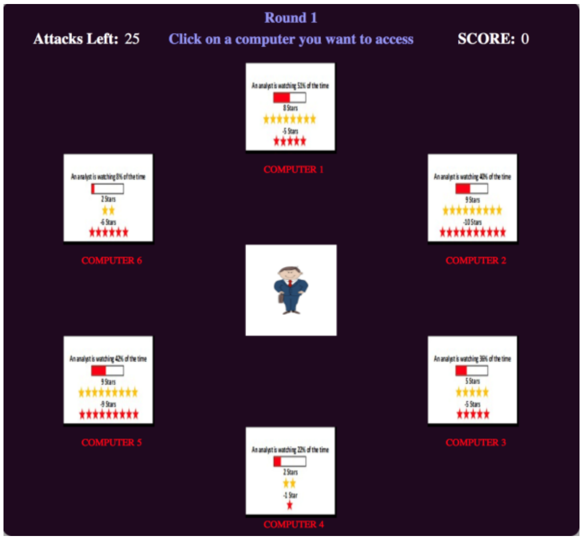
The goal of the peSSE is to reduce attacks on uncovered targets. Attackers earn a reward for successful attacks, suffer a loss for failed attacks, and earn zero for withdrawing. When a target is covered, the peSSE will always send a truthful signal. When uncovered, the peSSE will send a deceptive signal with a probability that brings the attacker's expected value of attacking, given a signal, to zero. This makes it equal to the utility of withdrawing the attack and, based on standard game-theoretic assumptions of perfect rationality, the attacker will break ties in favor of the defender and withdraw.

The peSSE is suitable for cyber defense where optimizing the probability of sending a deceptive signal can mitigate attacks on uncovered targets with little overhead. However, it is based on the assumption of perfect rationality while humans exhibit, at best, bounded rationality (Simon, 1956). To address this weakness of the peSSE, researchers have begun to develop signaling algorithms for security against boundedly rational attackers (Cooney et al., 2019). However, these algorithms do not offer substantial improvement over the peSSE in terms of reducing attacks and minimizing defender loss.

In what follows, we present an IBL cognitive model that accurately predicts human attacker behavior playing against the peSSE in a laboratory experiment. We propose that a personalized deceptive signaling scheme based on insights from the IBL model, in combination with model-tracing mechanisms, can be used to adapt defense signaling to the individual experiences of attackers at each point in time.

Cognitive Models of Human Attackers Playing Against Deceptive Signaling Algorithms

The Insider Attack Game (IAG) was designed to investigate the interaction between an attacker and defender in a cybersecurity scenario (Cranford et al., 2018). As shown in Figure 1, players take the role of the attacker (a company employee) and their goal is to score points by "hacking" computers to steal proprietary data. There are six potential computers to attack, but only two security analysts (defenders controlled by a computer algorithm) that can monitor one computer each. If the player attacks a computer that is monitored, they lose points, but if the computer is not monitored then they win points. Each computer shows its reward for winning, penalty for losing, and the probability that the computer is being monitored (reflecting the SSE for the game). On each turn, the player must select a computer to attack; after which, the signaling algorithm determines whether to send a truthful signal or a deceptive signal (with the signal, the player is presented the probability that the given signal is deceptive). The player must decide whether to continue their attack or withdraw and earn zero points. Players play four rounds of 25 trials each (after an initial 5 trials of practice). The payoff structures and monitoring probabilities of the targets are different in each round. Coverage and signaling of targets were precomputed for each trial. Therefore, each individual player experiences the same coverage and signaling schedule.



Round 1

Attacks Left: 25 Click on a computer you want to access SCORE: 0

An analyst is watching 51% of the time
8 Stars
★★★★★★★★
-5 Stars
★★★★★
COMPUTER 1

An analyst is watching 8% of the time
2 Stars
★★
-6 Stars
★★★★★★
COMPUTER 6

An analyst is watching 40% of the time
9 Stars
★★★★★★★★★
-10 Stars
★★★★★★★★★★
COMPUTER 2

An analyst is watching 42% of the time
9 Stars
★★★★★★★★★
-9 Stars
★★★★★★★★★
COMPUTER 5

An analyst is watching 36% of the time
5 Stars
★★★★★
-5 Stars
★★★★★
COMPUTER 3

An analyst is watching 22% of the time
2 Stars
★★
-1 Star
★
COMPUTER 4

Figure1: Screenshot of the IAG. The attacker is in the center surrounded by six targets. The monitoring probability is displayed as a percentage in text and represented visually by red bars, the yellow stars represent the potential reward, and the red stars represent the potential penalty.

Attacker Cognitive Model

Cranford et al. (2018) developed an IBL cognitive model of the attacker using the ACT-R cognitive architecture (Anderson & Lebiere, 1998; Anderson et al, 2004). Following collection of human data in the peSSE condition, we modified this model to better represent human behavior playing the IAG. For brevity, details of the model described below, and its underlying equations, can be found in Cranford et al., while specific changes are footnoted.

In the current model, decisions are made by generalizing across past experiences, or instances, that are similar to the present situation. For the IAG, instances include slots to represent the context of the selected target, the decision, and the outcome. The context includes the monitoring probability [0.0, 1.0], reward [1, 10], penalty values [-1, -10], and warning signal [present, absent]. The possible decisions are attack or withdraw, and the outcome is the reward or penalty based on the decision. In a given situation, for each possible decision, an associated utility is computed through blended memory retrieval weighted by contextual similarity to past instances. The decision with the highest utility is made. In the present game there are two decisions: attack or withdraw. However, withdrawing always results in zero points. Therefore, the model only needs to determine the utility of attacking in order to make a choice.

In ACT-R, the retrieval of past instances is based on the activation strength of the relevant instance in memory and its similarity to the current context. The activation of an instance reflects the power law of practice and forgetting, and includes a partial matching process[1] reflecting the similarity between the current context elements and the corresponding context elements for the instance in memory. A variance parameter *s* introduces stochasticity in retrieval. Similarities between numeric slot values are computed on a linear scale from 0.0, an exact match, to -1.0. Symbolic values are either an exact match or maximally different, -2.5, to prevent bleeding between memories for different actions and signal types.

A Boltzmann softmax equation[2] determines the probability of retrieving an instance based on its activation strength. The IBL model uses ACT-R's blending mechanism (Lebiere, 1999; Gonzalez et al., 2003) to calculate an expected outcome of attacking a target based on similarity to past instances. The expected outcome is the value that best satisfies the constraints of all matching instances weighted by their probability of retrieval.

In summary, the outcomes of past instances are weighted by their recency, frequency, and similarity to the current instance to produce an expected outcome. If the value is greater than zero then the model attacks, else it withdraws.

IBL Model Procedure To begin the IAG, the model is initialized with seven instances[3]: five represent a simulated practice round, and two represent knowledge gained from instructions (one instance had a signal value of *absent* and an outcome of 10, representing that attacking when a signal is absent will result in a reward; another instance had signal value of *present* and an outcome of 5, representing that attacking when a signal is present could result in either a penalty *or* a reward). On a given trial, the model first selects a target to attack. The model cycles through each target and generates an expected outcome of attacking via blending. The model selects the target with the highest expected outcome. Target selection is a passive process; therefore, no instances are saved in memory that could influence future decisions.

After selecting a target, the context is augmented with the value of the signal[4] (i.e., present or absent). The model then decides whether to attack or withdraw by generating an expected outcome via blended retrieval. The similarity of the selected target's context to past instances is based solely on the value of the signal[5] (monitoring probability, reward, and penalty values are ignored). In the IAG, the pop-up warning message covers all information about the selected target. Therefore, we inferred that humans base their decisions only on the value of the signal and ignore, or forget, the occluded target information.

After determining the expected outcome, an instance is saved in memory that represents the model's expectations[6]. Humans tend to remember not only the actual experience, but also their expectations prior to the experience (Gonzalez et al., 2003). This serves as an implementation of confirmation bias, in which one's preconception of winning/losing can increase the likelihood of attacking/withdrawing on future trials (i.e., generating positive/negative expected outcomes).

After generating an expected outcome, a decision is made, and the action and outcome slots of the current instance are updated to reflect the action taken by the model and the ground-truth outcome. This final instance is saved in memory and thereby influences future decisions.

The model continues for four rounds of 25 trials each. The model behavior reflects its experiences. If an action results in a positive/negative outcome, then its future expectations will be increased/decreased, and the model will be more/less likely to select and attack that target in the future. Also, the impact of a particular past experience on future decisions strengthens with frequency and weakens with time.

IBL Model Evaluation Against Human Players

The attacker IBL model was compared to human behavior in the IAG. In a laboratory experiment, human participants (i.e., "attackers") played against the peSSE signaling scheme. Participants were recruited via Amazon Mechanical Turk. All participants resided in the United States. For completing the experiment and submitting a completion code, participants were paid $1 plus $0.02 per point earned in the game, up to a maximum of $5.50. Four participants were removed from analysis because they had incomplete data (e.g., data recording errors) or restarted the experiment after gaining experience, resulting in a final sample size of 100.

The data was analyzed for the probability of attack and the number of points earned by attackers across rounds. The probability of attack was calculated as the proportion of players that continued the attack on a given trial. Points were separated into mean losses and gains per round. Losses/gains were calculated as the total number of points lost/gained per round by attacking targets that were/weren't monitored.

The model played the IAG 1000 times to generate stable predictions of the probability of attack and total number of points obtained per round. At the end of each run, the model was reset to its initial state and its memory cleared. Due to the stochastic nature of the model, and the influence its experiences have on its future decisions, the model behaves differently on each run and can therefore represent a diverse population of human attackers without the need to parameterize for individual differences.

[1] The mismatch penalty parameter for the activation equation was originally set high at 2.5, but was reduced to the ACT-R default 1.0.

[2] The temperature parameter was changed from the ACT-R default of $\sqrt{2} * s$ to a neutral value of 1.0 which results in retrieval probability reflecting the original presentation probability.

[3] The model was originally initialized with 8 instances representing edges of the decision space, but we believe the current method is a more accurate representation of participants' experience.

[4] Representing the deception probability as an additional context slot in the instance resulted in a poorer model fit. It appears that humans do not consider, or know how to utilize, the information. Therefore, the deception probability was excluded from the context.

[5] In the original model, the full context was used, but this resulted in an over-selection and reduced attack rate of high-reward targets.

[6] The model did not originally save this instance in memory and attacked far less often than humans. Saving this instance increased the mean probability of attack. This insight was key to understanding the biases humans have in the game and why they attacked so often.

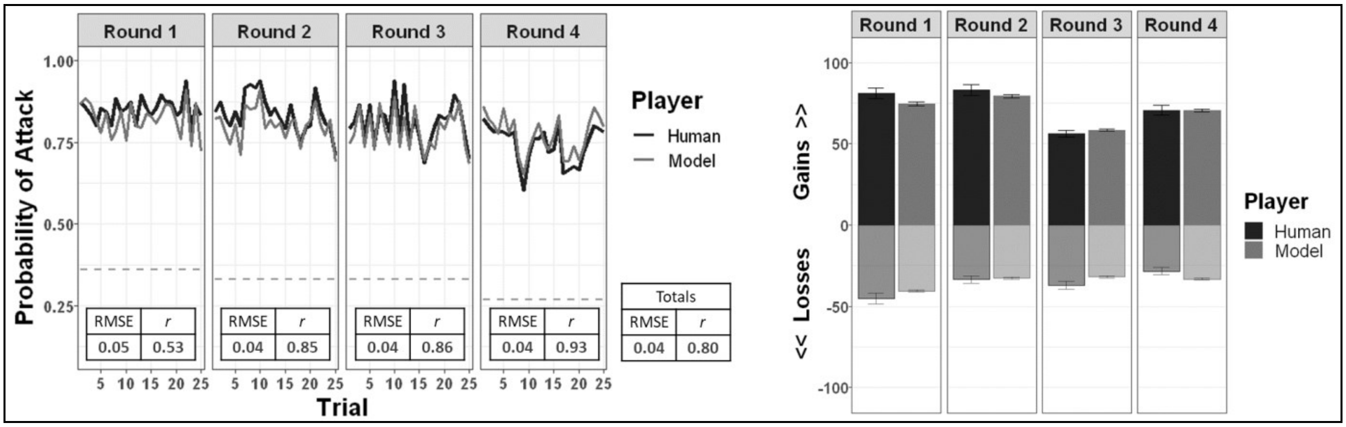


Figure 2: Probability of attack across trials and rounds (left side) and mean gains/losses per round (right side) for the humans compared to the IBL model. For probability of attack, RMSE and correlations (*r*) between human and model data are displayed under each round, and the aggregate values across the entire game are on the right under the legend.

Figure 2, left side, shows the mean probability of attack across trials and rounds for humans compared to the model. The dashed, gray line represents the peSSE predictions under assumptions of perfect rationality. Humans attack far more than perfectly rational attackers. Meanwhile, compared to the human data, the model is an excellent predictor of performance. RMSE and correlations, comparing the model to human data, are included at the bottom of the graph. The model is sensitive to the schedule of coverage, just as humans are, which produces the spiking pattern across trials.

Figure 2, right side, shows the average gains/losses for the humans compared to the model. Humans attack at a high rate, earning many points from attacks on uncovered targets, while incurring fewer losses. Moreover, the model accurately predicts this behavior. The peSSE suffers because human biases (e.g., recency, frequency, and confirmation) lead them to attack at higher rate, resulting in more experiences of wins than losses. The IBL model captures these biases, and therefore, can feasibly be used as a predictive tool for personalizing deceptive signals for an individual attacker. Notably, the model has accurately predicted human performance against other signaling algorithms (not reported here) prior to collection of human data.

Toward Personalized Deception

To personalize deception, we can run the IBL model alongside the human to predict an individual's behavior and optimize the rate of deceptive signals. To make accurate predictions of an individual, two methods have proven useful to align the model behavior with the human's decisions: model-tracing and knowledge-tracing. Model-tracing aligns the model's *actual* actions and outcomes to those *observed* of the human. Knowledge-tracing aligns the *expected* actions and outcomes to match those *inferred* of the human.

Model-tracing

Model-tracing is a method used to align a model's behavior with that of the human and is commonly used to adjust feedback provided to the student in intelligent tutoring systems (see Anderson et al., 1995). The alignment helps in a way that future model predictions are adapted and optimized to the interaction with the human. For example, geometry tutors use model-tracing to keep track of where errors are made so that the learning experience can be tailored to the individual (Anderson, Boyle, & Yost, 1986).

We use model-tracing to synchronize the IBL model with the human's *observed* actions and experience in the IAG task. After each trial, the instance saved in memory that represents the model's decision and outcome is changed to reflect the human's action and outcome (i.e., the action and outcome slots are changed to match the human's). Therefore, on the next trial the model makes predictions based on the exact experience of the human and not on what it would have done based on its own past instances. With more trials, the model is expected to make more accurate predictions of a particular human's actions, as the model's memory aligns better with that of the human. Model-tracing changes the instances representing the *observed* ground truth decision and outcome. However, in order to generate accurate predictions, we must also align the model's expectations to those of the human.

Knowledge-tracing

The model produces instances that represent the expected outcome of attacking, which contributes to confirmation bias, and these must also be changed. *Knowledge-tracing* can be used to *infer* the expectations humans had prior to making a decision that would contribute to confirmation bias. For example, if the model and human both decided to attack (or both withdraw), then nothing need change and the expected outcome generated by the model can be used to infer the human's expectation. However, if the model expects a positive outcome for attacking, but the human withdrew the attack, then we can *infer* that the human expected to lose (or vice versa). For these instances, we can modify the expected outcome slot to match the expectations of the player. We cannot infer this expectation precisely, so we set the expected outcome to either the reward or penalty of the selected target.

Model Predictions with Model & Knowledge-tracing

To test the effectiveness of model- and knowledge-tracing for predicting human decision making, the model was run alongside human data in the peSSE condition. On each trial, the model simply makes a decision, which is recorded, and is then updated via model-tracing and knowledge-tracing. The model decision was then compared to the decision the human made to generate a probability of agreement between the model and human. The mean probability of agreement for rounds 1-4 are 86.4% (*SD* = 12.3%), 90.8% (*SD* = 11.4%), 89.6% (*SD* = 12.4 %), and 86.8% (*SD* = 15.5%), respectively.

The trial-to-trial agreement is highly accurate, just short of accounting for the entirety of human stochasticity. In fact, even at the 1st trial the model is accurate to 83.3%. Moreover, the model adapts well to the individual's probability of attack. Figure 3 shows the overall probability of attack of individual model runs compared to the human it traced. The model is exceptionally accurate in adapting to the human, $r^2 = 0.95$. Using techniques of model-tracing and knowledge-tracing, the model makes very accurate predictions and could feasibly be used in designing a personalized signaling scheme.

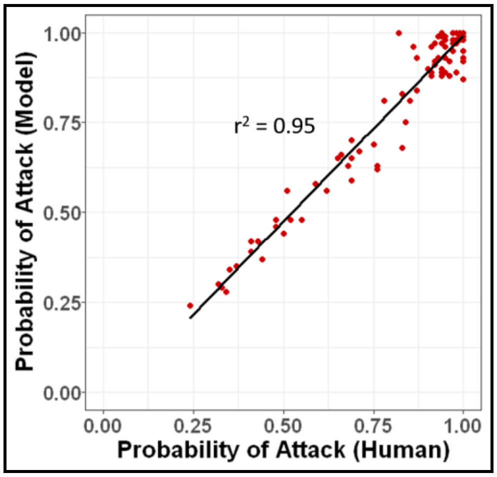


Figure 3: Overall mean probability of attack comparing individual humans to the model run that traced him/her, in the peSSE condition using personalized signaling.

A Personalized Deceptive Signaling Scheme

The peSSE signaling scheme uses deceptive signals on uncovered targets but not on covered targets. These schemes invite attacks with impunity when no signal is given. Therefore, a broader and more symmetrical approach may be warranted, as has been explored in recent game-theoretic research (Cooney et al., 2019). The following signaling scheme also uses deceptive signals when a target is covered.

If the goal is to minimize the probability of attack as a function of the warning signal then it can be shown that we must reach an equilibrium where the probability of attack given a warning, P(A|W), is equal to the probability of attack given no warning, P(A|NW). A signal must be generated at a rate that preserves this equality. We can examine the impact of the presence or absence of a signal in various situations. For example, given an attack, if a target is covered, the attacker will lose, and their future probability of attack will be lower. If a target is uncovered, the attacker will win, and their future probability of attack will be higher. Each outcome thus increases or decreases one of the attack probabilities. In particular, the change in attack probability (decrease or increase) is determined by whether the selected target is covered or not, respectively, while the probability impacted (signal or no signal) is determined by the presence or absence of a signal, respectively. This results in the following algorithm for signal generation: if the selected target is covered, if P(A|W) is greater than the P(A|NW) then generate a signal, otherwise do not generate a signal; but if the selected target is not covered, if P(A|W) is greater than the P(A|NW) then do not generate a signal, otherwise generate a signal.

The role of the cognitive model in this algorithm is to determine P(A|W) and P(A|NW). We know the model generates expected outcomes of attacking and decides to attack if the value is greater than zero. Therefore, we can simply generate the expected outcome of attacking given the presence or absence of a signal and compare them to compute the conditions used in the algorithm above. An essential point is that those expected values are not the true expected values, but the model's subjective expected value given its limited experience and its reflection of human cognitive biases.

Intuitively, if the selected target is covered, then we decide on whether to generate a signal or not depending on which condition is most likely to lead to an attack. This corresponds to trying to catch the attacker when the target is covered, lowering the future probability of attack. Conversely, if the selected target is not covered, select the condition (signal or not) least likely to lead to an attack. Again, the accuracy of the cognitive model is essential in this approach to capture the subject's intention to attack or not. We can use the current model to track an individual's decisions and generate predictions of their probability of attack given the situation.

Effectiveness of Personalized Signaling Scheme

To generate predictions of the effectiveness of this personalized signaling scheme, we ran the IBL model through the IAG while using the personalized signaling scheme described above to make predictions about the expected outcome of attacking, given a signal and given no signal. Based on those predictions and the underlying coverage of the selected target, the scheme determined whether or not to give a signal on each trial.

Compared to the human performance in peSSE, the personalized signaling method is expected to reduce the probability of attack by an average of 2.7% (RMSE = 6.6%). Meanwhile, Figure 4 shows that personalized signaling will result in fewer gains and more losses. Looking further into the data, Figure 5 plots the probability of attack across the various targets, based on their monitoring probability. Compared to human performance, the personalized signaling method seems to shift the distribution of attacking toward targets with a higher monitoring probability, and therefore the IBL model incurs more penalties.

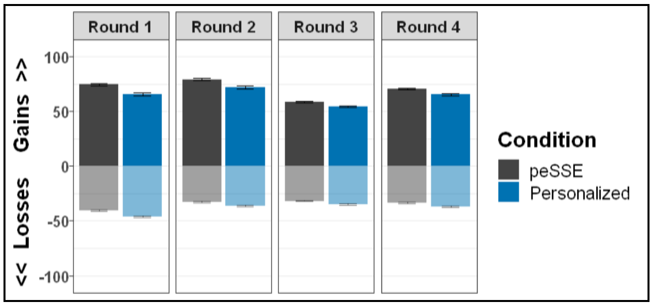


Figure 4: Comparing the mean gains/losses across rounds in the personalized signaling model to humans in the peSSE.

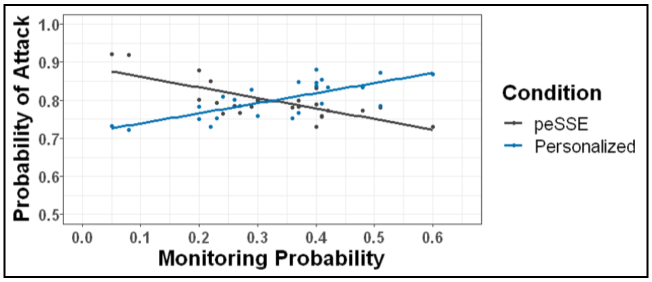


Figure 5: Mean probability of attack across targets, by their monitoring probability, comparing the personalized signaling model to humans in the peSSE condition.

Conclusions

The present research shows that we can leverage the predictive power of a generalizable IBL model to infer an individual's knowledge, trace their experience, and exploit their biases to design an adaptive signaling scheme that is personalized for an individual. The current method is an initial attempt toward developing a personalized deceptive signaling scheme for cyber defense. Although the current scheme did not greatly reduce the probability of attack, the cognitive model proved to be an accurate predictor of human behavior. Future research will test the personalized signaling scheme against human attackers. Insight gained from human experiments will provide information about how to modify the signaling logic to create a more effective scheme.

Acknowledgments

This research was sponsored by the Army Research Office and accomplished under Grant Number W911NF-17-1-0370.

References

Al-Shaer, E., Wei, J., Hamlen, K. W., & Wang, C. (2019). Dynamic Bayesian Games for Adversarial and Defensive Cyber Deception. In *Autonomous Cyber Deception*. Cham: Springer.

Anderson, J. R., Boyle, C. F., & Yost, G. (1986). The geometry tutor. *Journal of Mathematical Behavior*, 5-20.

Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of Learning Sciences*, *4*, 167-207.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-1060.

Battigalli, P. (2006). Rationalization in signaling games: Theory and applications. *International Game Theory Review 8,* 01, 67–93.

Cranford, E. A., Lebiere, C., Gonzalez, C., Cooney, S., Vayanos, P., & Tambe, M. (2018). Learning about cyber deception through simulations: Predictions of human decision making with deceptive signals in Stackelberg Security Games. *Proceedings of the 40th annual conference of the Cognitive Science Society* (pp.258-263). Madison, WI: Cognitive Science Society.

Cho, I.-K. & Kreps, D. M. (1987). Signaling games and stable equilibria. *The Quarterly Journal of Economics, 102*(2), 179–221.

Cooney, S., Vayanos, P., Nguyen, T. H., Gonzalez, C., Lebiere, C., Cranford, E. A., & Tambe, M. (2019). Warning time: optimizing strategic signaling for security against boundedly rational adversaries. *Proceedings of the 18th AAMAS.* Montreal, Canada: IFAAMS.

Gonzalez, C., Ben-Asher, N., Oltramari, A., Lebiere, C. (2014). Cognition and Technology. In Kott, C., Wang, A. & R. Erbacher (eds.), *Cyber defense and situational awareness*. Switzerland: Springer International Publishing.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science, 27*(4), 591-635.

Lebiere, C. (1999). A blending process for aggregate retrievals. *Proceedings of the 6th ACT-R Workshop*. George Mason University, Fairfax, Va.

Pita, J., Jain, M., Ordónez, F., Portway, C., Tambe, M., Western, C., & Kraus, S. (2008). ARMOR Security for Los Angeles International Airport. *Proceeding of the 23rd AAAI conference on artificial intelligence* (pp. 1884-1885). Menlo Park, CA: AAAI Press.

Rowe, N. C., & Rrushi, J. (2016). *Introduction to Cyberdeception*. Switzerland: Springer.

Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., & Meyer, G. (2012). Protect: A deployed game theoretic system to protect the ports of the United States. *Proceedings of the 11th AAMAS* (pp. 13-20). IFAAMS.

Simon, H. A. (1956). Rational choice and the structure of the environment. Psychological Review, 63(2), 129-138.

Sinha, A., Fang, F., An, B., Kiekintveld, C., & Tambe, M. (2018). Stackelberg Security games: Looking beyond a decade of success. *Proceedings of the 27th IJCAI* (pp. 5494–5501). IJCAI.

Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned.* Cambridge University Press.

Xu, H., Rabinovich, Z., Dughmi, S., & Tambe, M. (2015). Exploring information asymmetry in two-stage security games. *Proceedings of the National Conference on Artificial Intelligence* (2, pp. 1057-1063). Elsevier B.V.

A Study on Teamwork in a Dynamic Task

Cvetomir M. Dimov (cdimov@andrew.cmu.edu)
John R. Anderson (ja@cmu.edu)
Shawn A. Betts (sabetts@andrew.cmu.edu)
Dan Bothell (db30@andrew.cmu.edu)
Department of Psychology, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Abstract

Skill acquisition experiments have rarely focused on collaborative tasks. Here we attempt to fill this gap with a study on teamwork in a dynamic task. The task - Coop Space Fortress - is computer game, in which subjects fly spaceships to destroy a space fortress. This task presents two challenges: learning how to fly a spaceship in a frictionless environment and developing a strategy on how to coordinate. When learning to play this computer game, subjects not only master the game controls but also typically settle on team roles to more efficiently achieve their goal, despite not being allowed to communicate. The data from this study will pave the way to an ACT-R model of teamwork in a dynamic task.

Keywords: skill acquisition, dynamic task, teamwork, Space Fortress, ACT-R

Introduction

From stumbling into our first steps, to learning a foreign language in middle school and our first mathematical analysis class at university, our lives are replete with various tasks that we master to different extents. It is astounding how skilled we can become after a sufficient amount of practice: the tightrope walker was once a toddler falling after a couple of steps; Shakespeare was once mumbling incomprehensible words and even Euler – the most prolific mathematician ever – was once unable to count.

This gradual shift of the unskilled becoming fully proficient has been characterized as proceeding in distinct phases. Specifically, Fitts (1964, Fitts & Posner, 1967) described motor skill as progressing through three phases: a Cognitive Phase, an Associative Phase, and an Autonomous Phase. Anderson (1982) also adopted the understanding that skills go through three phases and applied this to cognitive skills, whereby he modeled the successive periods of skills acquisition in the cognitive architecture ACT*. Others too have accommodated the idea that skill acquisition is a 3-phase process (e.g., Ackerman, 1988; Kim, Ritter, & Koubek, 2013; Rosenbaum, Carlson, & Gilmore, 2001).

Similarly to ACT*, in ACT-R (Anderson, 2007) – the current version of the architecture – skill transitions from a slower and more deliberative stage to a faster and more automatic stage. This architecture has been applied to model skill acquisition in a variety of tasks, such as solving linear equations (Anderson, 2005), a complex aviation task (Taatgen et al., 2008) and past tense learning (Taatgen & Anderson, 2002). Further support of ACT-R's characterization has been provided in neuroimaging studies, which uncovered qualitative changes in the recorded neural patterns as subjects become more proficient (i.e., in solving pyramid problems; Tenison, Fincham, & Anderson, 2016). Finally, a modification of the architecture was used to model an amount of transfer between skill acquisition tasks not completely accounted for by ACT-R (Taatgen, 2013).

Cognitive Skill Acquisition

ACT-R does not adhere to a 3-phase view of skill acquisition. Instead, this architecture models each subcomponent of the skill as transitioning from a declarative to a procedural endpoint. At one extreme, the cognitive system only has declarative knowledge about a certain task domain. This knowledge is typically stored in terms of operators in declarative memory, which are composed of three pieces of information: the state in which they apply, the action that should be taken, and the state that results after that action is taken. Operators are the building blocks of a subject's mental model of the task and they are typically acquired when reading the task instructions. When an unskilled subject faces a task, operators are retrieved to determine what action should be taken next.

These operators are gradually converted to procedural knowledge through a process called *production compilation* (Taatgen & Anderson, 2002). When an operator is compiled into a production, its actions are directly performed by that production without the need to retrieve the operator. The result is, first, a faster execution of that action as the time cost of retrieval is no longer incurred and, second, retrieval processes are no longer occupied and can be used for other purposes. Moreover, it is possible for two subsequent actions to be complied into a single action if there is no conflict of cognitive resources. The relative rate at which operators are compiled is a function of how often they are evoked, meaning that different subcomponents of skill can be proceduralized to different extents at certain time points.

Dynamic Tasks

The majority of skill acquisition tasks modeled with ACT-R follow a linear perception-cognition-action pattern. Yet, real-world tasks are complex and dynamic, meaning that they involve the coordination of cognitive, perceptual and motor activities in an ever-changing, yet predictable world. To investigate the applicability of ACT-R's approach to dynamic tasks, learning in the arcade game Space Fortress was addressed (Anderson et al., 2019). Space Fortress was selected because it is simple enough to be suitable for an experiment, challenging at first and still learnable within a single experimental session. In addition to its dynamic nature, Space Fortress differs from the majority of skill acquisition tasks modeled with ACT-R in that it requires learning to tune skill to features in the environment so that actions are successful. To this end, ACT-R was extended with a new module – the Controller (Anderson et al., 2019). The model of this dynamic task underwent the same process of skill acquisition that other models did by gradually compiling operators into productions. However, the increased complexity of this task relative to others meant that the model spent much more time compiling operators than simpler models do. Moreover, while operators in declarative memory were being proceduralized, the Controller module was tuning actions to relevant environmental features.

Teamwork

One aspect of skill acquisition that has been rarely researched in the lab is how people learn to execute a novel task while working as a team. In a team, individuals' tasks become interdependent and their goals shared (Dyer, 1984). To achieve high performance, each team member needs to successfully manage the tasks that are independent of the other team members (i.e., *taskwork*) and the tasks that are intertwined with the others (i.e., *teamwork*; Salas, Cooke, & Rosen, 2008). Both taskwork and teamwork depend on the processes of encoding, storage and retrieval of information, while two additional factors are key to teamwork. Specifically, *shared cognition* (i.e., shared mental models and situation awareness; Salas & Fiore, 2004) and communication facilitate coordination and cooperation between team members. To investigate how people learn to work in a team in a dynamic task, we created a new cooperative computer game, Coop Space Fortress.

Coop Space Fortress

Space Fortress has a history in the study of skill acquisition dating back to the end of the 80's (Donchin, 1989; Frederiksen & White, 1989; Gopher et al., 1989). The goal of the game is to accumulate as many points as possible, which can be achieved by destroying a fortress located in the center of the screen while avoiding crashing into a rectangle, which defines the playing field. We relied on the Pygame implementation of Space Fortress (Destefano & Gray, 2008) to create a cooperative version of the game – Coop Space Fortress.

In Coop Space Fortress, two players control two ships (see Figure 1). Their goal is to destroy a fortress in the center of the screen. However, the fortress has an impenetrable shield around it (the small hexagon), which is only partially disabled when the fortress shoots a missile. When this happens, the back of the fortress is no longer shielded and the fortress can be destroyed. Consequently, for the team to destroy the fortress, one ship needs act as a bait: it needs to enter the big hexagon, which triggers the fortress to aim and shoot at that ship if the ship moves sufficiently slowly. While the fortress is shooting and its back is exposed, the other ship needs to navigate behind the fortress, aim at it and destroy it (see Figure 2). To keep things simpler, we did not allow players to communicate. Thus, players needed to figure out their roles based solely on the common instructions that they received.

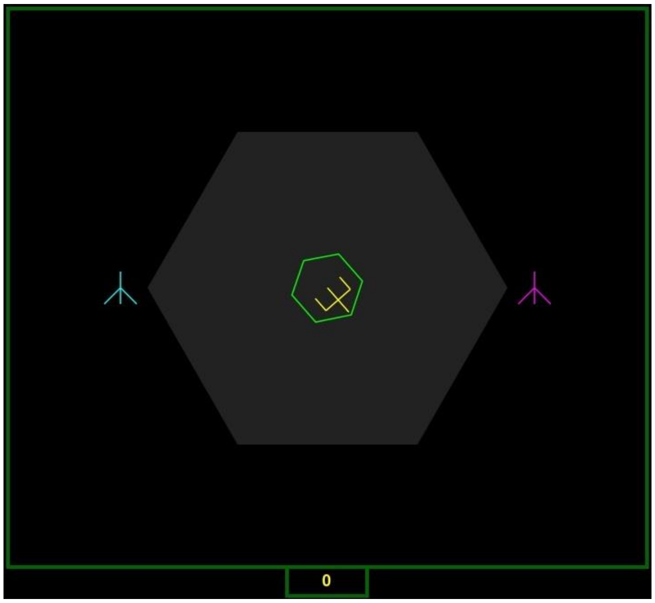


Figure 1: Start of the game. Both players are outside of the hexagon and the fortress has no target. The players should enter the hexagon and try to destroy the fortress.

When the fortress is destroyed, the score is incremented by 100 points. The ships need to then exit the hexagon. When both ships are outside of it, the fortress respawns and the ships can again attempt to destroy it. When outside the hexagon, the ships should avoid hitting the outer border (big square), because they would explode and reduce their common score by 100 points. In addition to penalizing deaths, reckless shooting is also penalized by 10 points for each missile that does not hit the fortress.

Navigation in Coop Space Fortress relies on three actions: rotating clockwise (key "D"), rotating counterclockwise ("A") and thrusting ("W"), while shooting is achieved with the spacebar. Despite having only 4 actions overall, learning to play Coop Space Fortress is a challenging task. A major difficulty is that frictionless space is counterintuitive to operate in. First, the ship's orientation is independent of its direction of flight. Moreover, the ship does not slow down on its own and no breaks are available. Instead, to slow down one needs to turn in a direction opposite the flight direction and thrust. Similarly, moving in a desired direction requires thrusting in a direction, whose vector sum with the flight velocity results in the desired flight path. Another challenge for players is learning how key press durations map to acceleration or rotation rate.

In addition to learning how to control the ship, Coop Space Fortress poses the additional challenge of coordinating with a teammate, because unless each player does their task, no player will earn any points. For example, the player that acts as a bait needs to stay inside the hexagon and fly at a slow speed while the shooter is aiming and shooting. Note that if the bait accidentally exits the hexagon or is shot down by the fortress, the shooter becomes the bait and both players need to reset their current goals. Similarly, if the shooter does not succeed in commanding the ship with enough proficiency to destroy the fortress, the team will perform poorly. Finally, both players need to exit the hexagon for the fortress to respawn. As a consequence, the final performance in the game is an interaction between the skills of each player: If even one player struggled to performs his/her task, the common score would remain low. On the other hand, if each player performed at a reasonable level, the common score would increase.

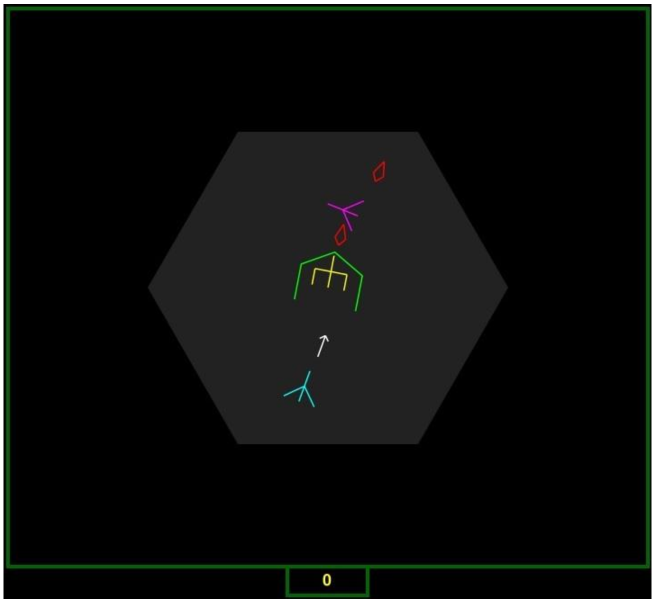


Figure 2: Players coordinating: one player acts as a bait, while the other is shooting at the fortress. The fortress, having shot at the first player, has its back exposed. Once the fortress is destroyed, the two players should exit the big hexagon so that the fortress respawns.

Methods

Participants

Thirty subjects (13 males, mean age: 22.4 years, min age: 18, max age: 35) from the Pittsburgh area, mostly students from Carnegie Mellon University and the University of Pittsburgh, participated for money, which included a base payment ($15) and a bonus payment (Mean: $2.03, Min: $0.10, Max: $11.85) based on their performance. Pairs of participants were formed either randomly, restricted by participants' availability, or by asking participants to bring another participant to play with. Informed consent approved by the Carnegie Mellon University Institutional Review Board (IRB) was obtained from each participant. The data of the first pair of subjects was not included in the analysis as it was not completely recorded. Only 4 of the 28 subjects reported having played a similar game (2 reported Asteroids, while Snake and Minecraft were each considered similar by a single subject each) in the post-experimental questionnaire.

Procedure

The experiment consisted of 4 tasks: (1) a demographics questionnaire, (2) game instructions, (3) 20 3-minute-long rounds of playing Coop Space Fortress, and (4) a feedback questionnaire. A task needed to be completed by both participants before the subsequent one could be started. Participants were given a 1-minute break after 10 games. The overall experiment took between 1h15min and 1h20min of participants' time.

Demographics questionnaire. This questionnaire consisted of general demographics questions and of game-related questions. The general demographics questions inquired about the subject's sex, age, ethnicity, and field of study. The game-related questions requested information about the subject's video game experience, such as whether they ever played or currently play video games, the frequency of play, the platform they played on and the preferred genre of video games.

Post-experimental questionnaire. The post experimental questionnaire elicited information about a subject's experience during the experiment. It inquired what difficulties subjects faced during gameplay, what strategies they attempted and what strategy they finally settled on.

Results

All pairs of participants exhibited learning in the course of the 20 games of Coop Space Fortress. Figure 3 shows the average, minimum and maximum points obtained by subject pairs over the course of the 20 games. On average, teams monotonically increased their performance as the experiment progressed. Yet, there was a substantial variability in the amount of points achieved in a game.

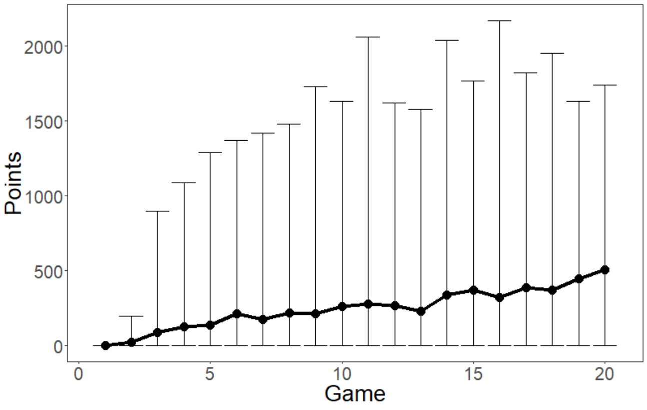


Figure 3: Game score progression over 20 3-minute games. Error bars represent minimum and maximum points

achieved in a game. The mean score increases steadily, but the variability is large.

A major reason for the large variability in score is the between-subject variability (Figure 4), which likely reflects prior experience with video games. Note that total points result from an interaction of the ability of both players: If one player is of low skill, the pair would not reach a high total score no matter how skilled the second player. The skewed distribution of average team score is likely a consequence of this interaction. A second major contributor to the large variability in score is the game-to-game variability within pairs of subjects (error bars in Figure 4).

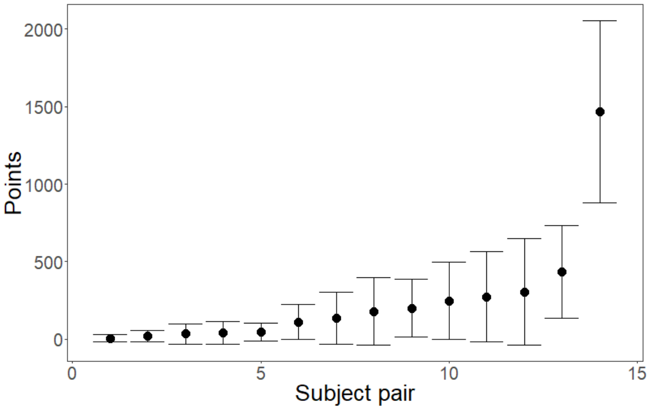


Figure 4: Average score per game for each of the 14 pairs of subjects. Error bars plot standard deviations. There is a large variability in skill between pairs of subjects.

Points are determined to a large extent by the number of fortress kills and number of player deaths. Not surprisingly, as the game progresses, players become better at destroying the fortress and less likely to die (see Figure 5).

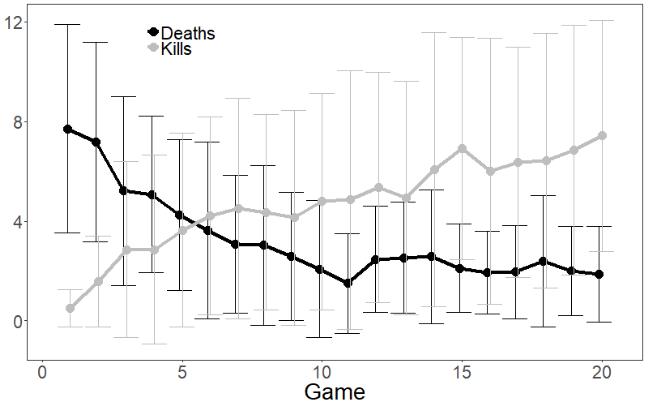


Figure 5: Average number of fortress kills and average number of deaths over 20 3-minute-games. Error bars represent standard deviations. Players progressively become better at killing the fortress and avoiding crashing into obstacles.

The primary cause of death in the beginning of the game is hitting the outer border of the game field (i.e., the large square), which reflects players' poor navigation abilities. As players become more skilled at controlling the ship in the frictionless environment, they also almost never hit the outer border and their total number of deaths decreases substantially.

Pairs of subjects differ significantly in skill. Where some subjects are highly skilled at coordinating their actions and aiming accurately while moving fast and, consequently, are able to achieve a lot of kills, other subjects' poor navigation skills force them to fly at a slow speed to avoid crashing into an obstacle, which leads them to reaching a lower number of kills per unit time. Moreover, these subjects are also typically worse at aiming and precisely navigating their ship to successfully coordinate with each-other.

Individual Skill Acquisition

Learning to navigate is a primary challenge in the frictionless environment. In addition to becoming better and pressing keys at durations that would lead to the intended ship state, players also learn to stay within reasonable ranges of their flight speed, because too high speeds easily lead to losing control over the ship and crashing (Figure 6).

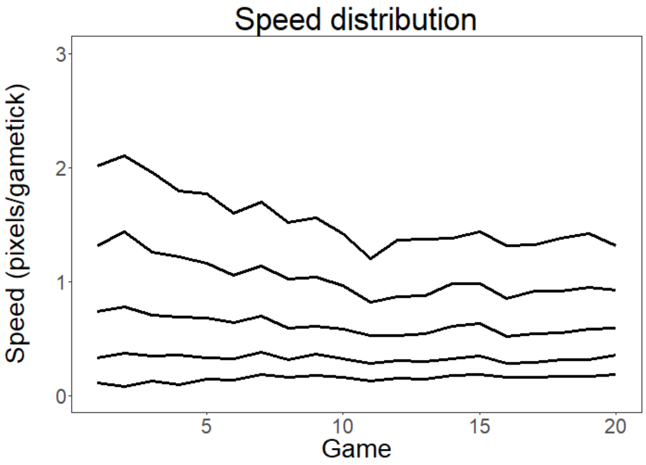


Figure 6: Speed distribution over 20 games. Players learn that excessive speed leads to loss of ship control and thrust less. The lines correspond to the 10th, 25th, 50th, 75th and 90th percentile.

Another key component of the game is the ability to destroy the fortress, which requires learning how to aim and when to shoot. In the initial games, players shoot more frequently overall (Figure 7a) and need many more shots to achieve a fortress kill (Figure 7b). As players become more skilled, they asymptote towards needing 2 shots per fortress kill on average and reduce their total number of shots overall, which further increases their point total as each missed shot leads to a 10-point penalty.

Cooperation Strategy

No matter how skilled at flying the ship, aiming and timing a shot, players still need to coordinate their action in order to achieve a high score in the game. An efficient strategy would exploit the strengths of each player and allow players to learn quickly. As indicated by in-game variables and subject reports, the majority of teams settled on a cooperation strategy that required each player to adopt a specialized role, whereby one player acted as a bait, while the second one as a shooter. Specifically, out of the 14 teams, in 9 at least 2/3 of all kills were committed by one player (see fraction of fortress kills per player in Figure 8).

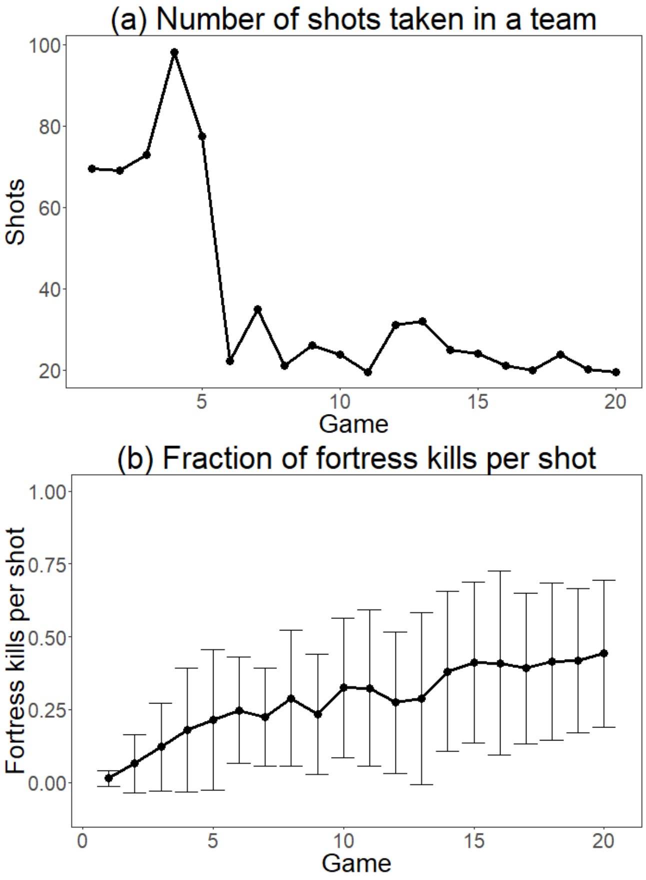


Figure 7: (a) Total number of shots and (b) fortress kills per shot over 20 games, averaged over 14 subject pairs. Error bars plot standard deviation. Teams become more efficient at destroying the fortress as the game progress.

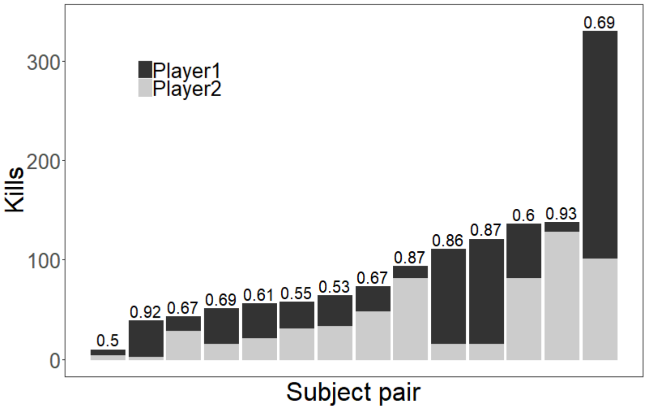


Figure 8: Total number of fortress kills for each player in each pair of subjects. The fraction of total fortress kills of the subject that committed more kills is displayed on top of each bar.

Role adoption did not happen immediately. While initially each player was, on average, equally likely to shoot down a fortress, role separation slowly started emerging. By games 5-6, the player acting as a shooter destroyed the fortress on average 70-75% of the time (see Figure 9). Note that there was a large inter-subject and inter-game variability, which was due to, first, the teams that did not adopt a role and, second, to poorly performing teams, for which the fraction of fortress kills varied more strongly.

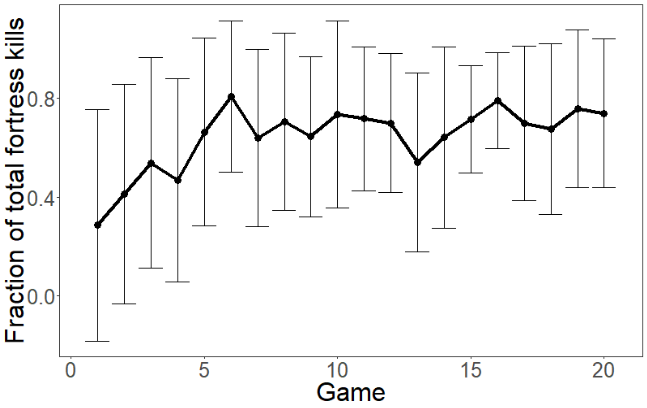


Figure 9: Average fraction of fortress kills per game of the player with more total kills over all 20 games. Error bars plot one standard deviation. As the games progress, players become more likely to adopt and stick to a role.

In their post-experimental reports, 9 of the 14 pairs of subjects reported purposely adopting a role, 4 pairs of subjects did not report this or reported purposely alternating roles, and the subjects in one pair had conflicting intentions – one player attempted to act as a bait, while the second did not adopt the role of the shooter. Instead, his strategy was to try to get any positive score and then try not to die by the end of the 3-minute game. Moreover, each team followed their idiosyncratic cooperation path. For example, out of fairness considerations, the skilled player in one pair reported intentionally taking turns in acting as a bait and as a shooter until realizing that it is more efficient to stick to the same role.

Most pairs did not report why they adopted their role. Of the 3 that did, for 2 the shooters were the players that were better at controlling the ship and for 1, the bait was the player better at controlling the ship. Interestingly, 7 of the 28 subjects also mentioned that one difficulty in playing the game was their inability to communicate with their partners, which they claimed would facilitate strategizing and role assignment.

Finally, independent of their role, many subjects also reported trying fly on the opposite side of the fortress than their teammate. Evidence for this could also be seen when observing player’s trajectories, which for some teams revealed that the teammates stayed in opposite quadrants of the playing field.

Discussion and Conclusion

We presented the results of an experiment that investigates how subjects acquire skill in a dynamic teamwork task. The task, Coop Space Fortress, is a modification of the dynamic game Space Fortress that requires pairs of subjects to cooperate in order to earn points. All pairs of subjects learned to play the game, although there were large inter-subject differences in ability. Subjects improved their game score both by becoming more skilled at controlling their ship and by typically settling on a role.

Why are subjects adopting distinct roles? Adopting roles likely simplifies skill acquisition, because it is easier to learn the actions associated with a single task as opposed to with two separate tasks. Moreover, it is likely more efficient, because there are no switching costs. Yet, how do subjects decide who should adopt what role, given that they are not allowed to communicate? As hinted by the post-experimental questionnaires, different roles might require a different amount of skill. Consequently, the more competent player should lean towards adopting the more difficult task. Interestingly, the three subject reports did not all agree on which role is more difficult. If subjects are equally skilled, random factors such as who happens to be targeted by the fortress first might turn the scales in one direction.

One way of exploring these questions more deeply would be to extend the existing ACT-R model of Space Fortress (Anderson et al., 2019), which captures individual skill acquisition, to include shared mental models. One component of shared mental models are the game instructions, which are represented as operators. Additionally, the model of each player needs to represent the past actions of the teammate, which would then enable it to infer the teammate's likely future actions. As suggested by Lebiere, Jengtsch, and Ososky (2013), one could rely on Instance-based Learning Theory (Gonzalez, Lerch, & Lebiere, 2003) to store instances of the teammate's past actions and their outcomes. This final model would then allow us to trace out the skill acquisition trajectory in this cooperative task to better understand how people learn to work in teams.

Acknowledgments

This work was supported by AFOSR/AFRL award FA9550-18-1-0251.

References

Ackerman, P. L. (1988). Determinants of individual differences during skill acquisition: Cognitive abilities and information processing. *Journal of Experimental Psychology: General, 117*, 288-318.

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89,* 369-406.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science, 29,* 313-341.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?*. Oxford University Press.

Anderson, J. R., Betts, S., Bothel, D., Hope, R., & Lebiere, C. (2019). *Transfer of rapid and precise action skills*. Manuscript submitted for publication.

Destefano, M., & Gray, W. D. (2008). *Progress report on Pygame Space Fortress*. Troy, NY: Rensselaer Polytechnic Institute.

Donchin, E. (1989). The learning-strategies project: Introductory remarks. *Acta Psychologica, 71,* 1-15.

Dyer, J. L. (1984). Team research and team training: A state of the art review. In F. A. Muckler (Ed.), *Human Factors Review* (pp. 285–323). Santa Monica, CA.

Fitts, P. M. (1964). Perceptual-motor skill learning. *Categories of Human Learning*, *47,* 381-391.

Fitts P.M., Posner M.I. (1967). *Learning and Skilled Performance in Human Performance*. Brock-Cole, Belmont, CA.

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica, 71,* 89-146.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27,* 591-635.

Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to training of complex skills. *Acta Psychologica, 71*, 147–178.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (2013). An integrated theory for improved skill acquisition and retention in the three phases of learning. *Theoretical Issues in Ergonomics Science, 14*, 22-37.

Lebiere, C., Jentsch, F., & Ososky, S. (2013). Cognitive models of decision making processes for human-robot interaction. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 285-294). Springer, Berlin, Heidelberg.

Rosenbaum, D. A., Carlson, R. A., & Gilmore, R. O. (2001). Acquisition of intellectual and perceptual-motor skills. *Annual Review of Psychology, 52*, 453-470.

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human Factors, 50*, 540-547.

Salas, E., & Fiore, S. M. (Eds.). (2004). *Team cognition: Understanding the factors that drive process and performance.* Washington, DC: APA.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review, 120*, 439-471.

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? A model of learning the past tense without feedback. *Cognition, 86*, 123-155.

Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General, 137*, 548-565.

Tenison, C., Fincham, J. M., & Anderson, J. R. (2016). Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive Psychology, 87*, 1-28.

Predicting Performance in Cardiopulmonary Resuscitation

Kevin A. Gluck[1] (kevin.gluck@us.af.mil)
Michael G. Collins[2] (michael.collins.ctr@us.af.mil),
Michael A. Krusmark[3] (michael.krusmark.ctr@us.af.mil)
[1]Air Force Research Laboratory, [2]ORISE at AFRL, [3]L3 Technologies at AFRL
Wright-Patterson Air Force Base, Ohio, USA
Florian Sense (f.sense@rug.nl), Sarah Maaß (s.c.maass@rug.nl), Hedderik van Rijn (d.h.van.rijn@rug.nl)
Department of Experimental Psychology, University of Groningen
Groningen, The Netherlands

Abstract

Cardiopulmonary resuscitation (CPR) is a real-world basic lifesaving skill that requires a complex combination of declarative memory and psychomotor skill. It is also simple and brief enough to be practical for laboratory use. Here we describe a repeated measures study with increasing lags between sessions. At the time of the writing of this initial manuscript submission, the final session of CPR performance data has not been run. This paper documents our participant-level performance predictions for that final session, using the Predictive Performance Equation (PPE; Walsh, Gluck, Gunzelmann, Jastrzembski, & Krusmark, 2018). With the final lag period for that final experiment session at approximately one year for every participant, we will be able to assess predictive accuracy of PPE over an ecologically relevant timeframe.

Keywords: skill acquisition; retention; learning; memory; CPR; performance prediction; mathematical model

Introduction

Cardiopulmonary resuscitation (CPR) is an essential component of first aid training. The Basic Life Support CPR procedure, as laid out by the European Resuscitation Council (ERC) guidelines (Soar et al., 2015), includes an assessment of the so-called victim (check response, check breathing) and a series of steps (alert emergency services, hand positioning) before the actual chest compressions and rescue breaths are administered in a cycle of 30 compressions and two rescue breaths. In addition to its being a critical life-saving skill, CPR is a useful domain for studying human performance. It is a task that combines declarative knowledge and psychomotor skills, and clear performance standards are available. High fidelity training and assessment equipment, such as the Laerdal QCPR manikin used in this study, record and store detailed performance measures automatically.

Crucially, CPR certification entails periodic retraining to ensure performance remains above criterion. For medical professionals, retraining is typically completed every other year. This interval is often considered suboptimal as performance is likely to drop below the criterion during this period (Stross, 1983). Furthermore, the American Heart Association (AHA) recognizes that large individual differences in CPR performance exist, which complicates the prescription of ideal methodology and frequency of training (Nolan et al., 2015). Consequently, cognitive-psychological theories of learning and retention over realistic time-frames could provide a benefit to public health and safety by accurately predicting when someone should be provided with additional training to remain above performance criteria.

To validate learning and retention theories for this purpose, we initiated the collection of a CPR dataset (Sense, Maaß, Gluck, & van Rijn, 2019, https://osf.io/m8bxe/). A benefit of studying CPR performance is that there exist certain sub-populations who have been trained previously on this task before entering the lab. Specifically, part of the requirements to obtain a German driver's license is to demonstrate CPR performance above criterion. Therefore, German students with a driver's license are a suitable population to test long-term retention of procedural and declarative knowledge because they had CPR training in the past, typically had no retraining, and there will be natural variation in time since last presentation.

Mathematical models of learning and retention can help describe fluctuations in CPR performance over time based on individuals' prior performance. Sometimes the motivation in research and application of these models is to optimize repetition schedules *within* individual learning sessions (van Rijn, van Woudenberg, & van Maanen, 2009; Sense & van Rijn, submitted). Earlier research has shown that an ACT-R-based cognitive model can use response accuracy and latency on a trial-by-trial basis to predict when each studied item is likely to be forgotten and ensure rehearsal before that moment. This improves retention of the studied facts (Sense & van Rijn, submitted) and allows the estimation of a learner's rate of forgetting (Sense, Behrens, Meijer, & van Rijn, 2016).

Other times the motivation is to predict performance over longer time periods *between* sessions. This is our primary interest in the analyses reported here. A model that has shown some promise regarding its predictive validity over those longer between-session intervals, regardless of the relative mix of declarative or procedural knowledge involved, is the Predictive Performance Equation (PPE; Walsh, Gluck, Gunzelmann, Jastrzembski, & Krusmark, 2018). PPE is a set of equations capturing key human performance dynamics. First, activation increases with the number of learning events (N). This is implemented as a power law of learning, with the *learning rate* fixed at 0.1 based on prior empirical evidence and model fits (Equation 1). Because participants enter the study at different experience levels, we add to N a free parameter, a, to represent each individual's past CPR experience. Second, performance drops as a function of elapsed time among practice events (T; Equation 1). This is implemented as a power function of forgetting, with the decay rate determined by the function expressed in Equation 4, below. In PPE the effects of learning and retention on activation is multiplicative, such that:

$$activation = (N + a)^{learning\ rate} \cdot T^{-decay\ rate} \quad (1)$$

Third, PPE captures the spacing effect, such that retention is better and more stable when practice is distributed over time. This is implemented in the forgetting function through T and the *decay rate*. T is computed as the sum of the weighted age of each practice event,

$$T = \sum_{i=1}^{n-1} w_i \cdot t_i, \quad (2)$$

where the weight, w_i, is an exponential decay function of time,

$$w_i = t_i^{-x} \sum_{j=1}^{n-1} \frac{1}{t_j^{-x}}. \quad (3)$$

Thus, T weighs practice repetitions so that more recent events carry more weight, and the variable x, which is fixed at 0.6, controls the degree of the weighting.

The *decay rate* is computed as a function of the complete history of lags between successive practice opportunities (lag_j):

$$decay\ rate = decay\ intercept + decay\ slope \cdot average\left(\frac{1}{log(lag_i)}\right) \quad (4)$$

Finally, in PPE performance is computed as a logistic function of activation:

$$Performance = \frac{1}{1 + exp\left(\frac{threshold - activation}{scalar}\right)} \quad (5)$$

PPE parameters are estimated separately for individuals based on their performance histories. For each individual, we compute the optimal values of the *decay intercept*, *decay slope*, *threshold*, and *experience (a)* parameters that maximizes the likelihood of individual performance trajectories. These parameter values are then used to generate out-of-sample predictions of performance at future points in time.

An increasingly common modeling practice in environments with sparse and noisy data is to seed a model's parameter values with priors. This avoids over fitting and improves out-of-sample prediction (Yarkoni & Westfall, 2017). There has been some previous exploration of this approach in the context of PPE (Collins, Gluck, Walsh, & Krusmark, 2017; Collins & Gluck, 2018). Here we use priors generated from an independently completed CPR study (Jastrzembski et al, 2017). In that research, CPR compression data from four sessions separated by either one day, one week, one month, or three months were used to calibrate the model and generate model parameters for temporal predictions at either three or six months in the future. These model parameters were used in the current study to inform parameters generated during the model fitting process. The generalization of priors allows PPE to use available prior information about human performance on CPR.

The current study was devised to assess the accuracy of personalized performance predictions. The ERC's guidelines (Soar et al., 2015) state that "*The intervals for retraining will differ according to the characteristics of the participants*". The availability of predictive models that take an individual's performance profile over time into account permits such personalized predictions. Ideally, this would make retrainings more efficient and reduce the interval during which a medical professional might perform below criterion. The viability of personalized refresher schedules crucially depends on the accuracy of the predictions: Requiring people to retrain too early is a waste of resources but requiring them too late can cost lives.

Method

Participants

Fifty participants (age range = [18, 27]) were recruited for a first learning session, 40 took part in the second session, and 35 participated in the third session. All participants held a valid German driver's license.

Procedure and Stimuli

The full experiment protocol includes four sessions in which CPR performance data are collected. At the time these model predictions were run, participants had completed three experimental sessions, with the fourth session upcoming. In addition to assessing CPR performance in all sessions, a set of computerized laboratory tasks more typical of experimental cognitive psychology were also administered in some of the sessions. These are documented elsewhere in detail (Sense, Maaß, Gluck, & van Rijn, 2019, https://osf.io/m8bxe/) and are not a focus in this paper. A graphical summary of the CPR-specific experiment protocol is provided in Figure 1.

Session 1: Test 1.1. At the beginning of each session, participants signed the informed consent forms. In the first session participants then entered the experimentation room where a Laerdal Resusci Anne QCPR manikin was lying on the ground. Participants read the following instructions: "*You volunteered for community service to help elderly neighbors with chores in their homes. When you enter the house of Mr. Johnson, you find him on the living room floor. There are no signs of bleeding or open wounds and no one else is in the house. Based on your first aid training, take the steps* *necessary in this situation on the manikin to* ***assess*** *and* ***react upon*** *the situation*."

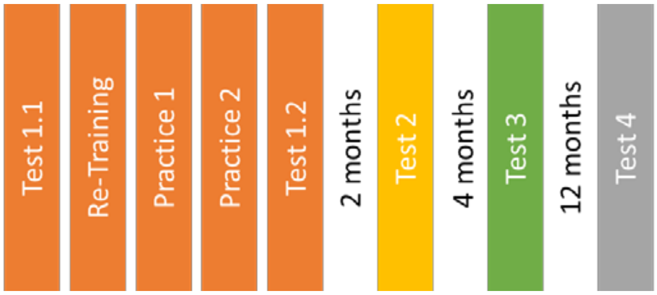


Figure 1. Overview of the experiment protocol.

This scenario was chosen to sketch a hypothetical scenario that required participants to perform CPR on the manikin. They were asked to perform the necessary actions required according to the ERC guidelines (Soar et al., 2015). This means participants were supposed to alternate between 30 compressions and two rescue breaths. Participants were stopped after administering four cycles of compressions and rescue breaths (i.e., 30-2, 30-2, etc.) to avoid fatigue. We refer to this procedure (i.e., initial steps followed by four rounds of 30-2) as one run-through of CPR.

Performance scores were based on Laerdal's proprietary scoring algorithm using the European guidelines (ranging from 0 to 100%; a score of 75% or higher is considered "proficient").

Retraining. After the initial assessment, participants were re-trained. First, participants watched a short instructional video (see https://osf.io/9er6g/) demonstrating the initial steps, as well as instructions on how to correctly apply chest compressions and rescue breaths. This video was specifically made for this research project.

Subsequently, participants had the opportunity to practice compressions on the manikin with its live feedback option enabled for one minute. That is, for each compression participants could track their depth and frequency and adjust if necessary. Then participants practiced giving rescue breaths until the live feedback indicated that two correct breaths had been given in a row. Following retraining, participants completed a basic lab task. As noted earlier, due to space limitations, details about the basic lab tasks will not be discussed.

Practice 1 and 2. Participants were instructed to "*Perform the complete procedure you saw in the video, with four rounds of compressions and rescue breaths*" twice while their performance was scored.

After the run-throughs of CPR, participants completed questionnaires to gather demographic information, the date their driver's license was issued, and the approximate number of months between obtaining their license and completing their CPR training. The time between the mandatory training and obtaining the driver's license ranged from 1 to 60 months (mean = 9.92 and SD = 12.71). Participants then completed two more basic lab tasks.

Test 1.2. Following the computerized tasks, participants were asked to complete another run-through of CPR. If the score of this test was below 75%, participants were re-trained until they reached criterion.

Session 2: Test 2. Participants completed a run-through of CPR. If performance was below 75%, they repeated the run-through. Participants also completed the full set of basic lab tasks.

Session 3: Test 3. Participants completed another run-through of CPR. If performance was below 75%, they repeated the run-through. Participants also did one minute of chest compressions without live feedback from the manikin, then rescue breaths until two consecutive ventilations were correctly performed.

Session 4: Test 4. Participants will complete another run-through of CPR. Then participants will watch the short instructional video again (as in Session 1) and complete another run-through of CPR.

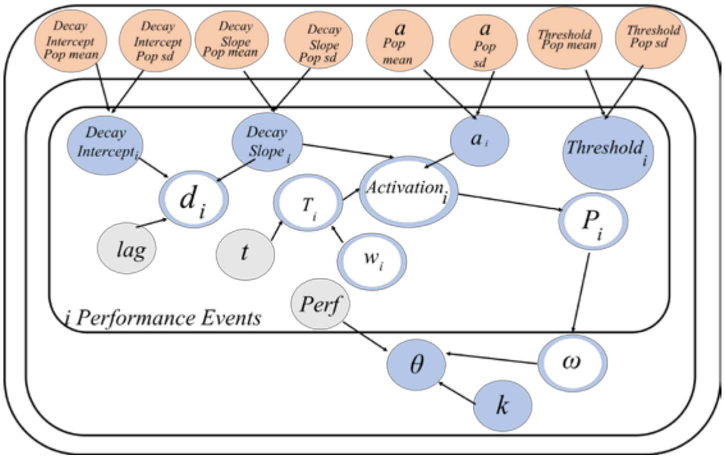
Predicting Future CPR Performance

As noted earlier, individual participant single performance event-level predictions present a small data challenge, especially earlier in the protocol. To manage and avoid overfitting to unexplained individual event-level variation we use Hierarchical Bayesian Modeling (Kruschke, 2014; Lee & Wagenmakers, 2013) to bias the PPE parameters with priors from a previous CPR study and generate posterior predictive distributions for each participant remaining in this study through Session 3.

In predicting CPR performance on Session 4, PPE's free parameter values were estimated using the model shown in Figure 2. For each participant, the model estimates probability distributions for PPE's free parameters (*decay intercept*, *decay slope*, *a*, *threshold*) that best characterize performance over the first three sessions. The estimates are based on a set of hyperparameters (decay intercept$_{pop}$ decay slope$_{pop}$, a$_{pop}$, threshold$_{pop}$). that were estimated from individual-level CPR performance data collected in a previous CPR study (Jastrzembski et al, 2017).

Unique parameter distributions are sampled for each individual participants from the hyperparameters to derive a distribution of values for each free parameter. The sampled set of parameters are then combined with the student's fixed time variables (t, N) and are transformed into performance predictions ($Perf_{i:n}$). The average of these performance predictions ($Perf_{i:n}$) is represented by variable omega (ω) and is then combined with free parameter k, to represent the model's prior beliefs of the student's performance ($\boldsymbol{\theta}$). This prior is then combined with the student's actual performance to generate a posterior estimate of performance.

Under this methodology, PPE's free parameter values are treated as a probability distribution, representing our degree of belief in a particular parameter value to generate a prediction. The final posterior probability distribution used to determine PPE's prediction is affected by two factors: (1) Prior, the beliefs about the most likely free parameter values before observing the performance of a participant (i.e., Prior



$$\text{Decay Intercept}_{\text{pop mean}} \sim \text{Normal}(.02\ .001) \qquad \text{Decay Intercept}_{\text{pop sd}} \sim \text{gamma}(854, 82536)$$

$$\text{Decay Slope}_{\text{pop mean}} \sim \text{Normal}(.003, .03\ ) \qquad \text{Decay Slope}_{\text{pop sd}} \sim \text{gamma}(127.97, 11652.7)$$

$$\text{a}_{\text{pop mean}} \sim \text{Normal}(.33\ .04) \qquad \text{a}_{\text{pop sd}} \sim \text{gamma}(3.80, 106.91)$$

$$\text{Threshold}_{\text{pop mean}} \sim \text{Normal}(.46\ .02) \qquad \text{Threshold}_{\text{pop sd}} \sim \text{gamma}(32.88, .195.96)$$

$$\textit{Decay intercept}_i \sim \text{Normal}(\text{Decay Intercept}_{\text{pop mean}}\ , \text{Decay Intercept}_{\text{pop sd}}\ )$$

$$a_i \sim \text{Normal}(\text{a}_{\text{pop mean}}\ , \text{a}_{\text{pop sd}}\ )$$

$$\textit{Threshold}_i \sim \text{Normal}(\text{Threshold}_{\text{pop mean}}\ , \text{Threshold}_{\text{pop sd}}\ )$$

$$\textit{Decay slope}_i \sim \text{Normal}(\text{Decay slope}_{\text{pop mean}}, \text{Decay slope}_{\text{pop sd}}\ )$$

$$d_i = decay\ intercept_i + decay\ slope_i * \left(\frac{1}{n-1} * \sum_{j=1}^{n-1} \frac{1}{\log(lag_j + e)}\right)$$

$$T_i = \sum_{l=1}^{i} w_l * t_l \qquad w_l = -t_l^{-.75} \sum_{m=1}^{i} \frac{1}{t_m^{-.75}}$$

$$Activation_i = (N + a_i)^c * T^{-d}$$

$$P_i = \frac{1}{1 + \exp\left(\frac{Threshold_i - Activation_i}{.1}\right)}$$

$$\kappa \sim \text{gamma}(1, 1)$$

$$\omega = \frac{\sum P_i}{\max(i)}$$

$$\theta \sim beta(\omega * (\kappa - 2), (1 - \omega) * (\kappa - 2)\ )$$

Figure 2. The hierarchical Bayesian model used to estimate free parameter values (*Decay Intercept$_i$*, *Decay slope$_i$*, *a$_i$*, *Threshold$_i$* - all shown in blue) for PPE prediction of performance in the 4th CPR session, given prior distributions (*Decay Intercept$_{pop\ mean}$* , *Decay Intercept $_{pop\ sd}$*, *Decay Slope$_{pop\ mean}$* , *Decay slope $_{pop\ sd}$*, *a$_{pop\ mean}$*, *a $_{pop\ sd}$*, *Threshold $_{pop\ mean}$*, *Threshold $_{pop\ sd}$* - all shown in salmon) based on a sample of CPR performance from a different study and the participant's prior performance across the first three sessions (*Perf*). Random variables are represented as circles, deterministic variables are represented as double circles, and observed variables (*lag*, *t*, *Perf*) are in grey.

CPR performance), and (2) the performance of a particular participant over the course of three sessions (i.e., likelihood). These two factors are combined together to generate a posterior distribution for each of PPE's free parameters. It is this posterior distribution that is used to make predictions for each participant's next performance. We do this iteratively through the experiment protocol for each of the 35 participants, culminating in a prediction for their upcoming performance in Session 4.

Results

Data collection for Session 4 is scheduled for May 2019, after the submission deadline of this manuscript. Our key interest at present is in documenting our predictions for each participant's CPR performance when they return for Session 4, approximately one year after they did Session 3.

In the process of generating Session 4 CPR predictions for each participant, we ran several simulations to assess how different assumptions about participants' past CPR experience would affect predictions. Recall that the participants in the current study were German college students with a valid driver's license, which required them to successfully complete CPR training prior to getting their license. Thus, from the issue dates on the licenses, we know the approximate date of each participant's initial CPR training. Although no performance measures are available, we assumed that all participants reached criterion level of performance (i.e., 75%) during this initial training. Based on this information, we combined the 75% performance score that we assumed at the time of licensing with the data from Sessions 1 and 2, and predicted performance on Session 3. We then compared the accuracy of these predictions to predictions from the model with only data from Sessions 1 and 2 predicting Session 3. Results of this comparison showed that predictions were more accurate when we ignored the licensing data. A possible explanation for this is that we were making assumptions about the level of performance participants reached when they received their license, but not that they started with no experience. Thus, we ran the model again assuming that performance was 0 prior to their initial training, and that it increased to 75 afterwards. But again, this did not improve predictive accuracy. Predictions were more accurate when we made no assumptions about CPR performance prior to the onset of the study.

Figure 3 plots data for the fit and prediction methodology described in the previous section for each of the 35 participants. A data file documenting the raw values used to generate the graph is available at (https://osf.io/5ma29/). Performance scores are exported from Laerdal's proprietary software, which combines the compression and rescue breath performance into a single score.

On the initial test at the beginning of the first session, only two participants demonstrate proficient performance (a score of 75% or above), while many score below 25%. The CPR Retraining administered between CPR Test 1 and CPR Practice 1 results in a marked increase in performance, making the majority of participants reach criterion. Testing for a difference between those two scores with a paired Bayesian t-test yields a decisive Bayes Factor of 2.4×10^{17} in favor of a difference. The second practice marks a further increase in overall performance and the vast majority of participants retain above-threshold performance until CPR Test 2 at the end of the first session. In the eight-week interval until the second session, and performance decreases (BF$_{H1}$ = 9.95) but many participants still exhibit near-ceiling performance.

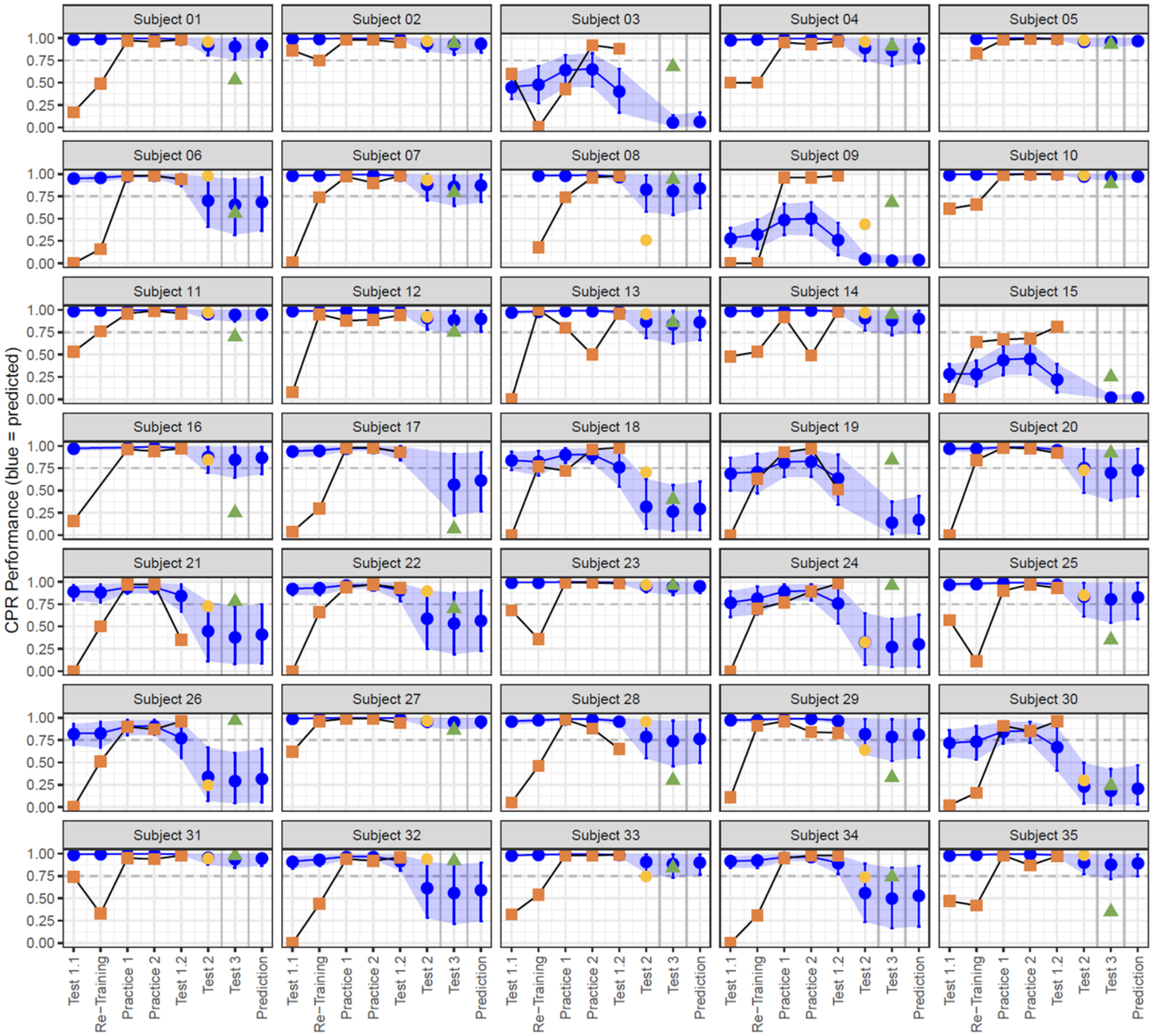


Figure 3. CPR performance for each of the 35 participants. Observations within a session are connected by lines; sessions are indicated by color and shape. Session 1 is orange squares. Session 2 is a yellow circle. Session 3 is a green triangle. The model's predicted performance is shown in blue, with mean predicted performance at each measurement indicated by the blue circle. The blue ribbons indicate the 95% highest density interval (HDI) of the posterior distribution at each instance. The predictions for the final session are the rightmost blue circles in each panel.

Another way to summarize the data and contrast the predictions with the recorded data is to compute a prediction error at each measurement event. In Figure 4, predicted performance has been subtracted from the actual performance to express the prediction error at each measurement event. The color-coding indicates the session and the root-mean-squared-error (RMSE) is listed for each event to summarize the prediction error. At the first event, most errors are negative, suggesting that predicted performance was consistently estimated to be higher than the recorded performance. This is also apparent in Figure 3, where we see human performance nearly always worse than the model's posterior predictions at Test 1.1. The RMSE decreases over the events in the first session and increasingly normally distributed around zero, suggesting that the model's posteriors become less biased: Performance is overestimated about as often as it is underestimated.

Discussion

The focus of the current work is the prediction of future CPR performance over ecologically valid periods. After completion of the first three sessions, individual predictions have been made for CPR performance in the 4th session.

As can be seen in Figure 3, the priors do a generally poor job of representing the actual performance of participants in the early trials. This is a risk in generalizing parameters from one study to another. They are different samples, with participants in the previous study starting at and maintaining higher levels of proficiency. Given that participants in the study reported here started at a lower proficiency, it is to be expected that the prior distributions based on better performers would not predict worse performer data very well. However, most participants in both studies achieved and maintained higher levels of proficiency after several trials, so this is bias-variance tradeoff we are willing to make in the interest of what we hope will be an improvement in predictive validity in Session 4. Additionally, the use of Bayesian Hierarchical Modeling as a method of parameter estimation provides posterior predictive distributions for each individual's learning profile. The use of prediction intervals allows for a quantification of certainty in our out-of-sample predictions.

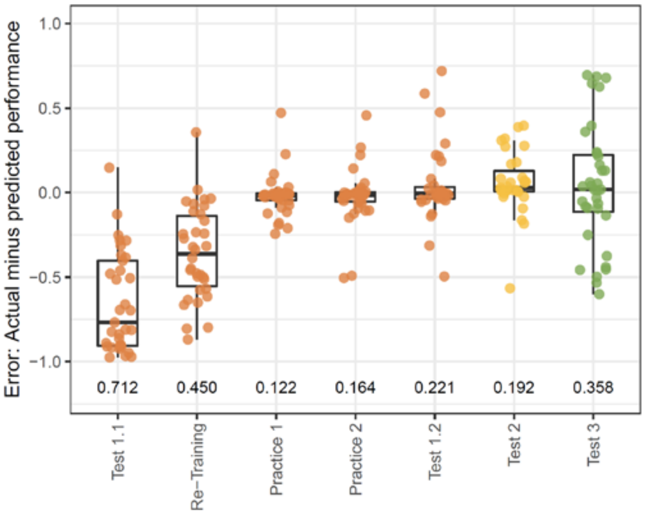


Figure 4. Prediction errors at each measurement event. Colors indicate the session; numbers at the bottom are the RMSE at each event.

The ERC's report (Soar et al., 2015) states that CPR performance is known to deteriorate within months of training and, therefore, even annual retraining might not be frequent enough for some people. Due to the fact that CPR training can be time consuming and optimal training intervals are currently unknown, they suggest that frequent "low dose" training using video instructions and hands-on practice can be as effective as instructor-led courses (Nolan et al., 2015). The work presented here confirms that there is a swift improvement in performance after such CPR retraining.

In summary, we report an experimental setup in which the learning and forgetting of CPR is assessed over ecologically relevant timeframes. We test a mathematical model's ability to predict future CPR performance using very sparse data. A first wave of predictions is presented here and an evaluation of the accuracy of those predictions will be presented at the conference.

Acknowledgments

This work was supported by EOARD grant #11926121 and by the Air Force Research Laboratory's 711 Human Performance Wing, Cognitive Models and Agents Branch.

References

Collins, M. G., Gluck, K. A., Walsh, M. M., & Krusmark, M. A. (2017). Using prior data to inform initial performance predictions on individual students, In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*.

Jastrzembski, T., Walsh, M. M., Krusmark, M., Kardong-Edgren, S., Oermann, M., Dufour, K., . . . , & Stefanidis, D. (2017). Personalizing training to acquire and sustain competence through use of a cognitive model. In D. D. Schmorrow & C. M. Fidopiastis, *Augmented cognition. Enhancing cognition and behavior in complex environments* (pp. 148-161). Switzerland: Springer International Publishing AG.

Kruschke, J. K. (2014). *Doing Bayesian data analysis: A tutorial with R and BUGS*. New York, NY: Academic Press

Lee, M.D., & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. Cambridge University Press.

Nolan, J. P., Hazinski, M. F., Aickin, R., Bhanji, F., Billi, J. E., Callaway, C. W., ... & Gent, L. M. (2015). Part 1: Executive Summary: 2015 International Consensus on Cardiopulmonary Resuscitation and Emergency Cardiovascular Care Science with Treatment Recommendations. *Resuscitation*, 95, e1-e31.

Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials. *Topics in Cognitive Science*, 8: 305–321.

Sense, F., Maaß, S., Gluck, K. A., & Van Rijn, H. (2019). Within-subject performance on a real-life, complex task and traditional lab experiments: Measures of word learning, Raven matrices, tapping, and CPR. *Journal of Cognition*, 2(1), 12. DOI: http://doi.org/10.5334/joc.65

Sense, F., & Van Rijn, H. (submitted). *Optimizing Within-Session Repetition Schedules with a Response-Latency-Based Cognitive Model Improves Retention*.

Soar, J., Nolan, J. P., Böttiger, B. W., Perkins, G. D., Lott, C., Carli, P., ... & Sunde, K. (2015). European Resuscitation Council guidelines for resuscitation 2015. *Resuscitation*, 95, 100-147.

Stross, J.K. (1983). Maintaining competency in advanced cardiac life support skills. *Journal of the American Medical Association*, 249(24), 3339–3341.

van Rijn, H., van Maanen, L., & van Woudenberg, M. (2009). Passing the Test: Improving Learning Gains by Balancing Spacing and Testing Effects. In A. Howes, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 9th International Conference on Cognitive Modeling* (pp. 110–115). Manchester, UK.

Walsh, M. M., Gluck, K. A., Gunzelmann, G., Jastrzembski, T., & Krusmark, M. (2018). Evaluating the theoretical adequacy and applied potential of computational models of the spacing effect. *Cognitive Science*, *42*(S3), 644-691. DOI: https://doi.org/10.1111/cogs.12602.

Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100-1122.

Evolutionary Optimization of Neural-Network Models of Human Behavior

Anonymous Submission

Abstract

Neural network models are essential tools in understanding how behavior arises from information processing in the brain. Recent advances in computing power and neural network algorithms have made more complex models possible, increasing their explanatory power. However, it is difficult to make such models work: they have many configuration parameters that have to be set right for the model to work properly. Consequently, automated methods are needed to optimize them. This paper proposed an evolutionary approach to this problem. An Age-Layered Evolutionary Algorithm is introduced and evaluated by fitting training parameters for BiLex, a self-organizing map model of lexical access in bilinguals. The resulting configurations are highly optimized and able to generalize to previously unseen human data, showing that evolutionary optimization of complex models has the potential to play an integral role in cognitive modeling in the future.

Keywords: Neural Networks; Cognitive Modeling; Evolutionary Algorithms

Introduction

Over the last few decades, connectionist neural networks have become an essential tool to characterize and investigate human cognition. Models based on such networks are usually not intended as physiologically accurate simulations of biological neurons and their interactions; nevertheless, they exhibit many characteristics of information processing in biological systems, including robustness to damage and input errors, and the ability to learn and generalize. This property of brain-like information processing on an abstract level is the main advantage of neural network-based models, enabling them to capture many aspects of high-level cognition while relying on mechanisms that are plausible analogs of the underlying neural substrate.

Recent progress in computing technology, such as GPU computing and software frameworks that rely on it, like Theano and TensorFlow (Abadi et al., 2015; Theano Development Team, 2016), have dramatically increased the performance and complexity of achievable models. At the same time, advances in neural network algorithms and architecture like deep learning and reservoir methods (Schmidhuber, 2014; Maass, Natschlager, & Markram, 2002) have made use of these capabilities, and thus the scale and performance of neural network applications have increased in equal measure.

Together these advances can significantly improve cognitive neural network models. Most importantly, rather than simulating behavior on an abstract and qualitative level, sufficiently large and complex networks can now be built so that clinical and psychometric tests can be modeled directly and quantitatively. Furthermore, rather than demonstrating that a certain kind of function of behavior can plausibly occur in a model, modern architectures can be used to investigate the link between environmental factors on the one hand, and the resulting individual differences on the other.

Building this new brand of models presents new and unique challenges. Most importantly, their ability to capture individual differences in behavioral data makes them sensitive to a large set of interdependent parameters governing e.g. module sizes, extent and intensity of training and pre-training, and input/output behavior of different classes of artificial neurons. In contrast to typical models in the past, fitting a model's many parameters manually in order to account for behavioral data is no longer feasible.

Another significant challenge is that the amount of individual human data available is often limited. Since the required amount increases with larger parameter spaces, and since quantitative measures need to be elicited for both target behavior and any individual differences of interest, acquiring the data necessary for accurate parameter fitting becomes prohibitively difficult.

Third, for an interdependent set of parameters that influence the behavior of the model in a non-linear way, fully evaluating a given set of model parameters involves training a complete model for each human participant. The resulting goodness-of-fit measure provides no gradient w.r.t. the parameter set. Therefore, the standard gradient based methods of metalearning cannot be used to optimize these models.

This paper proposes an evolutionary approach to these issues. The goal is to make parameter fitting of complex neural network models to limited human data workable in practice. In order to limit the cost of evaluation, the proposed EA uses a variant of the previously introduced Age-Layering technique (Shahrzad, Hodjat, & Miikkulainen, 2016), which aims to focus detailed evaluations on the most promising candidates.

The approach is evaluated in optimizing parameters for BiLex, a neural network model of the bilingual lexicon (Anon et al., 2016). BiLex simulates tests used in clinical practice, and captures the complex interactions between exposure to different languages and the resulting individual differences in bilingual lexical access. It is a complex model of individual subjects, for which little training data is available. It is therefore an appropriate test case for the proposed approach.

The next section gives an overview of bilingualism and the BiLex model. Using BiLex as a working example, the following sections then introduce and evaluate the proposed model fitting method, and discuss the results.

Bilingualism and the BiLex Model

The mental lexicon, i.e. the storage of word forms and their associated meanings, is a central component of language processing. The lexicon of bilinguals is considerably more complex than that of monolinguals, and the ways in which multiple language representations can develop, coexist, and interact are incompletely understood.

Given that the majority of the world's population is bilingual or multilingual (Bhatia & Ritchie, 2005), extending existing modeling approaches to improve our understanding of these additional complexities is of considerable practicle importance, and computational models of the bilingual lexicon could contribute to novel approaches in bilingual research, education, and clinical practice.

Current theoretical models of the bilingual lexicon generally agree that bilingual individuals have a shared semantic (or conceptual) system, and that there are separate lexical representations of the two languages (L1 and L2). However, the models differ on how L1 and L2 interact with the semantic system and with each other. The most recent model is the Revised Hiearchical Model, proposed by Kroll & Stewart (Kroll & Stewart, 1994). It assumes connections of varying strength between all three components, depending on relative language dominance.

The physiological structure and location of the lexicon in the brain are still open to some debate, but converging evidence from imaging, psycholinguistic, computational, and lesion studies suggests that the lexicon is laid out as one or several topographic maps, where concepts are organized according to some measure of similarity (Caramazza, Hillis, Leek, & Miozzo, 1994; Spitzer et al., 1998).

Self-organizing maps (SOMs; Kohonen, 2001) are neural networks that model such topographical structures, and are therefore a natural tool to build simulations of the lexicon. SOM models have been developed to understand e.g. how ambiguity is processed by the lexicon (Miikkulainen, 1993), and how the lexicon is acquired during development (Li, Zhao, & MacWhinney, 2007).

Following the Kroll & Stewart model, and using SOMs as its building blocks, the BiLex model consists of three separate maps: one for word meanings, and one each for phonetic symbols in L1 and L2, as illustrated in figure 1. All maps are linked by associative connections of varying strength, which allow network activation to flow between them.

Training Corpus During model training, the semantic and phonetic maps need to organize according to similarity, i.e. on the semantic map, words with similar meaning will tend to be close, while on phonetic maps, words that sound similar will tend to be close. For this organization to occur, semantic and phonetic symbols need to be encoded as vectors that reflect this similarity.

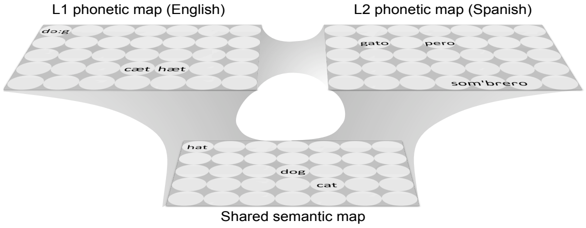


Figure 1: The BiLex model consists of three SOMs, one each for semantics, L1, and L2, that are linked by associations that enable the model to translate between semantic and phonetic symbols, simulating lexical access in bilingual humans.

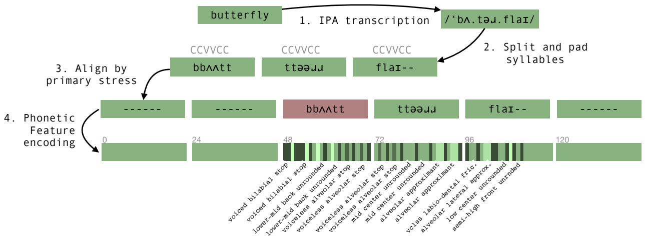


Figure 2: A word is encoded as a phonetic vector representation, creating the basis for phonetic map organization in the BiLex model.

Feature-based semantic and phonetic vector representations were developed for a training corpus of 638 concrete nouns in English and their direct translations to Spanish. Semantic representations were derived from feature data developed by Sandberg, Gray, and Kiran (2018). For each word, 10-20 relevant attributes (e.g. "can fly") were used that were normed on healthy adults using Amazon MTurk. Overall, data from more than 320,000 interactions of the type "does word X have feature Y?" were used to produce semantic vectors of 400 features.

Phonetic representations were based on phonetic transcriptions of English and Spanish words, which were split into spoken syllables, and padded such that the primary stress lined up for all words. The individual phonemes comprising each syllable were represented numerically as a set of phonetic features like height and front-ness for vowels, and place, manner, etc. for consonants (Ladefoged, 2001). Figure 2 illustrates the encoding process. The final phonetic representations consisted of 144 real-valued features for English, and 192 for Spanish.

Model Training

Using the semantic and phonetic symbols as input data, the organization of the three maps and the associations between them are learned simultaneously. Symbols are presented to two of the maps at the same time; the two exposed maps adapt, and over time become more likely to represent each symbol in the corpus accurately. At the same time, associative connections between corresponding semantic and phonetic symbols grow stronger.

Varying relative exposure to English and Spanish can be

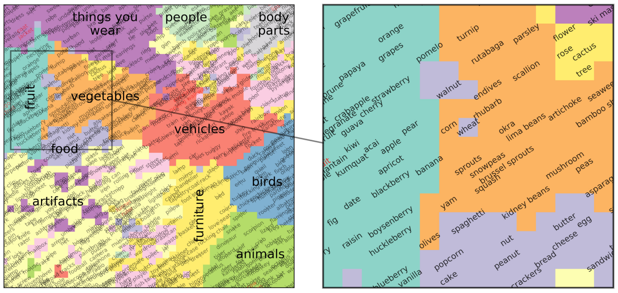


Figure 3: A well-trained semantic map, with winner units for each word labeled (LEFT). Map units are colored according to semantic categories, showing good global organization. RIGHT: Detail demonstrating that semantic similarity is reflected locally as well (e.g. walnut, nut, peanut are neighbors). Carefully designed training parameters are essential in creating this kind of highly organized map.

simulated by presenting English and Spanish phonetic symbols at proportional frequencies during model training, enabling the model to capture the effects of an individual's language learning history.

SOM Training Each SOM consists of a two-dimensional grid of neurons; each neuron is associated with a weight vector that encodes a semantic or phonetic symbol. The maps are trained using the standard SOM training algorithm (see e.g. Kohonon, 2001), which causes the weight vectors to become representations of the symbol vectors. At the same time, neighboring weight vectors become similar, and the map learns to represent the space of symbols as a two-dimensional layout where units that are close to each other on the map are similar either semantically (in the semantic map) or phonetically (in the phonetic maps).

SOM training is mainly governed by two parameters: the learning rate α determines the intensity of training, and the *neighborhood size* σ determines whether a larger or smaller part of the map changes in response to a training input.

The effectiveness of SOM training depends critically on how σ and α change over time. To develop the map's global structure first, the size of the neighborhood usually starts relatively large (on the order of the size of the map), and is gradually reduced, which causes the map to learn the similarity relations between input patterns at a more and more fine-grained level. Similarly, the learning rate is usually reduced over time, which allows the map to fine-tune its weight vectors in later stages of training.

Figure 3 shows an example of a well-trained semantic map, with colors encoding rough semantic categories. The categories tend form contiguous areas that border on similar categories. The detail on the right illustrates that locally, concepts tend to be arranged according to semantic similarity as well, e.g. ”walnut”, ”nut”, and ”peanut” form a tight cluster.

A central working assumption underlying the BiLex model is that, similar to the training schedules necessary to achieve well-organized SOMs, language acquisition during human development requires an equivalent progression of factors governing learning. In other words, the cortical structures that underlie the human lexicon start out highly flexible and adaptive, but later in life adapt only to a smaller degree, both in terms of learning intensity and overall flexibility. In this way, SOM-based models can provide a mechanistic explanation for the age-related limitations on second language learning that occur in humans.

Training Associative Connections In addition to map training, associative connections between the maps are adapted simultaneously based on Hebbian learning, i.e. by strengthening connections that link active neurons:

$$a'_{ij} = a_{ij} + \alpha\theta_i\eta_i\theta_j\eta_j,$$

where a_{ij} is the weight of the associative connection from unit i in one map to unit j in the other map, η_i is the activation of unit i, and θ_i is a function defining the current map neighborhood.

In order to prevent the associative strengths from increasing indefinitely, the the overall sum of outgoing associative connections is normalized such that for each neuron, the L2 norm of outgoing connections to each target map is 1.

Additionally, since lexical access can decline in humans with age or lack of exposure to a language (Kavé, Knafo, & Gilboa, 2010), small amounts of random noise (with a given variance γ) are added to the associative connections during training.

Simulating Naming Tests Once a BiLex model is trained, the task of naming an individual word in either language can be simulated by first presenting its semantic representation to the semantic map. The resulting map activation is then propagated to the phonetic map via the associative connections; the weight vector of the most highly activated phonetic unit is then compared to all phonetic representations in the corpus, and the word with the minimal distance is produced as output. If the output word matches the original input, the word is counted as correctly named. The simulated naming performance for a set of words is the percentage of words that are correctly named in this way.

Evolutionary Parameter Fitting

In BiLex, age and relative language exposure over time are based on individual human data: the age of an individual determines the number of epochs used for model training, one training epoch per simulated year. The relative exposure to each language determines the proportion of English vs. Spanish words randomly selected for training during each epoch. However, appropriate settings for all remaining parameters governing the training process are initially unknown, including how learning rates and neighborhood sizes for the SOMs change over time. Finding parameter settings that enable BiLex to match an individual's naming performance given past language exposure is a complex problem, involving precise tuning of a large set of interdependent parameters. The remainder of this section describes an Evolutionary Algorithm (EA; e.g. Bäck et al., 1997) designed to solve this problem.

EAs are a class of population-based optimization algorithms that use mechanisms inspired by biological evolution, like reproduction and mutation, to solve optimization problems. EAs maintain a population of candidate solutions, using a *fitness function* to determine the quality of each candidate. Highly fit candidates are more likely to be selected to reproduce, and recombine with other highly fit candidates. In this way, evolution tends to produce better candidate solutions over time.

Representation of Candidate Parameter Sets For the present problem of finding the best possible parameter settings for BiLex, each candidate solution was a set of BiLex training parameters, excluding age and relative language exposure, but including α and σ at different simulated ages, the scale γ of the random noise added to simulate aging and attrition effects, and the size N for each SOM.

To avoid overfitting, both α and σ were assumed to be the same for all three maps. Specific values for α and σ were evolved at a number of simulated ages (1,4,7,10,13,19,25, and 50 years), and interpolated linearly for intermediate values. Additionally, both α and σ were constrained to non-increasing values, i.e. at each time, the minimum of all values so far was used for training.

Training the associative connections also requires a learning rate α' at each time during training. To limit the number of parameters, a single factor k was added, such that at each time, $\alpha' = k \times \alpha$. In this way, the scale of α' was independent of that for SOMs, but changed in the same way over time.

To account for the fact that monolinguals tend to score above zero on naming tests in the other language, a minimum exposure parameter ε was added such that exposure for each language was clipped to values between ε and $1 - \varepsilon$.

Initially, the number of words trained per simulated year were also included in the set of evolved parameters, which turned out to be unnecessary. In the reported experiments, the number of trained words per simulated year was set to a fixed value of 1.5 x the size of the training corpus.

Overall, each candidate parameter set was encoded using 20 numeric values; the initial population of 100 candidates was generated using random values within reasonable intervals, which were chosen empirically for each parameter. E.g. neighborhood sizes were constrained to an interval between 0 and 10, and initial learning rates ranged from 0 to 0.4.

Evaluation and Age-Layering In order to evaluate how well a particular candidate was able to match the naming performance of a given human participant, a BiLex model was trained, and the naming tests administered to the human participant were simulated using the trained model. The goodness-of-fit for a given candidate on a human individual i (GOF_i) was then calculated as the sum of squared residuals for the naming scores in both languages.

Based on this GOF measure, the straightforward way of fully evaluating the fitness of a candidate would be to evaluate it on all training samples, and compute the fitness as the mean GOF measure, requiring training and evaluating a complete model for each i, and making the evaluation function extremely expensive.

As a possible solution, age-layered EAs (Shahrzad et al., 2016) attempt to limit complete evaluations to only the most promising candidates. Candidates that score highly on an initial limited evaluation are further evaluated, while weak candidates are eliminated, saving computing resources.

Age Layering is particularly useful for noisy and expensive evaluations, and has been shown to speed up evolution significantly. To optimize BiLex parameters, a slight variation was used that accounts for the small, fixed set of human individuals on which each candidate can be tested: rather than ranking candidates by their overall fitness, a separate ranking for each human data set i was computed, and candidates were then discarded if their average ranking was below the 50th percentile within their age layer.

EA Design The remaining components of the Evolutionary Algorithm were fairly standard (see e.g. Bäck, 1997); Parents were chosen by standard roulette-wheel selection; offspring was created using uniform crossover, and mutated by adding normally distributed noise with uniform probablility (p=0.05) and standard deviation 0.025, scaled by the size of the initialization interval for each parameter.

In order to simplify distributed evaluations across remote compute nodes, and because age-layering makes the time required for evaluations unpredictable, a steady-state EA was used, i.e. rather that proceeding in distinct generations, population size was maintained between 50 and 70 candidates by adding new candidates continually as needed.

Finally, if none of the most recent 500 candidates was able to improve on the previous best solution, a mutation burst was performed, i.e. to maintain diversity, new candidates were added without recombination, but using a high mutation rate of 0.5. If no improvement was observed in the 1000 most recent candidates, the EA terminated, and the current best solutions were used as the final result. All parameters governing the EA were set empirically.

Experiments

Human Data The human data used to evaluate the parameter tuning methodology were collected from 33 healthy adult individuals, including 28 Spanish-English bilinguals and 5 monolinguals (2 Spanish, 3 English), who were included in order to provide the EA with appropriate edge cases w.r.t. language exposure and naming performance.

Relative exposure to English vs. Spanish over each individual's lifetime was estimated using a standard Language Use Questionnaire (LUQ19; Kastenbaum, 2018), which included questions about age, native and second languages, as well as a detailed self-reported linguistic profile that included relative exposure to both languages.

In order to measure lexical access (i.e. naming performance) in English and Spanish, all participants completed the Boston Naming Test (BNT; Kastenbaum, 2018), as well as another 60-item picture naming screener test used in clinical practice. To reduce the noise inherent in such tests, both tests were averaged to obtain one composite naming score for each language.

The provided data on language exposure and age made it possible to modulate relatitve English vs. Spanish exposure over the course of the simulated lifetime for each individual human, creating an individual BiLex model whose naming performance could be measured and compared to the actual test scores.

Validating the Evolutionary Parameter-Fitting Method In order to evaluate the generalization performance of the proposed evolutionary method, a five-fold cross-validation run was conducted, using the human data described above as either training or test data. The initial set of 33 participants was divided randomly into five test sets, with each test set containing one monolingual. For each test set, the EA parameter optimization was performed using the remaining 26 or 27 healthy controls as training data. Generalization performance was measured as the goodness-of-fit on the respective test sets: For each individual in a test set, a model was trained using parameters that were evolved to fit the naming performance of the respective training set. Since each control subject was part of one test set, this was possible for all 33 controls.

Results

All five cross-validation runs produced highly fit candidate solutions; final best-fit parameter sets were found after evaluating 2749 (SD=1023) candidates on average, training and evaluating an average of 13549 individual BiLex models. Complete evaluation of all candidate parameter sets would have required over 7x as many trained models, suggesting that the age-layering approach was highly effective in reducing the number of required evaluations.

Most parameters in the best-fit candidate parameter sets tended to be similar, e.g. low but finite minimum exposure ε (0.04, SD=0.0137), and large initial neighborhood size (08.06, SD=1.17) that decreased dramatically (0.59, SD=0.049) by age 25.

Simulated composite naming scores were highly predictive of human data for both English ($R^2 = 0.77$, $p << 0.0001$) and Spanish ($R^2 = 0.63$, $p << 0.0001$). Figure 4 shows predicted vs. actual composite naming scores for both languages, using predicted naming scores from the top five parameter sets found by each of the five EA runs.

Figure 5 illustrates the way in which L2 age of acquisition (AoA) and exposure influence the structure of BiLex maps using concrete phonetic maps from four individual BiLex models; each map was trained using evolved training parameters and the language history of one of the bilingual study participants. The individual maps were chosen to represent extreme AoA/exposure combinations: Panel A shows early AoA and high exposure, from a model with high L2 naming score (> 90%). Panel B demostrates that as long as the AoA is early, the L2 map organizes and performs well even for low exposure. Panel C shows a late-AoA/high exposure map; the global organization deteriorates to some degree, but performance is still acceptable (70%). Finally, panel D shows how the combination of late AoA and low exposure leads to a badly organized map that accounts for low performance (<40%).

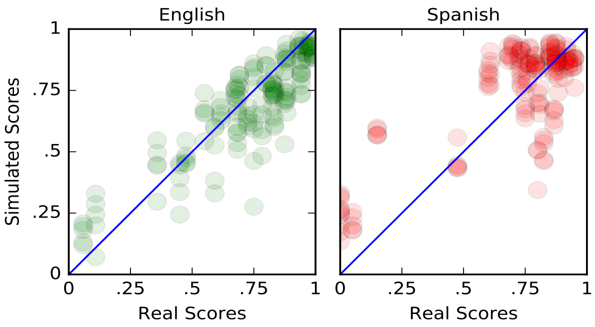


Figure 4: Simulated naming scores on test sets are highly predictive of human data for both English (left, $R^2 = 0.77$) and Spanish (right, $R^2 = 0.63$), indicating that models trained with evolved parameters are able to generalize to simulate bilingual access of previously unknown individuals.

Discussion

The complexity of the BiLex model, the infinite possible combinations of individual language history, and the comparatively small amount of human data available in this case make BiLex an appropriate test case for the evolutionary parameter fitting method proposed in this paper. The reported results demonstrate clearly that using evolution, a complex model like BiLex can be configured to capture complex interactions between environment and behavior, in a model that itself plausibly models neural information processing.

In addition to capturing the link between language exposure and naming ability quantitatively, the same link was visible in the organization of phonetic L2 maps in the optimized model: either early L2 acquisition or high exposure lead to well-organized L2 phonetic maps and high naming performance, while low exposure and late acquisition led to deficient map organization and naming ability.

In this way, models based on known theories, and designed to account for quantitative data on a more abstract level, can still provide additional insight and generate unexpected explanations for mechanisms underlying a given phenomenon – in this case, about the way in which AoA and exposure modulate lexical acces through phonetic map organization.

Note that, while BiLex was used as a concrete example throughout, the method extends to similar models, and aims to make parameter fitting of complex neural network models to limited human data workable in general.

Finally, while evolution can help models such as BiLex explain normal human cognition and capture the ways in which

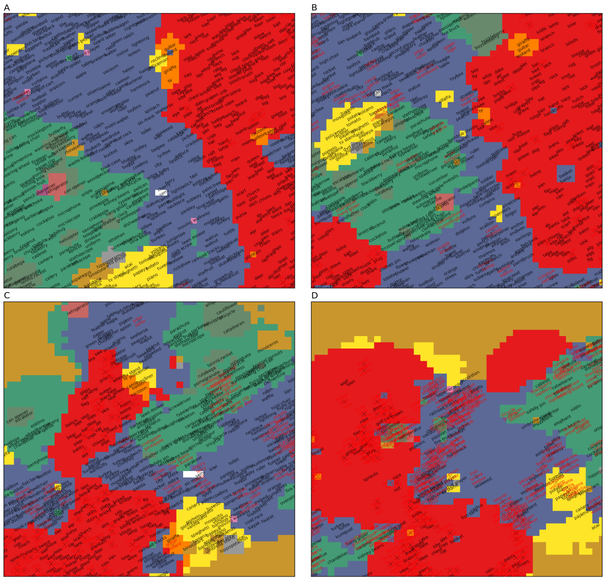


Figure 5: L2 phonetic maps of individual EA-optimized BiLex models. (A) Early age of acquisition (AoA) and high exposure leads to well-organized L2 phonetic maps. (B) Early AoA leads to well-organized maps despite low exposure. (C) Late L2 AoA impacts both global organization of the phonetic map even at high exposure. (D) Late AoA and low exposure lead to deficient global and local map organization. Taken together, the maps offer a mechanistic explanation for AoA/exposure effects seen in humans.

underlying brain mechanisms, environment, and cognitive function interact, the resulting models of normal cognition can also serve as a basis to investigate how these functions break down, and potentially inform the development of improved diagnostic methods and clinical interventions.

In current research, EA-optimized BiLex models are used in this way to create individual models of bilingual patients suffering from Aphasia; the resulting pre-morbid patient models are then used to simulate the onset of Aphasia, and to predict outcomes of alternative interventions. The approach is currently evaluated in an ongoing clinical trial, making it (to our knowledge) th first time a neural network model has been used in this way – the systematic, mechanical way of optimizing the model that was introduced in this paper makes novel modeling application such as these possible.

Conclusions

This paper proposed an evolutionary approach designed to make fitting complex NN-based models of higher cognition to limited data workable in practice. An Evolutionary Algorithm was introduced and evaluated by optimizing training parameters for BiLex, a connectionist model of the bilingual lexicon. Using EA-optimized parameters, BiLex was able to capture the complex interactions between exposure to different languages and the resulting individual differences in bilingual lexical access, demonstrating how evolution can help build the next generation of computational models of cognition.

References

Abadi et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* (Software available from tensorflow.org)

Bäck, T., Fogel, D., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. Oxford Univ. Press.

Bhatia, T. K., & Ritchie, W. C. (Eds.). (2005). *The handbook of bilingualism*. Blackwell Publishing.

Caramazza, A., Hillis, A., Leek, E., & Miozzo, M. (1994). The organization of lexical knowledge in the brain: Evidence from category- and modality-specific deficits. In L. Hirschfeld & S. Gelman (Eds.), *Mapping the mind.* Cambridge University Press.

Kastenbaum, J. G. e. a. (2018). The influence of proficiency and language combination on bilingual lexical access. *Bilingualism: Language and Cognition*, 1–31.

Kavé, G., Knafo, A., & Gilboa, A. (2010). The rise and fall of word retrieval across the lifespan. *Psychology and Aging*, *25*(3).

Kohonen, T. (2001). *Self-organizing maps* (3rd, extended ed.). Berlin: Springer.

Kroll, J. F., & Stewart, E. (1994). Category interference in translation and picture naming: Evidence for asymmetric connections between bilingual memory representations. *Journal of Memory and Language*, *33*.

Ladefoged, P. (2001). *Vowels and consonants: An introduction to the sounds of languages*. Oxford: Blackwells.

Li, P., Zhao, X., & MacWhinney, B. (2007). Dynamic self-organization and early lexical development in children. *Cognitive Science*, *31*.

Maass, W., Natschlager, T., & Markram, H. (2002). Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, *14*(11).

Miikkulainen, R. (1993). *Subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory*. MIT Press.

Sandberg, C. W., Gray, T., & Kiran, S. (2018). Development of a free online interactive naming therapy for bilingual aphasia. In *American speech language hearing association convention.*

Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *CoRR*, *abs/1404.7828*.

Shahrzad, H., Hodjat, B., & Miikkulainen, R. (2016). Estimating the advantage of age-layering in evolutionary algorithms. In *Proceedings of the genetic and evolutionary computation conference (gecco-2016, denver, co).*

Spitzer, M., Kischka, U., Gückel, F., Bellemann, M. E., Kammer, T., Seyyedi, S., ... Brix, G. (1998). Functional magnetic resonance imaging of category-specific cortical activation: Evidence for semantic maps. *Cognitive Brain Research*, *6*.

Theano Development Team. (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, *abs/1605.02688*.

A Skill-based Approach to Modeling the Attentional Blink

Corné Hoekstra (c.hoekstra@rug.nl)
Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence
University of Groningen, Groningen, the Netherlands

Sander Martens (s.martens@umcg.nl)
Department of Biomedical Sciences of Cells and Systems
University Medical Center Groningen, University of Groningen, Groningen, the Netherlands

Niels A. Taatgen (n.a.taatgen@rug.nl)
Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence
University of Groningen, Groningen, the Netherlands

Abstract

People can often learn a new task very quickly. This suggests that people are able to use skills that they have learned from a previous task, and apply them in the context of a novel task. In this paper we used a modeling approach based on this idea. We created a model of the attentional blink (AB) out of the general skills needed to perform an AB-task. The general skills were acquired from creating separate models of other tasks, in which these same basic skills are used. Those models showed a good fit with reported data, indicating that the basic skills we created are valid. Subsequently, we created the AB model by tying together the basic skills taken from the basic models. The AB model generated the same basic AB effects as reported in the literature. The models produced by the skill-based approach suggest that this is a feasible modeling method, which could lead to more generalizable models. Furthermore, it shed new light on previously difficult to explain findings in the AB literature.

Keywords: Attentional Blink, PRIMs, ACT-R, Skill-based modeling, Cognitive model, Instruction learning

Introduction

Humans have the impressive ability to learn certain relatively simple tasks with minimal instruction and in a very short period of time. The experimental tasks used in (cognitive) psychology are particularly good examples of these types of tasks. Participants have often never encountered these tasks before, yet they are quickly able to work out what to do. This quick learning suggests that people reuse previously learned skills and apply them to new contexts (Salvucci, 2013; Taatgen, Huss, Dickison, & Anderson, 2008). For example, if a task requires a stimulus to be remembered for later recall, people do not have to work out how to remember the stimulus, but they can simply reuse the already learned ‘remembering-skill’. It would be unnecessary, in this case, to reinvent the wheel. Learning how to do a new task simply means selecting the appropriate skills, assuming all these skills have already been acquired.

Reusing skills speeds up learning, but it can also have negative side effects that lead to sub-optimal performance even though the cognitive system is, in principle, capable of optimal performance. That is, it is sub-optimal strategy that underlies the impaired performance, not a fundamental information processing limit (e.g., Taatgen, Juvina, Schipper, Borst, & Martens, 2009). One factor underlying the sub-optimal strategy-choice might be the selection of the wrong skills, either because the "right" skill is not available, or because the interpretation of the task cues the wrong skill. A well-known instance of this is the Stroop effect (Stroop, 1935). Because people are so used to reading words, this automatically triggered skill interferes with the task of naming the color of the word. In this case, selecting the ‘reading-skill’ leads to worse performance. Another instance where this can happen is the attentional blink (AB).

The AB is a well-studied phenomenon in cognitive psychology (Martens & Wyble, 2010). It refers to the finding that the second of two to-be reported targets in a stream of distractors presented at a rate of 100 ms per item is often missed when it is presented within an interval of 200-500 ms after the first target (T1) (Raymond, Shapiro, & Arnell, 1992). Interestingly, the second target (T2) is hardly ever missed if it is presented directly (i.e., within 100 ms) after the first target (lag-1 sparing). This suggests that the cognitive system does possess the processing capacity to identify both targets, but that the chosen strategy prevents the second target from being reported.

The crucial aspect of the strategy that most participants use can be the selection of a sub-optimal skill to consolidate the targets in memory. Many theories of the AB assume that consolidation of T1 into memory underlies the AB. Memory consolidation is thought to be a serial process, meaning that only one consolidation process can occur at a time and that the consolidation has to be completed before a new item can be consolidated. This means that T2 cannot always be consolidated straight away, but sometimes has to wait for the consolidation of T1 to be completed. This leads to the AB when consolidation of T1 has not yet been completed before T2 has disappeared from visual short-term memory. However, such theories all assume that targets are consolidated as separate memory items, whereas in other areas of memory research it is assumed that multiple items are consolidated in a single chunk.

The strongest indication that strategy underlies the AB phenomenon is an experiment by Ferlazzo and colleagues (Ferlazzo, Lucido, Di Nocera, Fagioli, & Sdoia, 2007). In their experiment, participants were instructed to report two target letters (which were always a vowel and a consonant) either separately or as a single syllable. In the latter condition participants did not exhibit an AB. A possible explanation is that the original instruction cues a strategy in which all targets are consolidated separately, while the syllable instruction encourages consolidation of both targets in a single chunk. We will explore this difference by creating two versions of an AB-model that only differ in their consolidation strategy.

To create the model, we have used a novel approach. Instead of creating the model specifically for the AB, we built a model from the general skills that we have constructed as parts of other models. In other words, the AB model only links together existing skills. We chose this approach because it mirrors how participants performing an AB-task work out what to do. They do not start from scratch, but they tie skills they already possess together in such a way that allows them to perform an AB-task.

We created this model in the cognitive architecture PRIMs (Taatgen, 2013, 2014). PRIMs is based on ACT-R (Anderson et al., 2004) and works in a highly comparable way. The architectures of both ACT-R and PRIMs consist of a ‘central workspace’ and a number of modules capable of performing specific cognitive functions. The modules can communicate (i.e., exchange the results of their cognitive operations) with each other through the central workspace, which is subdivided in buffers. This exchange of information between the modules in PRIMs is controlled in largely the same way as it is in ACT-R. In ACT-R this is done by productions, and in PRIMs it is done by operators, but they have similar functionalities. A crucial difference between ACT-R and PRIMs is that in PRIMs operators are further organized into skills. A skill is a collection of general operators capable of accomplishing a certain goal or processing step. The generalizability of skills makes it possible to use the same skills in models of different experimental tasks. The organization into skills thus allows us to employ a novel approach to constructing cognitive models, placing them in a context of related models, tasks, and skills.

Each skill has a number of variables that are instantiated when a skill is used in the context of a task. It is by this mechanism that we tie together tasks, but also fill in specific values.

We had two main goals in this project. Firstly, we wanted to investigate the feasibility of creating a cognitive model by tying together already existing skills. Secondly, we wanted to create a model of the AB which is capable of capturing most of the effects found in the AB-paradigm, including differences due to instruction.

Method

Instead of creating operators specifically for the attentional blink, we first considered which general skills are required to perform an AB-task and assembled the AB-model from these skills.

Based on previous work and other models of the attentional blink, we identified four basic skills (cognitive processing steps) which had to be performed by our model of the AB. We developed these four skills by first creating models of other tasks which share (some) of these same basic skills. This step was done to get a better idea of what these general skills should be capable of and to test the plausibility of these skills.

First, we will describe the three models that provided the building blocks for the AB-model. The three models are: (1) a visual search model, (2) a model of a simple working memory (SWM) task and (3) a model of a complex working memory (CWM) task. Not all parts of all three models will be used for the AB-model, but all three contain at least one of the four basic skills needed to perform an AB-task.

The first model, the visual search model, is very straightforward. The goal of this model is to find a vowel on a screen filled with other letters. It is composed of three skills. The main search skill processes the current visual item and determines its category through memory retrieval. If it does not match the target category (vowel in this case), it transfers control to another skill which focuses on the next search item. In visual search this is a shift of attention to another item. If it does match the target category, it transfers control to a third skill, in this case a skill that clicks on the target with the mouse. Finally, if it runs out of items to attend to, it transfers control to yet another skill, which is not instantiated in the visual search model. In the AB-model, we will reuse the search skill to find targets, but we will instantiate it differently.

To illustrate, here are the operators that make up the search skill, slightly abbreviated for clarity. In these operators `V`*x* refers to a slot in the visual buffer, `RT`*x* refers to a slot in the retrieval (declarative memory) buffer, and `G`*x* refers to a slot in the goal buffer.

```
operator look-for-target {
      V1 <> nil // if there is a visual input
==>
      *fact-type -> RT1 // build a
      V1 -> RT3 // retrieval request
      nil -> V1 // and clear the input
}

operator keep-looking {
      V1 = nil
      RT2 <> *target-type // if it is not a target
==>
```

```
      *next-stim -> G1 // change to the skill that
                       // selects the next stimulus
}

operator found-target {
      RT2 = *target-type // if it is a target
==>
      RT3 -> G8 // Store the target in the goal
      *after-found-target -> G1 // and
               // switch to the skill to handle a target
}
```

In these operators, values that are preceded by an asterisk are variables that need to be instantiated for a particular task. For visual search, we instantiate `*fact-type` with `vowel`, `*next-stim` with the `attend-next` skill, and `*after-found-target` with the `click-item` skill.

The second and third basic model are strongly related and provide the final basic skills. Both models deal with working memory tasks which require the participants to remember presented items and, after presentation of the items, recall which items have been seen. Although they both include a consolidation step, they accomplish this step with a different skill. Both build a chunk in working memory, however they differ in the moment of consolidation. The “consolidate-separate” skill, used in the SWM-model, starts consolidation immediately after an item is encountered. In contrast, the “consolidate-chunk” skill, used in the CWM-model, only starts consolidation after all items have been presented. Using these two consolidation skills, we created two versions of the AB-model, a “consolidate-separate” version and a “consolidate-chunk” version.

Finally, these two working memory task models provide the “retrieve” skill and the “respond” skill. The “retrieve” skill retrieves the appropriate consolidated item from memory and the “response” skill gives the appropriate response based on the retrieved item.

The four skills described above form the basic building blocks of both versions of our attentional blink model. To finalize the AB-model, the basic skills were put together in one model and were instantiated to fit the context of an AB-trial. This procedure was the same for both versions of the AB-model. In the AB-model, after presentation of a stimulus, the “search” skill checks, whether this is a target or a distractor. In other words, the `*fact-type` variable is instantiated with `letter`. If the stimulus is a distractor, it is ignored and the model waits for the next stimulus (`*next-stim` is instantiated with `wait`). If the stimulus is a target it switches to the consolidate skill (by instantiating `*after-found-target` with that skill) that moves the stimulus into a working memory slot. The consolidate skill is the source of the attentional blink in our model. Depending on which skill is used to accomplish consolidation, the model either starts consolidating directly after encountering the first target or postpones consolidation until the second target is encountered. If the chunk is consolidated, no other operator can be executed for a period of, on average, 200 ms (the imaginal delay parameter in ACT-R), leading to a possible attentional blink. If consolidation is postponed until the arrival of the second target, no attentional blink will occur at this point and the model will keep performing the task normally. After all stimuli are presented, the model will retrieve the targets that were consolidated on this trial (the “retrieve” skill) and then, after the retrieval, responding to the retrieved items (the “respond” skill).

Results

We compared the behavior of the models with human performance. This was done in order to verify the feasibility of the basic models and to check how well the final AB-model could model the AB phenomenon. The comparisons were made with existing data from the literature.

We did not find suitable data to which we could compare our visual search model. This is likely due to the fact that our visual search model is very simple and does not have any other functionalities besides what is described in the method section. Furthermore, the visual search model was not our primary interest, as it is not responsible for creating the AB.

Firstly, we will discuss the comparison between the SWM-model and human performance. The specific task we modeled required participants to remember a certain number of digits and report them at the end of a stream (Anderson et al., 1998). The critical manipulation in this experiment was that the digits were presented in multiple groups. This grouping was thought to influence chunking of the digits, digits grouped together during presentation would also be grouped together in memory (i.e., chunked together). The findings supported this expectation, such that participants showed longer reaction times during recall for the first item of a group, indicating that the groups were remembered (and recalled) as one chunk. The data from the simple working memory model showed this same pattern in reaction times as reported in Anderson et al. (1998).

As can be seen in Figure 1, the reaction times produced by the model show the same typical pattern as the human participants. This reflects the strategy used by the model (and presumably the participants) of recalling the remembered digits. The digits are stored in chunks of three in memory and this influences how the recall occurs. Firstly, the full chunk containing all three digits is retrieved from memory and, subsequently, the three responses are given without any further memory retrieval. Note however that the model is unable to capture the extra-long reaction times at the start of the recall-phase. These increased reaction times are likely due to processes relating to getting started on a new task, an aspect of the task unrelated to working memory so we chose not to model it at this moment.

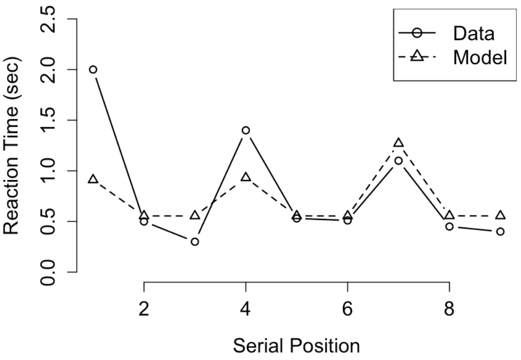


Figure 1: Model fit for reaction times in the SWM-task. Figure depicts the RTs produced by the model (dashed line) and human data (solid line).

Secondly, we will discuss the comparison between the CWM-model and human performance. In the task we modeled, a series of 3, 4, 5, or 6 digits were presented to the model. In between presentation of the digits, the model did a word-decision task in which it had to distinguish between nouns and adjectives. We compared the performance of our model on this task to a similar experimental task (Daily et al., 2001). In this task, participants were instructed to remember a series of digits (also 3 to 6), but here the digits were presented among letters which they were required to read aloud. Both of these tasks have in common that working memory is required to perform the interrupting task (either deciding between a noun or adjective or reading a letter aloud). This demand on working memory makes it impossible for the participants (and the model) to chunk the items in memory.

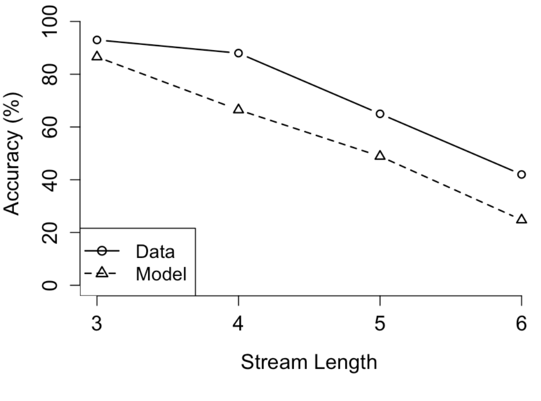


Figure 2: CWM-model fit for accuracy data. The average accuracy as a function of list length for the model (dashed line) and the human data (solid line).

We compared model performance with human performance with respect to accuracy (see Figure 2). Generally, the model shows a good fit to the human data reported by Daily et al. (2001). Both the model and the participants show decreased accuracy when the length of the presented list is longer. This decreased accuracy for longer lists occurs in the model because the presentation of the longer lists takes a longer time to be completed. The longer time required for presentation allows for additional item-decay in memory, leading to reduced accuracy for longer lists. The model, however, generally underestimates accuracy, this is probably due to the model being unable to capture the primacy effect (Murdock, 1962). The primacy effect is often modeled by including a rehearsal mechanism. The fact that we did not include such a mechanism to the model could thus explain the general underestimation of the accuracy.

Finally, we compared our AB-model (which resulted from the combination of the above discussed models) with human AB-performance (see Figure 3). The exact task we modeled is the classic AB-task reported in Chun & Potter, 1995. In this standard version of the AB, participants are instructed to identify two digits within a stream of distracting letters and, at the end of the stream, report which digits they have seen. We modeled this experiment with the version of the AB-model that used the "separate-consolidation" skill. The crucial effect in an AB-task is, unsurprisingly, the attentional blink itself. This refers to the strong performance decrement at lags 2 and 3, which our AB-model nicely captures. In the model, the AB occurs because consolidation of the first target (T1) is still in progress when the second target (T2) is presented. Therefore, T2 cannot be consolidated and will not be reported at the end of the stream. Our model also shows the typical lag-1 sparing effect. This is because consolidation of T1 often has not started at the moment that T2 is presented at lag 1. Therefore, they can both be consolidated into a single chunk and reported at the end of the stream. Finally, the model shows the slow performance increase for the later lags (lag 4 and higher). This is caused by the slow increase of the likelihood that T1 consolidation is finished by the time T2 is presented.

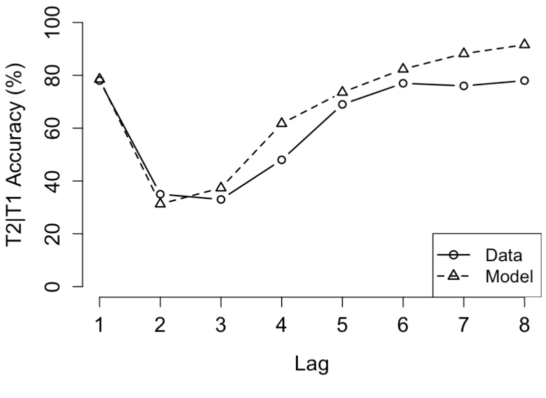


Figure 3: AB-model fit for T2 accuracy. Figure showing T2 accuracy in an AB-task for the model (dashed line) and human data (solid line).

Using the other version of the consolidation skill (the “consolidate-chunk” version) in the AB-model, however, will prompt the model to always try to consolidate both targets into a single chunk, thereby eliminating the AB all together. We compared the performance of the AB-model instantiated this way to the data from the study reporting a reduced AB when participants were instructed in a way that promoted chunking (Ferlazzo, Lucido, Di Nocera, Fagioli, & Sdoia, 2007) (see Figure 4). The model mirrored the general performance level and, crucially, showed no blink. The model, however, shows a slight performance decrease at lag 1. This is caused by our means of simulating noise in the visual system, which meant that occasionally T2 had already disappeared before it was processed fully and therefore it was missed. We do not consider this problematic, because in many AB experiments lag 1 performance is slightly lower than performance on long lags.

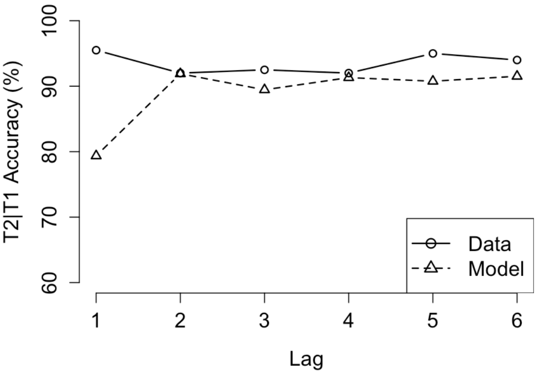


Figure 4: Model fit for the alternative AB model. Figure showing T2 accuracy for the alternative AB model (dashed line) and human data (solid line).

Discussion

Computational models of cognitive psychological phenomena are often able to accurately capture one specific phenomenon, however they are often hard to generalize to other tasks and cognition in general (Anderson et al., 2004). In this paper, we attempted to (partly) bridge this gap by employing a novel approach to building cognitive models, which mirrors the way people approach a new task. People do not consider every task in isolation but they use knowledge gained from the past. That is, they reuse skills learned from doing other tasks and apply them to the (new) task at hand (Salvucci, 2013; Taatgen, 2014). This paper describes our attempt to apply a similar approach. We created a new model of the attentional blink by reusing the models of other cognitive tasks. In short, we had two goals: (1) test the feasibility of the described approach and (2) create a model with the potential to shed new light on differences in AB due to instruction.

The comparisons between our models and human data show that our models are reasonably able to capture human performance. This result demonstrates the basic feasibility of the described modeling approach. It is possible to break a task down into a limited set of skills that are reusable in different tasks. This is an important first step towards creating more generalizable models, because it allows for a method of creating models that are built up from the same building blocks. Using existing building blocks when modeling a new task allows for much more integration of any new model into the already existing collection of models and, more importantly, might better reflect the way people approach a new task.

Note, however, that the devil is in the details. Building a model using this approach can be challenging, especially when it comes to determining how small differences between tasks should best be handled. Such differences make it difficult to use exactly the same operator (and therefore the same skill). Every operator has a condition-checking part (which checks whether this operator should be activated now) and an action-performance part (which actually executes the ‘cognitive action’ or PRIM). The action-performance part is relatively easy to generalize across tasks, but the condition-checking part is more challenging. Basically, the condition-checking part checks whether the situation matches the predefined situation in which this operator should be executed. This makes it difficult to generalize the condition-checking across tasks since a different task usually also means a different situation. We solved this problem in the models described in this paper by defining the conditions in such a way that they work for all the modeled tasks. This is a workable solution, but it is time-consuming and a better method for condition-checking is needed.

A further limitation to the models described here is that they did not perfectly capture all aspects of human performance. However, we do not see this as a major issue because we did not set out to create complete models of the described experimental paradigms. Instead we aimed to create models of the main findings only because we were merely interested in the skills that are important for the AB. Although there remain limitations and improvements to be made to the skill-based method, we consider it a feasible and promising approach to improve the generalizability of models.

The second goal we set out to achieve in this paper was to create a model of the AB that can account for differences due to instruction. The model described in this paper produces most of the basic effects from the classic AB-task, showing lag-1 sparing, the AB itself and the gradual improvement on later lags. Although there are many additional aspects of the AB reported in the extensive literature which we did not discuss, we believe that the model described here is an adequate first attempt that we will build on in future work.

For now, the fact that the model captured the basic AB-effects implies that these effects, at their core, may be caused by improper selection of skills. At the start of a new task, a participant has to figure out which skills to combine in order to be able to perform the new task. The models we created suggest that there are (at least) two different skills which can take care of the consolidation into working memory aspect of the task: (1) consolidate every presented target into working memory separately (as in the CWM-task) or (2) consolidate targets as larger chunks (as in the SWM-task). The chunk-consolidation skill as used in the SWM-task would be the optimal pick in this situation, two items can be consolidated into one chunk and there would be no negative unexpected effects. This approach is perhaps employed by participants after receiving the experimental instructions from the Ferlazzo et al. (2007) study. However, given that standard AB instructions consider targets as separate items probably prompts most participants to use the separate-consolidate skill from the CWM-task.

The emphasis put on strategy by our model could explain previous findings in the AB literature that have proven difficult to explain. This includes the effect of instructions as well as the existence of non-blinkers (individuals who do not show an AB) (Martens, Munneke, Smid, & Johnson, 2006), and the reduction of AB-magnitude because of training (Choi, Chang, Shibata, Sasaki, & Watanabe, 2012). All these effects could be explained by the type of consolidation strategy. Different instructions might cue the 'correct' consolidation skill, non-blinkers could be more naturally inclined to use the 'correct' chunking strategy compared to blinkers, and the training procedure by Choi and colleagues might nudge participants toward using the same optimal strategy.

To summarize, our novel skill-based approach to cognitive modeling resulted in valid models, created using a more natural and human-like method. In addition, we believe it shows great potential to generate more generalizable and thus more flexible models. Furthermore, it can lead to interesting new perspectives on well-established cognitive phenomena such as the AB. The choice of consolidation-strategy may play an important role in the AB, explaining individual differences as well as instruction and training effects of the AB.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An Integrated Theory of List Memory. *Journal of Memory and Language*, *38*(4), 341–380. https://doi.org/10.1006/jmla.1997.2553

Choi, H., Chang, L.-H., Shibata, K., Sasaki, Y., & Watanabe, T. (2012). Resetting capacity limitations revealed by long-lasting elimination of attentional blink through training. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(30), 12242–12247. https://doi.org/10.1073/pnas.1203972109

Chun, M. M., & Potter, M. C. (1995). A two-stage model for multiple target detection in rapid serial visual presentation. *Journal of Experimental Psychology: Human Perception and Performance*. US: American Psychological Association.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: a source activation account. *Cognitive Science*, *25*(3), 315–353.

Ferlazzo, F., Lucido, S., Di Nocera, F., Fagioli, S., & Sdoia, S. (2007). Switching between goals mediates the attentional blink effect. *Experimental Psychology*, *54*(2), 89–98. https://doi.org/10.1027/1618-3169.54.2.89

Martens, S., Munneke, J., Smid, H., & Johnson, A. (2006). Quick minds don't blink: electrophysiological correlates of individual differences in attentional selection. *Journal of Cognitive Neuroscience*, *18*(9), 1423–1438. https://doi.org/10.1162/jocn.2006.18.9.1423

Martens, S., & Wyble, B. (2010). The attentional blink: past, present, and future of a blind spot in perceptual awareness. *Neuroscience and Biobehavioral Reviews*, *34*(6), 947–957. https://doi.org/10.1016/j.neubiorev.2009.12.005

Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, *64*(5), 482.

Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary suppression of visual processing in an RSVP task: An attentional blink? *Journal of Experimental Psychology: Human Perception and Performance*. US: American Psychological Association.

Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, *37*(5), 829–860. https://doi.org/10.1111/cogs.12032

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*. US: Psychological Review Company.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*(3), 439–471. https://doi.org/10.1037/a0033138

Taatgen, N. A. (2014). Between architecture and model: Strategies for cognitive control. *Biologically Inspired Cognitive Architectures*, *8*, 132–139. https://doi.org/10.1016/j.bica.2014.03.010

Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology. General*, *137*(3), 548–565. https://doi.org/10.1037/0096-3445.137.3.548

Taatgen, N. A., Juvina, I., Schipper, M., Borst, J., & Martens, S. (2009). Too much control can hurt: A threaded cognition model of the Attentional Blink. *Cognitive psychology* (Vol. 59).

Discoveries of the Algebraic Mind: A PRIMs Model

Mark Y. Ji (y.ji@rug.nl)
Jacolien van Rij (j.c.van.rij@rug.nl)
Niels A. Taatgen (n.a.taatgen@rug.nl)
Department of Artificial Intelligence, University of Groningen
Nijenborgh 9, 9747 AG Groningen, Netherlands

Abstract

The PRIMs (primitive element) cognitive architecture addresses the issue of deterministic programming of production-rules (Taatgen, 2017). Motivated by infants' flexible discoveries of simple rule-like algebraic patterns (e.g., *a-a-b*, *a-b-a*, and *a-b-b* types of patterns, with variable individual syllable tokens), this study illustrates how the gradual integration of primitive operations to task-related contexts can be made possible through a reward-guided contextual learning mechanism. The promise of this prototypical model is demonstrated in its ability to (a) learn and generalize simple algebraic patterns, and (b) to account for infants' differential focusing time on a learned pattern and other unexposed new patterns. The modeled results are summarized from a developmental plasticity perspective.

Keywords: infant, discovery; robust; plastic; PRIMs.

Introduction

The ongoing presentation of distinct new micro-theories in cognitive psychology makes it difficult to see the forest for the trees (see Newell, 1990). Thanks to the advent of ACT-R (adaptive control of thought, rational; Anderson et al., 2004), seemingly disparate task-related aspects of cognition are now frameable within an overarching cognitive architecture. However, *a priori* programming of just one solution of a task again neglects other possible task-related solutions, and fails to capture the trial and error discovery processes observed in real-life task performance (see Taatgen, 2017). The need for modeling flexible discovery is especially motivated by infant learning, since young infants cannot be taught as how to complete a task but must arrive at their own solutions. In this paper, we present a model with particular focus on the flexible discoveries of simple algebraic patterns during infancy (see Marcus et al., 1999). We start by introducing the paradigm, and then briefly review previous models from a multilevel view (see Taatgen, 2017).

Infants seem innately capable of detecting simple algebraic patterns, and generalize them without relying on statistical features of the learned patterns. In an experiment reported by Marcus et al. (1999), infants with brief exposure to audible sequential presentation of the *a-b-a* or the *a-b-b* type of pattern (note the symbols *a* and *b* here refer to distinctive syllable tokens that are variable), during the test phase showed longer preferential focusing time for the other unexposed types of patterns as compared to the just learned pattern. This held even when each of the syllable token within the pattern were drawn from a different set (e.g., focusing longer for "*ko-ko-ga*" as compared to "*ko-ga-ko*" at test, after just exposing to "*le-we-le*"). Recent years have seen replication of this phenomenon with visually presented patterns (for a meta-analytical review, see Rabagliati, Ferguson, & Lew-Williams, 2019), and its extension to more complex variants of algebraic patterns (see Wilson et al., 2018). For instance, the acquisition of the *a-b-a* type of pattern is now considered a specific case of non-adjacent dependency learning, when an infant predicts that the first token *a* always matches the third token *b*.

With sensory perception as a point of departure, some modelers speculated that the acquisition of algebraic patterns is merely a basic form of feature detection (see McClelland & Plaut, 1999). Nevertheless, most models were built from a slightly higher level of abstraction (see Altmann, 2017), in assuming infants to be capable of forming representations from features (see Saffran & Thiessen, 2007). These representational models implicitly assume that young infants can deriving relational rules from a complex representation. However, very young infants cannot flexibly retrieve for instance a syllable "*le*" as parsed from a complex pattern of "*le-we-le*" (see Richmond & Nelson, 2007). Even when this constraint is suspended, a recent biologically-inspired representational model is only capable of generalize simple algebraic patterns at chance level (Alhama & Zuidema, 2018). This observation calls into question whether feature and/or representation alone are indeed sufficient or plausible in explaining the acquisition of simple algebraic patterns (see Dawson & Gerken, 2012; Frank & Tenenbaum, 2011).

On the contrary, models applying rule-based processes not only successfully modeled the learning of simple algebraic patterns (Seidenberg & Elman, 1999), but also stimulated a wide range of studies in the field of algebraic pattern acquisition (Frank & Tenenbaum, 2011). The Bayesian model of Frank and Tenenbaum (2011) demonstrated multiple algebraic solutions based on a hypothesis space of primordial rules. Moreover, the results modeled with this approach reflected emergent distinctions between *type*- and *token*-based processes (Frank & Tenenbaum, 2011). In other words, there is an empirical age-related distinction between young infants' early capability of detecting the *types* of *a-b-a*/*a-b-b* patterns; and their slightly delayed ability to detect the invariant *a-b* token pair in the *a-x-b* pattern where it is separated by a variable *x* (Dawson & Gerken, 2012). The distinction between *type*- and *token-based* processes are often interpreted in terms of the *exogenous*-to-*endogenous* transition, when early infants' passive *exogenous* reactions to the environment are gradually augmented by their active *endogenous* flexible retrieval of information as parsed from a complex representation (Diego-Balaguer et al., 2016).

How early infants might flexibly learn to recognize simple algebraic patterns remains to be explained. Emergent evidence now suggests that the infant brain possesses a modular architecture (see Dehaene-Lambertz & Spelke, 2015), thus calling for its conceptual implementation in studying the cognition of infants. Specifically, the *exogenous* reactions can be mapped to the passive encoding and comparisons at various modules, and the *endogenous* processes can be mapped to the active retrieval from declarative memory (Colombo & Cheatham, 2006; cf., Stocco & Anderson, 2008). Moreover, recent evidence indicates that the language-related prefrontal area is already functional during infancy (see Dehaene-Lambertz & Spelke, 2015), which can facilitate simple task-relevant processes such as the detection of syllable repetition (Bristow et al., 2009). Nakano et al. (2009) further reported selective activation of the prefrontal cortex in infants upon repetition of a syllable, and upon alteration of the syllable, which demonstrates inherent sensitivity of frontal structures to the establishment and alteration of the task requirement. It is possible that frontal activation follows a reward-guided mechanism that integrates and strengthens the currently acquired adaptive skills for future use (cf., Duncan, 2010).

Based on this empirical background, a modular and adaptive architecture is a well-suited tool for studying infant learning. Here, the PRIMs (primitive element) cognitive architecture is a promising candidate (Taatgen, 2013). It follows a modular structure pioneered by ACT-R, with additional prospects for the flexible discovery of rule-like patterns. This discovery mechanism is comprehensible from the perspective of functional development (see Bateson & Gluckman, 2011). To illustrate, initially randomly fired lower-level processes may occasionally lead to the successful detection of a repetition. This then entails a higher-level reward-guided mechanism that integrates various just applied lower-level operations to their associated task contexts, thus making them context-sensitive.

In this paper, we first aim to show how simple algebraic patterns can be acquired and generalized. Based on that, we attempt to account for the empirical findings in infants' focusing time differences in reacting to learned and other unexposed types of simple algebraic patterns.

Model

There follows a brief description of PRIMs operations at both the lower- and higher-levels.

Primitive Operations

The PRIMs cognitive architecture breaks down artificially programmed production-rules into elemental processes that can copy and compare information between separate slots in the input channel and the various memory modules (see Figure 1). These processes are called primitive operations, and they can be flexibly fired during task exploration.

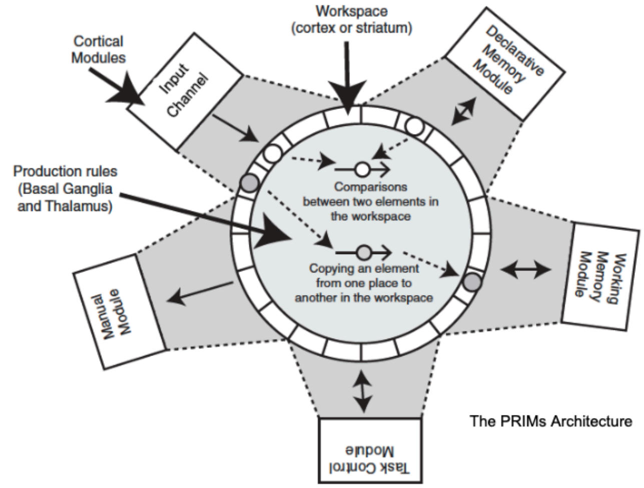

Workspace (cortex or striatum)
Cortical Modules
Input Channel
Declarative Memory Module
Production rules (Basal Ganglia and Thalamus)
Comparisons between two elements in the workspace
Copying an element from one place to another in the workspace
Working Memory Module
Manual Module
Task Control Module
The PRIMs Architecture

Figure 1: The PRIMs architecture for skill acquisition.

This PRIMs model of infant learning includes all possible lower-level primitive operations that encode (e.g., encode information within the input channel to the working memory module; see L_{encode}) or compare (e.g., compare whether information within the input channel matches to the working memory module; see $L_{compare}$) information between various modules (see Figure 1, Table 1, *ik* in table refers to $slot_k$ *in* $chunk_i$). However, a constraint is placed upon infants' processing capacity. This constraint acknowledges that the infants cannot yet simultaneously process multiple representations (e.g., encode/retrieve distinct representations "*le*" and "*we*" at the same time), and neither can they retrieve detailed information (e.g., syllable token "*le*") as parsed from a more complex representation (e.g., representation of the pattern "*le-we-le*"). When a condition is met for repetition detection, a scaffolding operation ($L_{scaffold}$) enables state transition to evaluation. Note that state transition to evaluation may also be flexibly entailed without scaffolding.

Table 1: Primitive operations.

L_{encode}	$input_{ik} \nRightarrow working/decl.\ memory_{ik}$	ignore
	$input_{ik} \Rightarrow working/decl.\ memory_{ik}$	encode
	$working/decl.\ memory_{ik} \Rightarrow control_{ik}$	encode
$L_{compare}$	$input_{ik} = working/decl.\ memory_{ik}$	compare
	$input_{ik} \neq working/decl.\ memory_{ik}$	compare
$L_{scaffold}$	"evaluation" $\Rightarrow control_{ik}$	transition

Reward-Guided Contextual Learning

In addition to the firing of lower-level operations, adaptive skills need to be arranged to satisfy and accomplish a defined task more efficiently. This is achieved by another higher-level evaluation operation. In this model, the evaluation operation is activated only when the presented stimulus at the input channel matches to the stored representation at the task control module ($H_{evaluation}$, see Table 2). This operation quickly entails a reward-guided contextual learning mechanism that reinforces the associations of just fired operations with their relevant task contexts – namely, *which* operation to fire at *what* context. For instance, to successfully detect a repetition in "*le-we-le*", the model always needs to encode the first token "*le*" with reference to its task contexts such as its general position "*first*" or its specific value "*le*". Gradually, the flexible firing of operations starts forming robust context-sensitive skills (i.e., encode-"*first*", or encode-"*le*"), which may be employed during relevant future contexts. Primitive operations can also be compiled to process more efficiently (e.g., "input ⇒ memory" and "memory ⇒ control" may be compiled into "input ⇒ control").

Table 2: Task-related operations.

$H_{evaluation}$	$input_{ik} = control_{ik}$	match
	$input_{ik} \neq control_{ik}$	mismatch

In this model, the weight of contextual association between a certain $operation_j$ and its relevant task context - in this case the specific syllable token stored in $slot_i$ of $module_k$ - is reflected in the following equation:

$$\Delta S_{jik} = \beta\ (\ S_{jik\ (current\ trial)} - S_{jik\ (previous\ trial)}\ )$$

in which, $S_{jik\ (current\ trial)} = default\ association \times (\ expected\ time - actual\ time\ )\ /\ expected\ time$

In this equation, *actual time* is the *actual* trial completion time, while *expected time* is hypothetically set initial trial completion time. The changing rate of the association weight S_{jik} is moderated both by (a) how efficiently the task was completed (when *actual time* < *expected time*) and (b) a learning rate parameter *β*. The *default association* sets the maximum weight for any contextual associations.

Default-Mode Operations

At the stage when the task is well learned, the firing of task-related operations become more efficient, increasing task-negative transitional spaces between them. The transitional spaces can become frequently occupied by the default-mode operation ($H_{default\text{-}mode}$, see Table 3), which are reinforced also by the contextual learning mechanism (cf., Smith et al., 2018). In other words, default-mode operation starts also to bind with task contexts whenever task-relevant operations are not active. The activation of default-mode operation is initially set at a low magnitude, but is gradually increased when it is more often fired and integrated also to the task contexts.

Table 3: Default-mode operations.

$H_{default\text{-}mode}$	$input_{ik}$ = "empty"	task-irrelevant processes

Methods

The modeled algebraic paradigm is adapted from Marcus et al. (1999, Exp. 2 and 3). In the first simulation, the focus is placed on the learnability of the *a-a-b*, *a-b-a*, and *a-b-b* patterns, each based on 100 model runs. The individual tokens of the trisyllabic pattern are presented each for 330 ms, with an ISI of 250 ms following each syllable token, and an ITI of 1000 ms following each pattern. The patterns are randomly drawn from a pool of 16 examples for each pattern type as adapted from Marcus et al. (1999, *a-b-a*/*a-b-b* type from Exp. 2 and *a-a-b* type from Exp. 3). The modeled trial is considered successful when repetition is detected during the evaluation operation (e.g., if $input_{ik} = control_{ik}$, estimated success = 1). This will in turn issue a reward to the model, strengthening associations of the manifestly adaptive lower-level operations with their relevant task contexts. Otherwise, the model is considered unsuccessful (e.g., if $input_{ik} \neq control_{ik}$ or no comparisons were made, estimated success = 0), and contextual associations for the operations during this trial will remain unchanged. To illustrate the gradual learning progression, 400 trials for each of the *a-b-a*, *a-b-b*, and *a-a-b* patterns are included.

Alternatively, simulations 2 and 3 focus on the generalization of the learned patterns, each based on 100 model runs. In the learning phase, the models are identical with the first simulation, except for the number of learning trials included. Specifically, simulations 2 was run for 150 learning trials, while simulation 3 was run for 500 trials to illustrate the effects of overlearning. The capacity of the model to generalize was then tested with novel examples of the learned pattern (e.g., *ko-ga-ko*) or other unexposed types of patterns (e.g., *ko-ga-ga, ko-ko-ga*). In the test phase that exams pattern generalization, specifications of primitive operations were not included. Instead, the models directly apply those operations and skills acquired from the learning phase to generalize them in the novel task contexts. To illustrate the trajectory of generalization, 150 transfer trials were included for each of the *a-a-b, a-b-a*, and *a-b-b* patterns.

Finally, the same learning and transfer models are applied in simulation 4 to illustrate critical differences in the empirical finding - in other words, infants' preferential longer focusing time on other unexposed types of patterns versus the learned pattern during test phases. The simulation consists of 100 learning and 10 transfer trials for each pattern, whereupon the frequency of default-mode operations during transfer trials is then calculated.

Results

Learning

Results of ***simulation 1*** demonstrate the model's ability to learn simple algebraic patterns (see Figure 2, averages with 95% CI error bars). Acquisition of all patterns converged to high percentages of correct predictions, albeit at different learning rates.

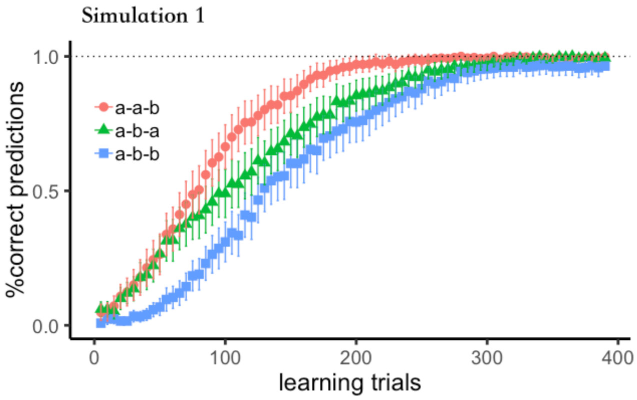


Figure 2: The discoveries of *a-a-b*, *a-b-a*, and *a-b-b* algebraic patterns. Horizontal axis shows learning trials from 1 to 400. Vertical axis shows the averaged percentages of correct predictions across 100 model runs (with 95% CI error bars).

It is easy to grasp that the learning of *a-b-a* is slightly more difficult than *a-a-b*, since irrelevance of second item *b* in *a-b-a* needs to be additionally acquired for repetition detection. However, it is less straightforward to explain the slower learning rate of *a-b-b*. This is nevertheless consistent with a recent finding showing 11-month-olds difficulty in detecting repetition in the *a-b-b-c* pattern (Schonberg, Marcus, & Johnson, 2017), and the slight advantage of initial versus late repetition (*a-a-b* versus *a-b-b*) in neonates at the neural level (Exp. 3, Gervain, Berent, & Werker, 2012). Note both findings were interpreted in terms of the primacy effect. Similarly, the simulation results similarly show a primacy effect at the skill level (see Figure 3). The first token in *a-b-b* must be "ignored" (**orange**) against the readily firing of various "encode" operations at the first position that are otherwise essential in learning *a-b-a* and *a-a-b*.

Another feature of the model is found in its ability to select an initial range of operations, while remaining capable of converging on to relatively invariant solutions when robust skills are formed (Figure 3). For instance, the model can flexible encode the item to the task control module, either from the working memory module (**brown**) or from the declarative memory module (**purple**). Nevertheless, when selection of the declarative memory route gradually organized into a robust state, it then becomes difficult to return to the initial flexible state in selecting an alternative working memory route. The modeled results in Figure 3 also revealed a gradual increase of default-mode operations (**blue**) when the selection of task-relevant operations gradually stabilize.

Generalization

Simulations 2 (150 training trials) and ***simulation 3*** (500 training trials) demonstrates the generalization of algebraic patterns from learning, based on 100 model runs (see Figure 4, averages with 95% CI error bars). Results in ***simulation 2*** show that an optimal level of learning facilitates transfer of a learned pattern for other novel patterns (Figures 4A, 4B, and 4C, cf., Taatgen, 2013). Note that the transfer rates are moderated also by the degree of difficulty to learn that pattern. To the contrary, modeled results of ***simulation 3*** predict hindrance of transfer due to overlearning (Figures 4A, 4B, and 4C). Although infants may not realistically be expected to participate in a prolonged learning session, overfitting to a particular context may still render the system less adaptive to a slightly altered context (e.g., a deterioration of prediction rates even for the same pattern with altered tokens).

Lastly, results of ***simulation 4*** shows a higher frequency of default-mode operation when a pattern has been learned (Figure 5, averages with 95% CI error bars). Default-mode operations may cause infants to divert from the task, and are therefore likely to have an inverse relation to the time they would be focusing on the task. These simulated results are consistent with the findings of Marcus et al. (1999).

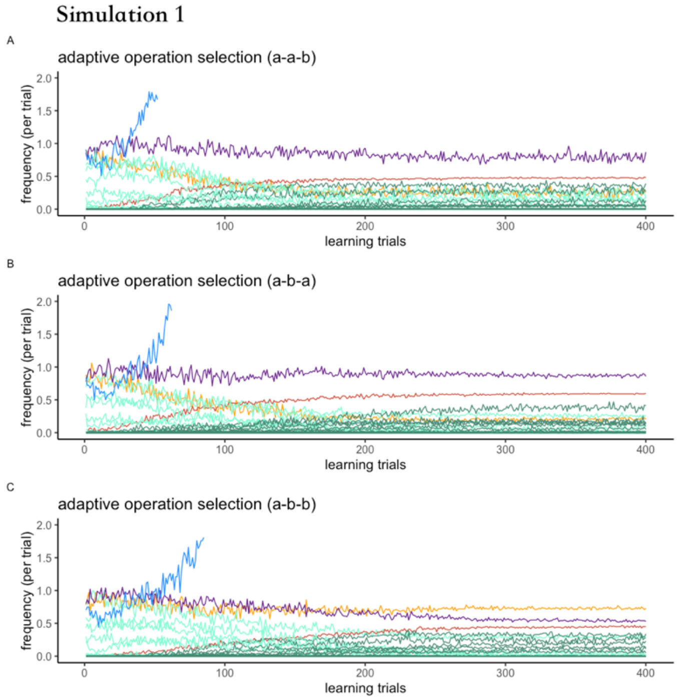


Figure 3: Operation selection over learning trials. Horizontal axis shows the number of learning trials. Vertical axis shows the frequency of various operations applied in a trial as averaged over 100 model runs (with 95% CI error bars). Color coding: **purple**, declarative mem. encode; **brown**, working mem. encode; **orange**, ignore; **light-green**, other primitive operations; **dark-green**, other compiled operations; **blue**, default-mode operations.

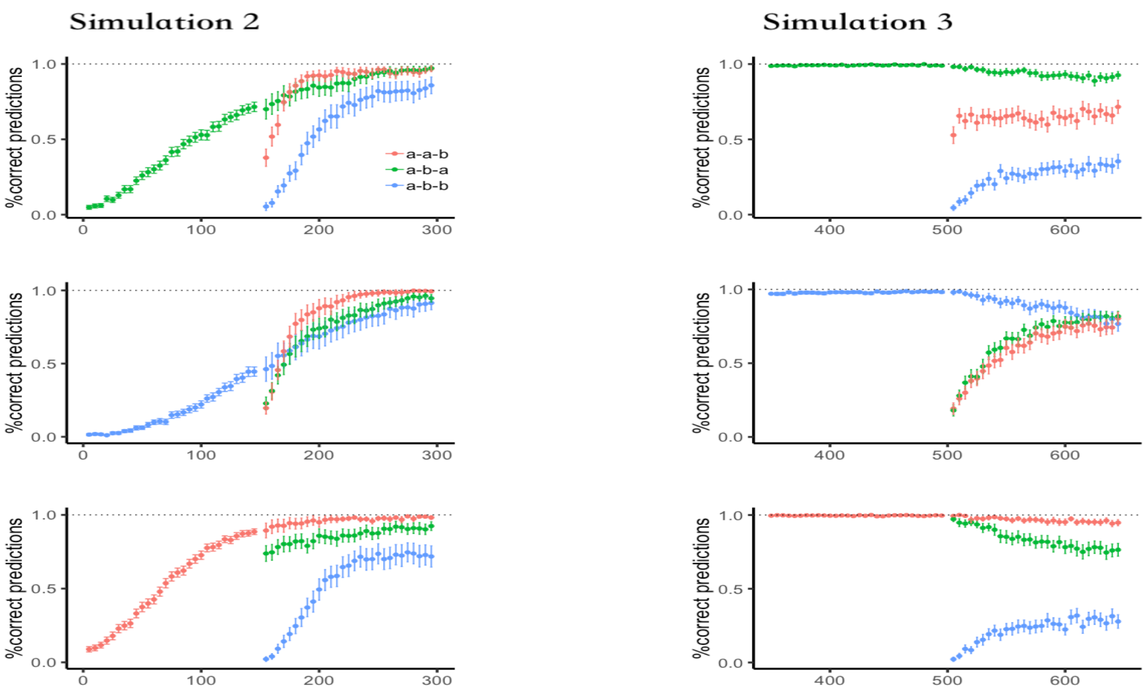


Figure 4. Generalization after learning or overlearning. Horizontal axis shows learning trials (150 trials in A, B, and C; 500 trials in D, E, and F) and the transfer trials (150 trials followed from the learning trials). Vertical axis shows the averaged percentages of correct predictions across 100 model runs (with 95% CI error bars).

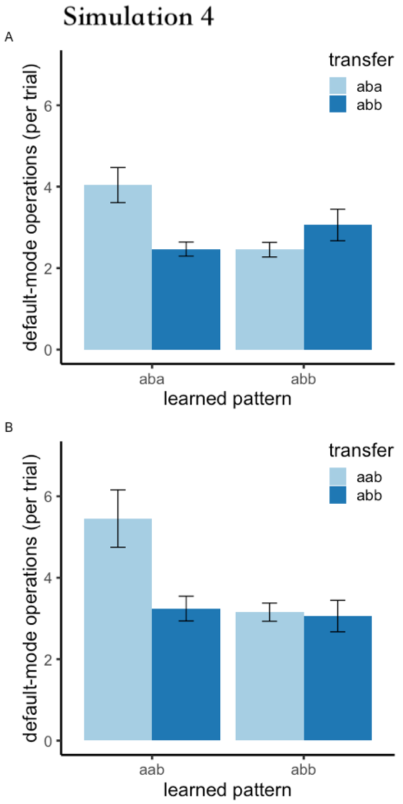


Figure 5: Frequencies of default-mode operations in transfer. Horizontal axis shows the learned type. The bar colors denote types applied in the 10 test trials on generalization. The vertical axis shows frequencies of default-mode operation per trial, as averaged across 100 model runs (with 95% CI error bars).

Discussion

An Aristotelian axiom *nihil est in intellectu quod non sit prius in sensu* holds that there is nothing in the intellect that was not originally derived from the senses. However, more recent literature on cognition in infants has disputed whether the detection of simple algebraic patterns is purely a lower-level statistical process or follows higher-level rules. Towards reconciling these two disparate views, results of our present PRIMs model suggest that seemingly rule-like patterns can be gradually acquired from the bottom-up. The promise of the model is reflected in its ability (a) to learn and generalize simple algebraic patterns (cf., Marcus et al., 1999; Schonberg, Marcus, & Johnson, 2017); and (b) to account for differences in infants' preferential focusing time on learned patterns versus other unexposed types of patterns (cf., Marcus et al., 1999). The modeled results may be framed in terms of a contemporary view on developmental plasticity (cf., Bateson & Gluckman, 2011).

Contemporary biology and psychology may be said to be correcting an earlier overemphasis on whether cognitive development is innate or learned. It is now clear that altering an innate property (e.g., presence or absence of certain trait-related genetic factors) is not always equatable with changes in learned characteristics. Instead, environmental conditions are crucial in shaping the precise characteristics of a learned skill (Bateson & Gluckman, 2011). The present PRIMs model demonstrates equal possibility of various routes in detecting syllable repetition. For instance, when flexible retrieval is not yet developed, infants can still distinguish between algebraic patterns (Dawson & Gerken, 2012).

Nonetheless, an innate structural architecture undoubtedly provides the basis for primitive operations to function.

Furthermore, distinct characteristics such as robustness and plasticity are not as cleanly separated as once thought. For instance, people maintains certain typical ways of dealing with a problem, but can also become flexible when the problem is changed. An emergent view now holds that robust outcomes can be derived from individual's plasticity (Bateson & Gluckman, 2011). This present PRIMs model shows that robust context-sensitive skills can be gradually integrated through a reward-guided contextual learning mechanism, and that the achieved robustness also raise barriers against the application of other possible skills that were currently not integrated. In addition, robust skills may be co-opted in other task contexts achieving generalization (see Taatgen, 2013). On the other hand, the present model also points to the detriments of overlearning and extreme robustness during one learning instance, which hinder the system for instance to accommodate the same type of pattern with just the syllable tokens altered. This extreme case may be taken as similar to a deterministically programmed model that only monotonously performs one single task.

Furthermore, the present PRIMs model suggest that the empirical finding concerning infants' shorter focusing time on the learned versus the other unexposed types of patterns may be product of the degree of robustness. Specifically, efficient processing of robust skills encourages the firing of default-mode operation, and gradually diverts the system from the focused task. In real terms, this may be associated with displacement of the infant's attention to need for food, comfort, play and so forth, curtailing the focusing time for the simple algebraic pattern. As illustrated from the present model, the accumulation of default-mode operations could occur whenever the system is still exploring the task. This in turn suggests that focusing time difference may not be directly relevant to how well an infant habituates a representation or masters a rule. Currently we are applying the same model to account for the counterintuitive reversed focusing time findings (longer focusing time on the learned versus the other patterns) to the generalization of non-adjacent dependency pattern *a-x-b* (Gómez & Maye, 2005).

Conclusion

Our PRIMs model firstly shows that simple algebraic patterns can be discovered bottom-up through the interplay between flexible primitive operations and a reward-guided contextual learning mechanism. This adaptive process produces robust context-sensitive skills that not only satisfies a given task, but may be also generalized in other relevant tasks. Secondly, the present study shows that infants' differential focusing time on the learned versus other unexposed types of pattern may be indirectly related to the robustness/plasticity of skill integration. In other words, efficient skill processing may encourage default-mode operation that reduces task focus. The modeled results suggest a more cautious position on drawing a direct link between infants' focusing times and the habituation/rule-bound operation of simple algebraic patterns.

References

Alhama, R. G., & Zuidema, W. (2018). Pre-Wiring and Pre-Training: What does a neural network need to learn truly general identity rules? *Journal of Artificial Intelligence Research, 61*, 927–946.

Altmann, G. T. (2017). Abstraction and generalization in statistical learning: implications for the relationship between semantic types and episodic tokens. *Philosophical Transactions of the Royal Society B: Biological Sciences, 372*, 20160060.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*, 1036.

Bateson, P., & Gluckman, P. (2012). Plasticity and robustness in development and evolution. *International Journal of Epidemiology, 41*, 219–223.

Bristow, D., Dehaene-Lambertz, G., Mattout, J., Soares, C., Gliga, T., & Baillet, S. (2009). Hearing faces: Crossmodal representations of speech in two-month-old infants. *Journal of Cognitive Neuroscience, 21,* 905–921.

Colombo, J., & Cheatham, C. L. (2006). The emergence and basis of endogenous attention in infancy and early childhood. *Advances in Child Development and Behavior, 34*, 283–322.

Dawson, C. R., & Gerken, L. (2012). Can rational models be good accounts of developmental change? The case of language development at two time scales. *Advances in Child Development and Behavior, 43*, 95–124.

Dehaene-Lambertz, G., & Spelke, E. S. (2015). The infancy of the human brain. *Neuron, 88*, 93–109.

Diego-Balaguer, R., Martinez-Alvarez, A., & Pons, F. (2016). Temporal attention as a scaffold for language development. *Frontiers in Psychology, 7*, 44.

Duncan, J. (2010). *How intelligence happens.* Yale University Press.

Gómez, R., & Maye, J. (2005). The developmental trajectory of nonadjacent dependency learning. *Infancy, 7*, 183–206.

Frank, M. C., & Tenenbaum, J. B. (2011). Three ideal observer models for rule learning in simple languages. *Cognition, 120*, 360–371.

Gervain, J., Berent, I., & Werker, J. F. (2012). Binding at birth: The newborn brain detects identity relations and sequential position in speech. Journal of *Cognitive Neuroscience, 24*, 564-574.

Marcus, G. F., Vijayan, S., Rao, S. B., & Vishton, P. M. (1999). Rule learning by seven-month-old infants. *Science, 283*, 77–80.

McClelland, J. L., & Plaut, D. C. (1999). Does generalization in infant learning implicate abstract algebra-like rules? *Trends in Cognitive Sciences, 3*, 166–167.

Nakano, T., Watanabe, H., Homae, F., & Taga, G. (2008). Prefrontal cortical involvement in young infants' analysis of novelty. *Cerebral Cortex, 19*, 455–463.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Rabagliati, H., Ferguson, B., & Lew-Williams, C. (2019). The profile of abstract rule learning in infancy: Meta-analytic and experimental evidence. *Developmental Science, 22*, e12704.

Saffran, J. R., & Thiessen, E. D. (2007). Domain-general learning capacities. In E. Hoff & M. Shatz (Eds.), *Blackwell handbook of language development* (pp. 68–86). Malden: Blackwell Publishing.

Schonberg, C., Marcus, G., & Johnson, S. (2017). The roles of item repetition and position in infant sequence learning. In: *CogSci 2017*. Cognitive Science Society.

Seidenberg, M. S., & Elman, J. L. (1999). Do infants learn grammar with algebra or statistics? *Science, 284*, 433–433.

Smith, V., Mitchell, D. J., & Duncan, J. (2018). Role of the default mode network in cognitive transitions. *Cerebral Cortex, 28*, 3685–3696.

Stocco, A., & Anderson, J. R. (2008). Endogenous control and task representation: an fMRI study in algebraic problem-solving. *Journal of Cognitive Neuroscience, 20*, 1300–1314.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review, 120*, 439–471.

Taatgen, N.A. (2017). Cognitive architectures: Innate or learned? *AAAI Fall Symposium Series,* 476–480.

Wilson, B., Spierings, M., Ravignani, A., Mueller, J. L., Mintz, T. H., Wijnen, F., van der Kant, A., Smith, A., & Rey, A. (2018). Non-adjacent Dependency Learning in Humans and Other Animals. *Topics in Cognitive Science*.

Flexible Timing with Delay Networks – The Scalar Property and Neural Scaling

Joost de Jong (j.de.jong.53@student.rug.nl)[1]
Aaron R. Voelker (arvoelke@uwaterloo.ca)[2]
Hedderik van Rijn (d.h.van.rijn@rug.nl)[1]
Terrence C. Stewart (tcstewar@uwaterloo.ca)[2]
Chris Eliasmith (celiasmith@uwaterloo.ca)[2]
[1]Experimental Psychology, Grote Kruisstraat 2/1, Groningen, 9712 TS, the Netherlands
[2]Centre for Theoretical Neuroscience, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada

Abstract

We propose a spiking recurrent neural network model of flexible human timing behavior based on the delay network. The well-known 'scalar property' of timing behavior arises from the model in a natural way, and critically depends on how many dimensions are used to represent the history of stimuli. The model also produces heterogeneous firing patterns that scale with the timed interval, consistent with available neural data. This suggests that the scalar property and neural scaling are tightly linked. Further extensions of the model are discussed that may capture additional behavior, such as continuative timing, temporal cognition, and learning how to time.

Keywords: Interval Timing; Scalar Property; Spiking Recurrent Neural Networks; Neural Engineering Framework; Delay Network

Introduction

Time is a fundamental dimension against which our mental lives play out: we remember the past, experience the present, and anticipate the future. Humans are sensitive to a wide range of temporal scales, from microseconds in sound localization to tens of hours in circadian rhythms. It is somewhere in between—on the order of hundreds of milliseconds to several seconds—that we consciously perceive time and coordinate actions within our environment (van Rijn, 2018). How does our brain represent time as accurately as possible, and how does it flexibly deal with different temporal intervals?

Scalar Property

Given the centrality of time to our experience, it is no wonder that timing and time perception have been the subject of extensive empirical study over the past 150 years. Many perceptual, cognitive, and neural mechanisms related to time perception have been studied, and perhaps the most well-known finding from the literature is the *scalar property* (Gibbon, 1977). The scalar property of variance states that the standard deviation of time estimates are linearly proportional to the mean of the estimated time. The scalar property has been confirmed by a wide variety of experimental data (Wearden & Lejeune, 2008). However, some research suggests that the scalar property does not always hold. It was already observed by Allan and Kristofferson (1974) that for well-practiced subjects in interval discrimination tasks, the standard deviation was constant for a range of relatively short intervals. Similar results were observed with pigeons, where the standard deviation remained flat for intervals up to around 500 ms (Fetterman & Killeen, 1992). Also, Grondin (2014) notes that the scalar property of variance critically depends on the range of intervals under consideration, and cites many examples with increases in slope after intervals of about 1.3 seconds.

Most models of timing take the scalar property as a starting point, or consider conformity to the scalar property as a crucial test. This seriously undermines their ability to explain violations of the scalar property. Here, we take the approach of not assuming the scalar property *a priori*, but instead construct a biologically plausible model that is trained to optimally represent time. We then systematically explore ranges of model parameters that lead the scalar property to be satisfied or violated, and provide a theoretical framework that aims to unify a variety of empirical observations.

Neural Scaling

Variance is not the only property of timing that scales with the estimated time interval. The firing patterns of individual neurons also stretch or compress proportional to the timed interval. In a recent study, Wang, Narain, Hosseini, and Jazayeri (2018) show that neurons in striatum and medial prefrontal cortex (mPFC) scale in this manner. During the timed interval, individual neurons display ramping, decaying, oscillating, or more complex firing patterns. In general, the specific shapes of temporal firing patterns for a given neuron remain the same, but become stretched for longer intervals and compressed for shorter intervals. Additionally, neurons in the thalamus display a different kind of scaling: their mean level of activity correlates with the timed interval. Both findings have been explained using a recurrent neural network (RNN) model (corresponding to neurons in striatum or mPFC) that receives a tonic input (originating from the thalamus) to scale the temporal dynamics of the network (Wang et al., 2018). The units in the neural network exhibit neural firing patterns and scaling similar to those observed experimentally. The model of timing we propose reproduces the same findings as the RNN model described in Wang et al. (2018). These findings suggest that, in order to perform timed actions as accurately as possible, the brain is able to flexibly scale its temporal dynamics. This implies a tight connection between the scalar property of variance and the temporal scaling of individual neurons.

Neural Models of Timing

Many neurally inspired models of timing and time perception have been proposed. Some models are based on ramping neural activity (Simen, Balci, deSouza, Cohen, & Holmes, 2011), some decaying neural activity (Shankar & Howard, 2010) and some on oscillating neural activity (Matell & Meck, 2004). Interestingly, all these neural firing patterns (and more complex ones) have been observed by Wang et al. (2018) in striatum and mPFC during a motor timing task. Therefore, appealing to only one of these neural firing patterns may be insufficient to fully explain timing performance. In line with this observation, the recurrent neural network model by Wang et al. (2018) exhibits a wide variety of firing patterns. However, their model does not show why this heterogeneity of firing patterns is important for timing performance or what the role is of ramping, decaying, or oscillating neurons in timing performance. Randomly-connected recurrent neural networks—referred to as *reservoir computers*—produce a wide variety of dynamics that can subsequently be extracted by a read-out population (Buonomano & Maass, 2009). A more structured approach to building a recurrent neural network may highlight the functional relevance of different neural firing patterns on timing performance.

One candidate for such a structured approach is the *delay network* (Voelker & Eliasmith, 2018). The delay network is a spiking recurrent neural network that approximates a rolling window of its input history by compressing the history into a q-dimensional state-vector. It has been observed that individual neurons in the delay network show responses similar to time-cells (MacDonald, Lepage, Eden, & Eichenbaum, 2011). Here, we use the delay network to explain both the scalar property of timing and the scaling of individual neural responses by comparing delay network data to empirical data from Wang et al. (2018).

Methods

We first discuss the mathematics behind the delay network. Then, we show how to implement the delay network as a spiking recurrent neural network using the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003). Lastly, we discuss the details of our simulations that follow the experimental setup of Wang et al. (2018).

The Delay Network

The delay network is a dynamical system that maintains a temporal memory of its input across a rolling window of θ seconds (Voelker & Eliasmith, 2018; Voelker, 2019). It does so by optimally compressing its input history into a q-dimensional state-vector. This vector continuously evolves through time in a way that captures the sliding window of history, while being amenable to representation by a population of spiking neurons using the NEF (as explained in the following subsection).

We consider the problem of computing the function $y(t) = u(t-\theta)$, where $u(t)$ is the input to the network, $y(t)$ is the output of the network, and $\theta > 0$ is the length of the window in time to be stored in memory. In order to compute such a function, the network must necessarily maintain a history of input across all intermediate moments in time, $u(t-\theta')$, for θ' ranging from the start of the window ($\theta' = 0$), going back in time to the end of the window ($\theta' = \theta$). This window must then slide forwards in time once $t > \theta$, thus always preserving the input over an interval of length θ. Computing this function in continuous time is challenging, as one cannot merely sample a finite number of time-points and shift them along; the time-step of the system could be arbitrarily small, or there may not even be an internal time-step as in the case of implementation on mixed-analog neuromorphic hardware (Neckar et al., 2019).

The approach taken by Voelker and Eliasmith (2018) is to convert this problem into a set of differential equations, $d\mathbf{x}/dt = \theta^{-1}(A\mathbf{x} + Bu)$, where $\mathbf{x}$ is a q-dimensional state-vector, and (A, B) are matrices governing the dynamics of $\mathbf{x}$. We use the (A, B) matrices from Voelker (2019; section 6.1.3). This results in the approximate reconstruction: $u(t-\theta') \approx \mathcal{P}(\theta'/\theta) \cdot \mathbf{x}(t)$, where $\mathcal{P}$ are the shifted Legendre polynomials. Importantly, the dimensionality q determines the quality of the approximation. This free parameter controls the number of polynomials used to represent the window – analogous to a Taylor series expansion of the input using polynomials up to degree $q-1$. Thus, q determines how much of the input's frequency spectrum, with respect to the period $1/\theta$, should be maintained in memory. Another notable property is that $1/\theta$ corresponds to a gain factor on the integration of $\mathbf{x}(t)$ that can be controlled in order to dynamically adjust the length of the window on-the-fly.

The Neural Engineering Framework (NEF)

Given this mathematical formulation of the computations that the neurons must perform in order to represent their past input, we turn to the question of how to recurrently connect neurons such that they perform this computation. For this, we use the NEF (Eliasmith & Anderson, 2003).

In the NEF, the activity of a group of neurons forms a distributed representation of some underlying vector space $\mathbf{x}$. In particular, each neuron i has some *encoder* (or preferred direction vector) $\mathbf{e}_i$ such that this neuron will fire most strongly when $\mathbf{x}$ is similar to $\mathbf{e}_i$. To produce heterogeneity in the neural population, each neuron has a randomly assigned gain α_i and bias β_i. Overall, the current entering each neuron would ideally be $\alpha_i \mathbf{e}_i \cdot \mathbf{x} + \beta_i$. This input current determines the spiking activity of the neuron, based on the neuron model. In this work, we use the standard leaky integrate-and-fire (LIF) model. This results in a pattern of neural activity over time $a_i(t)$ that encodes some continuous vector over time $\mathbf{x}(t)$.

If we have two groups of neurons, one representing $\mathbf{x}$ and one representing $\mathbf{y}$, and we want $\mathbf{y}$ to be some function of $\mathbf{x}$, then we can form connections from the first population to the second. In particular, we want to connect neuron i to neuron j with weights ω_{ij} such that the total sum from all the input connections will give the same result as the ideal equation assumed above. In other words, we want $\sum_i a_i(t)\omega_{ij} = \alpha_j \mathbf{e}_j \cdot \mathbf{y}(t)$ for all j (the bias β_j current is supplied separately). The ideal ω_{ij} are found using regularized least-squares optimization.

Furthermore, this method for finding connection weights can be extended to support *recurrent* connections (i.e., connections from the neurons in one population back to itself). These connections are solved for in the same manner, and, as has been shown Eliasmith and Anderson (2003), the resulting network approximates a dynamical system of the form $d\mathbf{x}/dt = f(\mathbf{x}) + g(\mathbf{u})$, where $\mathbf{x}$ is the vector represented by the group of neurons, $\mathbf{u}$ is the vector represented by the group of neurons providing input to this group, and the functions f and g depend on both the functions used to find the connection weights (as per the previous paragraph) and the temporal properties of the synapses involved (most importantly, the postsynaptic time constant).

The result is that the NEF provides a method for generating a population of neurons (to represent the q-dimensional state) and finding the ideal recurrent connections between those neurons such that they compute the differential equations required by the delay network.

It should be noted that the resulting network is structured exactly like a standard reservoir computer: a large number of neurons are recurrently connected, an input is supplied to that network, and we can decode information from the dynamics of the network by computing weighted sums of the overall neural activity. However, rather than randomly generating the recurrent weights, we are using the NEF to find the optimal weights for storing information over a rolling window in time. This method has been shown to be far more computationally efficient and accurate than various forms of reservoir computer for computing delays (Voelker, 2019).

An example of the resulting system is shown in Figure 1. Here the network is optimized to represent the past $\theta = 1$ s of its own input using $q = 6$ dimensions. Part A shows the (one-dimensional) input to the network over time. In this case, the input is a Gaussian bump centred at $t = 0.5$ seconds. The resulting neural activity (for 50 randomly-chosen neurons) is shown in Part B. Note that the neural activity at the beginning (before the input bump) and at the end (after $t > 1.5$ s) is fairly constant. This is the stable background activity of the network in the absence of any input. Since the network only stores the last second, in the absence of any input it will settle back to this state in ~ 1 second.

Part C shows one example of decoding information out of this network. In particular, we are decoding the function $y(t) = u(t - 0.5)$ – that is, the output should be the same as the input $\theta' = 0.5$ seconds ago. This output is found by computing the weighted sum of the spikes that best approximates this value, again, using least-squares optimization to find these weights. That is, $y(t) = \sum_i a_i(t) d_i$, where d_i is the decoding weight for the i^{th} neuron. We see that the network accurately represents the overall shape, although the Gaussian bump has become a bit wider, and the output dips to be slightly nega-

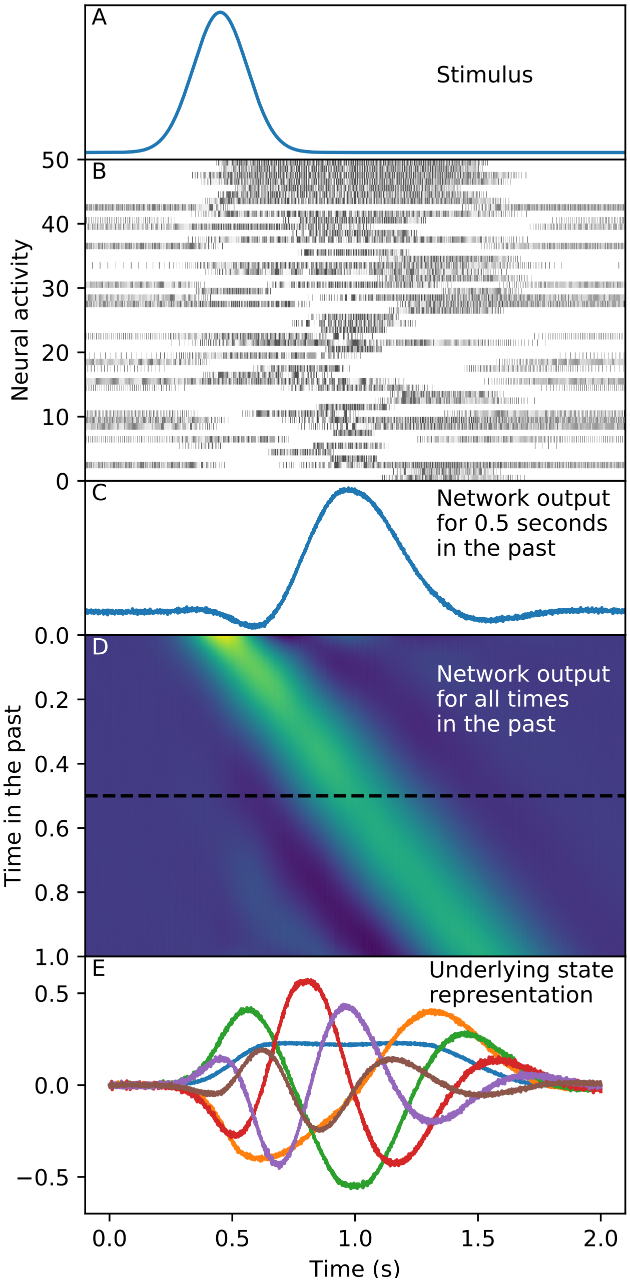


Figure 1: *The Delay Network – Optimized to represent the past* 1 *second of input history using* 6 *dimensions.* **(A):** The input to the network. **(B):** Neural activity of 50 randomly-chosen neurons within the network. **(C):** Decoding information from the network by taking the weighted sum of neural activity that best approximates the input from 0.5 seconds ago. **(D):** Decoding all information from the past 1 second. Each row is a different slice in time (from 0 to 1 second), and uses a different weighted sum of the same neural activity. The graph in part (C) is a slice through this image, indicated by a dotted line. **(E):** The underlying low-dimensional state information that represents the window.

tive before and after the bump. These are side-effects of the neurons approximating the ideal math for the delay network, and its compression of the input into 6 dimensions.

In Part D, we show the same process as in Part C, but for all times in the past from right now ($\theta' = 0$ s) to the furthest point back in time ($\theta' = 1$ s). This is to show that we can decode all different points in time in the past, and the particular case shown in Part C is just one example (indicated with a dotted line). Each of these different outputs uses the same underlying neural activity, but different decoders d_i out of the recurrent population.

Finally, Part E shows that we can also decode out the q-dimensional state representation $\mathbf{x}(t)$ that the delay network uses for its representation. These are the values that govern the dynamics of the delay network, and they form a nonlinear basis for all the possible functions that could be decoded out from the neural activity. Indeed, each row in Part D can also be interpreted as a different linear transformation of the data shown in Part E. Voelker and Eliasmith (2018) derive the closed-form mathematical expression that provides such a transformation, thus relating all time-points within the rolling window to this underlying state-vector.

These different views of the delay network can be seen as a very clear example of David Marr's Tri-Level Hypothesis (Marr, 1982), which we use here to understand this system at varying levels of abstraction. For instance, we may consider only the *implementational level*, which consists of leaky integrate-and-fire neurons with recurrent connection weights between them, a set of input weights from the stimulus, and multiple sets of output weights. Or we may consider the *algorithmic level*, where the system is representing a q-dimensional state-vector $\mathbf{x}$ and changing that vector over time according to the differential equations given in the previous section. Or we may consider the *computational level*, where the network is storing a (compressed) history of its own input, and different slices of that input can be extracted from that memory. All of these are correct characterizations of the same system.

Simulation Experiment

In the original experiment by Wang et al. (2018), monkeys were presented with a "cue" signal that indicated the interval to be reproduced: red for a short interval (800 ms) and blue for a long interval (1500 ms). Then, they were presented with a "set" signal that marked the start of the interval. The monkeys had to issue a response after the cued interval had elapsed. We have attempted to match the relevant details of their experimental setup as follows. The delay network (with $q = 4$) continually receives input from a control population that scales θ in order to produce intervals around 800 ms or 1500 ms. In effect, this gain population controls the length of the window on-the-fly. The effective value of θ is 1, divided by the value that the gain population represents. When the value represented by the gain population is greater than 1, it makes the length of the window shorter; when it is smaller than 1, it makes the window longer. This enables us to choose values for the gain population that will let the delay network time intervals around 800 ms or 1500 ms. The delay network receives input that is continually represented, along with the history of this input. The input signal is a rectangular impulse of 500 ms. The same read-out population decodes the delayed input signal as θ is varied.

Results

Scalar Property in the Delay Network

In order to quantify the scalar property in the spiking implementation of the delay network, we calculated the mean and standard deviation of the decoded output at θ seconds. We performed this analysis for delay networks with a range of values for θ and q while keeping the number of neurons per dimension fixed at 500. We considered only positive values around the peak of the decoded output. If the scalar property holds, we should observe a linear relationship between θ and the standard deviation of the impulse response. Our data suggests that the scalar property critically depends on q (Figure 2). The relationship between the standard deviation, θ, and q can be described as follows. The standard deviation remains constant for a range of short intervals and starts to increase linearly after some value of θ. Both the location of this transition and the slope of the linear increase depend on q. This helps explain some previous differences in experimental findings. For example, the flat standard deviation for ≤ 500 ms observed by Fetterman and Killeen (1992) can be explained by assuming that $q = 2$ within our model.

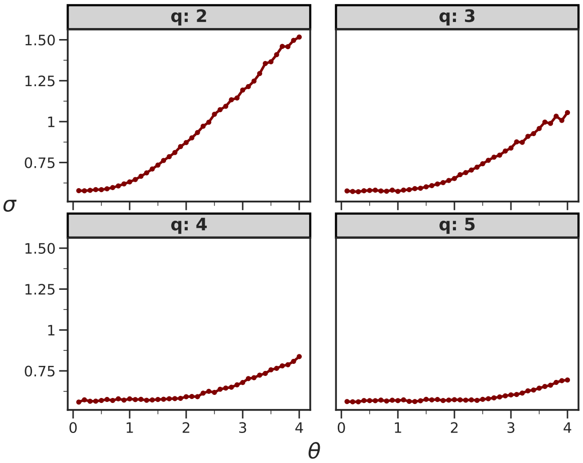


Figure 2: *Scalar Property*. The standard deviation of the impulse response plotted against θ for different values of q.

Neural Scaling in the Delay Network

Our simulations of the Wang et al. (2018) experiment produced results with a qualitative fit to the empirical data (Figure 3). First, the standard deviation of the decoded output increased with θ (also see previous section). Second, the neural responses were highly heterogeneous, with ramping, decaying, and oscillating neurons. These firing profiles were observed because they are linear combinations of the underlying state vector $\mathbf{x}(t)$ (see Figure 1E). Third, the responses of individual neurons stretched or compressed with the length of the timed response response, similar to the empirical data from Wang et al. (2018).

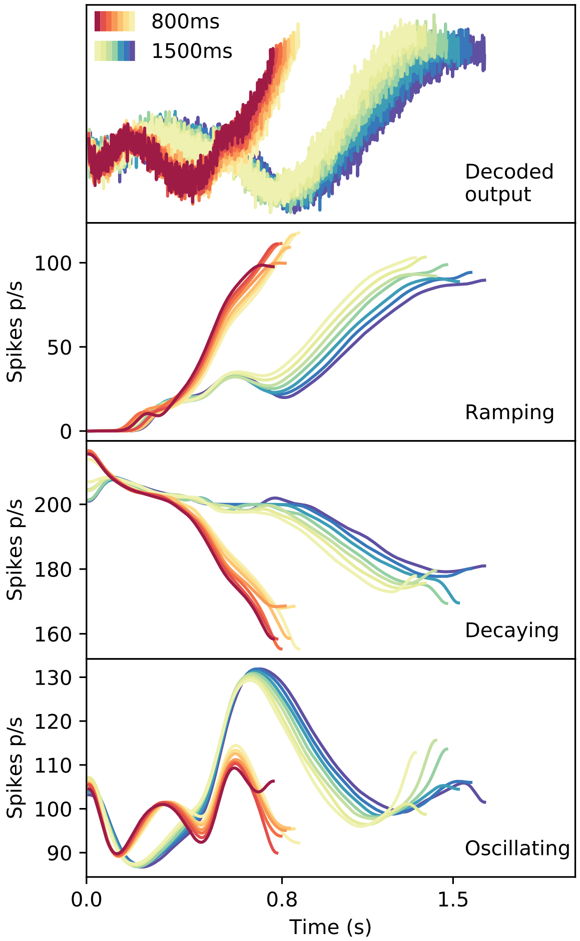


Figure 3: *Neural Scaling*. A square input was provided to the delay network, while varying the value of the gain input. The peak and standard deviation of the decoded output scale with the gain. The heterogeneous firing patterns of individual neurons also scale with gain. Here, neural firing patterns of three example neurons are shown that qualitatively fit the data from Wang et al. (2018). We focused on the first period of the neural response to the "set" stimulus. The top neuron shows ramping activity, the middle neuron shows decaying activity, and the bottom neuron shows oscillatory activity.

Discussion

The aim of the present study is to use the delay network to explain two findings in the timing literature: the scalar property of variance and neural scaling. We did not assume the scalar property *a priori*, but systematically explored the parameters of the delay network that lead the scalar property to be satisfied or violated. Our results suggest that the scalar property critically depends on q. Notably, the time-cell data that was analyzed in earlier work fit best for $q = 6$ (Voelker & Eliasmith, 2018). The temporal range of conformity to the scalar property and slope of the scalar property may be explained by the number of dimensions the delay network uses (q): for higher q, the range of short intervals with a constant standard deviation increases, whereas the slope of the scalar property decreases. We also found that scaling the dynamics of the delay network produces scaling of neural firing patterns, matching empirical data (Wang et al., 2018). Our model suggests that when the delay network represents its input history with more dimensions, neural firing patterns become more complex, as additional linear combinations of higher-degree Legendre polynomials are encoded by individual neurons. Furthermore, these findings suggest that the scalar property and the adaptive control of neural dynamics are tightly linked.

Previous Models

The delay network shares some features with previous neural models of timing, but there are also critical differences. First, similar to previous RNN models, the delay network is an RNN that uses population-level dynamics to time intervals. However, previous RNN models use a random connectivity approach to generate the necessary dynamics for accurate timing, whereas the delay network explicitly defines the required dynamics and optimizes neural connectivity to implement those dynamics. Also, previous RNN models of timing do not characterize how the input history is represented. Similar to memory models of timing (Shankar & Howard, 2010), the delay network makes this connection explicit. Even though memory models and the delay network both specify how input history is represented, the memory models do not specify how to optimally scale the dynamics of the network or compute arbitrary functions over the represented history. In contrast, the delay network is optimized to recurrently represent time, and comes with a general framework that links the input history, network representation, and spiking neural activity. In sum, we believe that the delay network is an improvement over previous models of timing by both explicitly specifying how time is represented and implementing that representation in a flexible neural framework.

Extending the Delay Network

In this work, we have used the delay network to explain the scalar property and neural scaling in a simple motor timing task. However, the delay network may be used to explain a wide variety of timing phenomena, including: continuative timing, temporal cognition, and learning how to time.

Continuative Timing First, the delay network can be extended to account for time perception in a wide variety of realistic situations. A classic dichotomy in the timing literature is between prospective and retrospective timing. Prospective timing is explicitly estimating an interval with knowledge beforehand that attention should be focused on time. On the other hand, retrospective timing is estimating, in hindsight, how long ago an event happened. However, this distinction may be arbitrary, since in realistic situations, one often notices the duration of an ongoing interval. For instance, you may notice that a web page is taking too long to load but wait an additional amount of time before checking your signal reception. When this happens, one neither has earlier knowledge that time should be attended to (prospective) nor the instruction to estimate how much time has passed since an event (retrospective). Therefore, a more appropriate term for timing in realistic situations would be continuative timing (van Rijn, 2018). The delay network, at any point in time, serves as a rich source of information regarding the temporal structure of ongoing events, including how long ago an event started and stopped. This information can be used to infer how much time has elapsed since a salient event and compared to the typical temporal structure of an event in memory. Such comparisons could then facilitate decision-making, such as in deciding whether to wait for an additional amount of time.

Temporal Cognition Second, time is a crucial factor in a wide variety of cognitive processes. Timing models have been successfully integrated in ACT-R (Taatgen, van Rijn, & Anderson, 2007) and models of decision-making (Balc & Simen, 2016). The delay network, built with the NEF, is compatible with other cognitive models that have been developed in the same framework, or indeed any cognitive models that can incorporate neural networks. Therefore, a future avenue of research will be to incorporate the delay network into existing models of cognitive processes, such as action-selection (Stewart, Bekolay, & Eliasmith, 2012) and working memory (Singh & Eliasmith, 2006).

Learning to Time Third, the delay network may be used to explain how timing is learned. In the experiment by Wang et al. (2018), the monkeys trained extensively before they could accurately perform the motor timing task. The monkeys received rewards according to the accuracy of their performance. Another open question is how an optimal mapping between cues and the gain population can be learned. Therefore, future work will focus on modeling how timing is mastered during reinforcement learning.

References

Allan, L. G., & Kristofferson, A. B. (1974). Psychophysical theories of duration discrimination. *Perception & Psychophysics*, *16*(1), 26–34.

Balc, F., & Simen, P. (2016). A decision model of timing. *Current Opinion in Behavioral Sciences*, *8*, 94–101.

Buonomano, D. V., & Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, *10*(2), 113–125.

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: computation, representation, and dynamics in neurobiological systems*. Cambridge, Mass: MIT Press.

Fetterman, J. G., & Killeen, P. R. (1992). Time discrimination in Columba livia and Homo sapiens. *Journal of Experimental Psychology: Animal Behavior Processes*, *18*(1), 80–94.

Gibbon, J. (1977). Scalar Expectancy Theory and Weber's Law in Animal Timing. *Psychological Review*, *84*(3), 279–325.

Grondin, S. (2014). About the (Non)scalar Property for Time Perception. In H. Merchant & V. de Lafuente (Eds.), *Neurobiology of Interval Timing* (Vol. 829, pp. 17–32). New York, NY: Springer New York.

MacDonald, C., Lepage, K., Eden, U., & Eichenbaum, H. (2011). Hippocampal Time Cells Bridge the Gap in Memory for Discontiguous Events. *Neuron*, *71*(4), 737–749.

Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. San Francisco: W.H. Freeman.

Matell, M. S., & Meck, W. H. (2004). Cortico-striatal circuits and interval timing: coincidence detection of oscillatory processes. *Cognitive Brain Research*, *21*(2), 139–170.

Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., ... Boahen, K. (2019). Braindrop: A Mixed-Signal Neuromorphic Architecture With a Dynamical Systems-Based Programming Model. *Proceedings of the IEEE*, *107*(1), 144–164.

Shankar, K. H., & Howard, M. W. (2010). Timing using temporal context. *Brain Research*, *1365*, 3–17.

Simen, P., Balci, F., deSouza, L., Cohen, J. D., & Holmes, P. (2011). A Model of Interval Timing by Neural Integration. *Journal of Neuroscience*, *31*(25), 9238–9253.

Singh, R., & Eliasmith, C. (2006). Higher-Dimensional Neurons Explain the Tuning and Dynamics of Working Memory Cells. *Journal of Neuroscience*, *26*(14), 3667–3678.

Stewart, T. C., Bekolay, T., & Eliasmith, C. (2012). Learning to Select Actions with Spiking Neurons in the Basal Ganglia. *Frontiers in Neuroscience*, *6*.

Taatgen, N. A., van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review*, *114*(3), 577–598.

van Rijn, H. (2018). Towards Ecologically Valid Interval Timing. *Trends in Cognitive Sciences*, *22*(10), 850–852.

Voelker, A. R. (2019). *Dynamical systems in spiking neuromorphic hardware*. Unpublished doctoral dissertation, University of Waterloo, Waterloo, ON.

Voelker, A. R., & Eliasmith, C. (2018). Improving Spiking Dynamical Networks: Accurate Delays, Higher-Order Synapses, and Time Cells. *Neural Computation*, *30*(3), 569–609.

Wang, J., Narain, D., Hosseini, E. A., & Jazayeri, M. (2018). Flexible timing by temporal scaling of cortical responses. *Nature Neuroscience*, *21*(1), 102–110.

Wearden, J. H., & Lejeune, H. (2008). Scalar Properties in Human Timing: Conformity and Violations. *Quarterly Journal of Experimental Psychology*, *61*(4), 569–587.

ACT-R model for cognitive assistance in handling flight deck alerts

Oliver W. Klaproth (Oliver.Klaproth@Airbus.Com)
Central Research & Technology, Hein-Sass-Weg 22
21129 Hamburg Germany

Marc Halbrügge (Marc.Halbruegge@ACM.Org)

Nele Russwinkel (Nele.Russwinkel@TUBerlin.De)
Chair of Cognitive Modeling in Dynamic Human Machine Systems, Marchstr. 23
10587 Berlin Germany

Abstract

The ability to respond to the needs of an individual operator is key for cognitive assistance in naturalistic settings. In order to keep track of changing operator demands in dynamic situations, a model-based approach for cognitive assistance is proposed. Based on model tracing with flight deck interactions and EEG recordings, the model is able to represent individual pilots' behavior in response to flight deck alerts. As a first application of the concept, an ACT-R cognitive model is created using data from an empirical flight simulator study on neurophysiological signals of missed acoustic alerts. Results show that uncertainty of individual behavior representation can be significantly reduced by the combination of cognitive modeling and EEG data. Implications for model-based cognitive assistance in flight deck operations are discussed.

Keywords: Cognitive modeling; flight deck alerts; model-based cognitive assistance; model-tracing; neuroadaptive technology;

Introduction

Individual user behavior

Representing individual user behavior is a challenge for cognitive modeling. Most models aim to simulate average user behavior under controlled conditions instead of individual performance in complex tasks (Rehling, Lovett, Lebiere, Reder, & Demiral, 2004). Representing individual behavior in naturalistic settings requires dealing with multiple sources of variation such as inter-individual differences (e.g., architectural and knowledge differences; Taatgen, 1999) and uncontrolled external factors of the situation. For example, when modeling pilot performance in commercial aviation, different levels of experience and changing weather conditions would need to be considered. A cognitive model that is able to keep track of the operational context and an individual users' cognitive dynamics can serve as the basis for cognitive assistance in operations (Zhang, Russwinkel, & Prezenski, 2018).

Cognitive assistance is about providing the right information at the right time. The quality of support that can be provided therefore depends on what is and can be known about the task environment and the operator's cognitive processes. In naturalistic situations, very extensive models would be needed to incorporate all sources of variation for explaining individual performance in a deterministic fashion. Regardless of the feasibility of such modeling, understandability of the model would be traded in for completeness, also known as "Bonini's paradox" (Dutton & Starbuck, 1971). Alternatively, leaner models would introduce epistemic uncertainty (Kiureghian & Ditlevsen, 2009), leaving specific aspects of behavior unexplained due to a model's lack of knowledge. A number of methods have been used to reduce epistemic uncertainty caused by individual differences, such as pre-test scores as predictors (Rehling et al., 2004), model tracing (Fu et al., 2006), inserting physiological data on user's workload into the model (Putze, Schultz, & Propper, 2015) and dynamic adjustment of parameters with pre-computed lookup tables (Fisher, Walsh, Blaha, Gunzelmann, & Veksler, 2016).

Cognitive assistance in aviation

Inattentional deafness leads to performance drops in the cockpit (Dehais, Roy, & Scannella, 2019) that can benefit from cognitive assistance, e.g. in the form of verbal reminders (Estes et al., 2016). Causes and consequences of overheard messages for individual pilots' performance need to be considered to identify the right information to be provided and the right timing to provide it for cognitive assistance in operations.

Causes can be diverse and situation dependent (e.g., perceptual/attentional factors, see Dehais et al., 2019) and are likely too complex for deterministic modeling of single occurrences of missed alerts. Often, alerts are declared as missed when pilots fail to react. Knowing what made a pilot fail to react or what pieces of information he or she was unable to process gives diagnostic value and helps to identify adequate means of support. For cognitive assistance in handling flight deck alerts, information about a message's contents and whether it was processed by the pilot is a viable alternative to complex models required for deterministic prediction of user states.

Consequences of an overheard or ignored message for pilots' performance can be anticipated with the help of a cognitive pilot model. ACT-R (Anderson et al., 2004) is a comprehensive and scientifically substantiated cognitive architecture that has produced models representing processes e.g. involved in "manual" flight control of single engine aircraft (Somers & West, 2013), visual attention allocation in a glass cockpit (Byrne et al., 2004) and the use of and skill acquisition for the flight management system (Schoppek & Boehm-Davis, 2004; Taatgen, Huss, & Anderson, 2008). For model-based assistance such formal descriptions of flight related tasks and processes can describe what constitutes normative performance.

Neuroadaptive cognitive model

In the present paper a modeling concept is proposed that is able to explain uncertainty in single instances of missed alerts by representing individual pilots' behavior. In the fashion of Putze et al. (2015) we extend the idea of model tracing (Fu et al., 2006) by incorporating physiological data. Whereas Putze et al. (2015) integrate physiological data to model architectural differences, i.e. occupying cognitive resources with a dummy model to model workload, the concept in this paper focuses on modeling knowledge differences (Taatgen, 1999) due to unprocessed auditory messages.

Model tracing based on monitoring pilot interactions with flight deck instruments enables the model to identify when performance deviates from normative behavior. Based on such deviations, the model can make inferences about the pilot's cognitive states. By treating instances of deviating behavior as situations it cannot explain due to lack of knowledge, the model consults external sources of information, i.e. event-related physiological data of the pilot it tries to represent.

Physiological measurements, e.g. electroencephalography (EEG) can provide information about cognitive operations. With a passive brain computer interface (Zander & Kothe, 2011) EEG can be recorded without interfering with the task and data can be processed in (almost) real-time. The integration of these data into the model allows for more refined representations of individual pilots. Such a neuroadaptive (Zander, Krol, Birbaumer, & Gramann, 2016) cognitive model would be able to adjust its generic or normative behavior to measurements of a pilot's current cognitive state and to identify current needs for assistance.

Physiological measures can be subject to errors that introduce intrinsic or aleatory uncertainty (Kiureghian & Ditlevsen, 2009). Whereas epistemic uncertainty represents defined model boundaries, aleatory uncertainty is hard to identify in single situations where there is no ground truth available. That is, the model is able to identify situations of deviating behavior, but it cannot say which of the physiological data are affected by measurement or classification error and which are not. In model-based cognitive assistance, thoughtful handling of the two types of uncertainty is required (see Figure 1 for an overview of type of uncertainty introduced by data source).

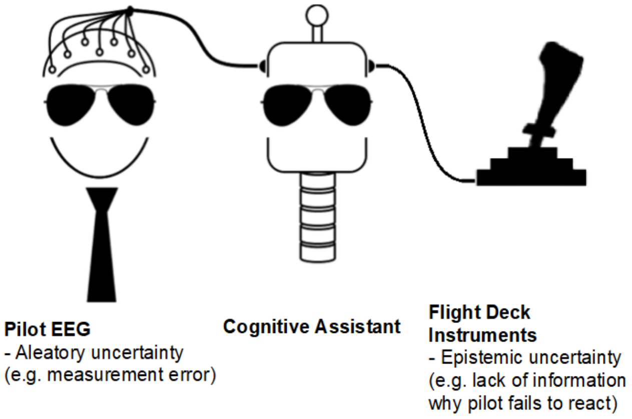


Figure 1: Sources of uncertainty in neuroadaptive concept

The objective of this study is to increase the effectiveness of modeling individual pilot behavior in response to flight deck alerts. For increased effectiveness, model tracing and EEG recordings are used to reduce uncertainty due to individual differences. Behavioral data from an empirical study on the neurophysiological reaction to auditory signals in simulated flight (Krol et al., 2018) are modeled to demonstrate how the proposed concept can be implemented. Accuracies of a neuroadaptive cognitive pilot model and normative model are compared to quantify the fraction of uncertainty reduced by inserting pilots' EEG data. Epistemic and aleatory uncertainty are quantified and examined regarding their implications for model-based cognitive assistance in flight operations.

Methods

Empirical study

21 air crew (one female) who were predominantly military pilots participated in the empirical flight simulator study. Participants had a mean age of 49.08 years (*SD* = 6.08) and an average experience of 3230 hours of flight (*SD* = 2330.71). All participating air crew had normal or corrected to normal vision, all but two were right-handed. Air crew were seated in a fixed base experimental flight simulator in single pilot setup that approximated Airbus A320 cockpit design. Participants were asked to perform an 18 minutes scenario that consisted of 9-14 events resembling flight deck alerts per participant, each preceded by auditory warnings or air traffic control (ATC) messages. The scenario had to be flown by selecting heading and altitude on the auto flight system according to ATC instructions. In addition, participants were asked to manage thrust manually and attend to alerts. Alerts included in the scenario could have low ("amber alert", e.g. fuel pump failure) or high priority ("red alert", e.g. engine fire) and ATC messages contained navigation or speed instructions. Speed warnings were issued dynamically whenever participants left a speed threshold area, which resulted in different numbers of acoustic events per participant. For the scenario, the open

source flight simulation software “FlightGear 3.4”[1] was used. Essential instrument properties and state changes in the scenario were recorded in log files with a sampling rate of 20 Hz.

Before the flight scenario, participants’ EEG was recorded while performing an auditory oddball paradigm (frequent versus rare sounds). A classification algorithm was trained on the EEG data to recognize activity patterns for processing of target (i.e. processed alerts) and standard sounds (missed alerts). The algorithm was tuned to have equal chances for false alarms and misses in case of incorrect response classification. Due to the frequent use of standard compared to rare target sounds in the training paradigm, classifier accuracy needs to be higher than 0.78 to perform significantly better than chance. EEG was recorded during classifier training and scenario with a 32 channel BrainProducts LiveAmp system.

Cognitive modeling

ACT-R was used to create a cognitive model to represent individual pilot’s behavior. ACT-R consists of memory, perceptual and motor modules that interact with each other by exchanging chunks of information through buffers. The declarative memory module can hold and store information about the task state, whereas procedural memory allows for modeling productions (condition-action-statements) that apply depending on the state of the task or the environment. Perceptual and motor modules allow for modeling of basic sensory processes and enable a model to interact with the environment. When modeling pilot activities, the respective modules can be used to represent storing and updating flight information such as altitude and speed, procedures for how to react in case of alerts, and auditory and visual perception of messages in the cockpit.

For assistance in operations, a cognitive pilot model needs to be flexible, adaptive at runtime and knowledgeable of the operational context. Not only does it need to know what constitutes optimal or normative performance of a task, but also alternative means to meet the objective. In case of deviations from normative performance, it has to be able to adapt its functionality and adjust its representation of the pilot. Finally, the model needs to be able to anticipate the consequences of both normative and alternative performance in a task so it can offer support when needed.

A scenario specific hierarchical task analysis (HTA; Stanton, 2006) was conducted identifying seven main tasks of which one routine and six alert specific tasks. Main tasks were then split up iteratively until the lowest level of actions that can be observed in simulator log files. Based on this HTA, an ACT-R cognitive model was created that was able to memorize flight information by reading airspeed and altitude data, decide when to adjust the throttle, process and respond to auditory messages, and check if its own actions match pilot’s actual behavior. This model will be referred to as the “normative” model.

An extended version of ACT-CV (Halbrügge, 2013) was used to create an interface and FlightGear log files. ACT-R did not interface with FlightGear directly (see Somers & West, 2013), but through recordings of individual participants’ performance The graphical interface of the flight simulator was represented textually, e.g., “on”, “539”, in ACT-R’s visual representation of the environment, the visicon. As the study’s focus was not on visual behavior, different parameters (e.g., airspeed, altitude, etc.) were presented at pre-defined locations independent of Airbus cockpit design. Parameter changes linked to events (e.g., engine1-on-fire from “0” to “1”) triggered sounds in ACT-R, so messages from the cockpit were presented in the same modalities as in the empirical study. Processed EEG data were displayed as event-related Boolean variable (“1” for alerts processed as target sound, “0” for standard sounds). Contents of ATC messages in the controller-pilot datalink communications could not be communicated through FlightGear. As a workaround, an extra buffer was added that gives the model access to information not displayed in the visicon.

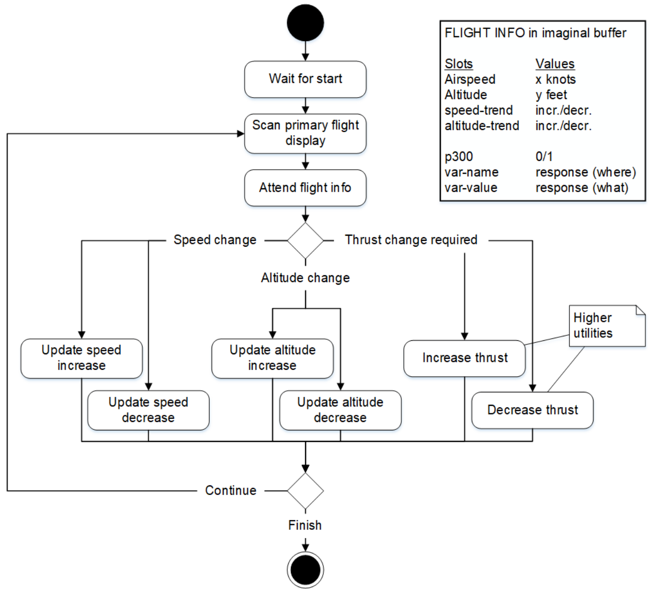


Figure 2: Routine loop in both models

For the routine task (see Figure 2), the model monitors variables of airspeed and altitude that were shown in the simulator’s primary flight display. Based on these data, it computes trends for speed and altitude and updates its internal representation of the flight information that is stored as declarative knowledge in an imaginal buffer. If airspeed approaches threshold values, the model prepares to adjust the thrust accordingly. If speed trend is not increasing or decreasing considerably, the model returns to monitoring speed or altitude after updating its flight information.

In case of auditory signals, the model leaves this routine loop and processes the sound and the corresponding message. In case of ATC messages, it processes

[1] http://home.flightgear.org/

navigational instructions and stores them in the imaginal buffer. If the model hears an alert, it retrieves a checklist matching the alert type and puts the response required from the pilot in its imaginal buffer. For all acoustic events, the normative model assumes that pilots will respond adequately and, after each event, it checks the log data for the required pilot response to evaluate if its assumption is correct. Situations where pilots do not respond adequately are treated as epistemic uncertainty and marked as cases when some sort of assistance should be provided.

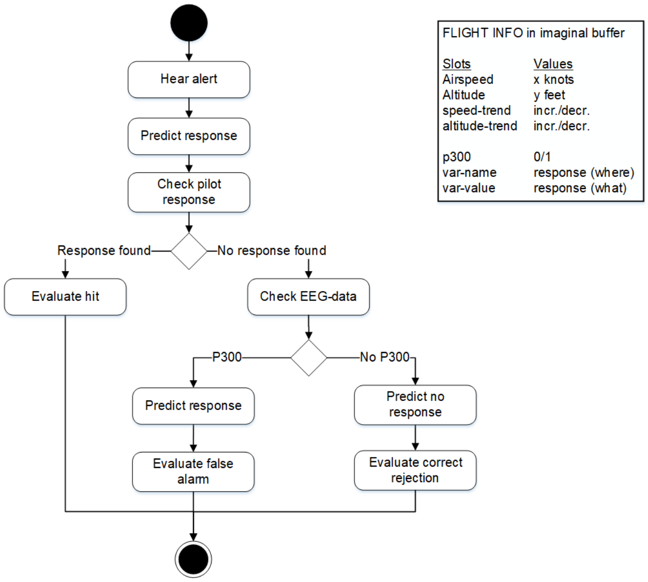


Figure 3: Alert procedure for neuroadaptive model

The neuroadaptive model forms an extension of the normative model. It follows the same courses of action for routine tasks and acoustic events that were followed by an adequate pilot response. If no adequate response is observed, the neuroadaptive model consults the EEG data to check if the pilot had paid attention to the sound (see Figure 3). If EEG data show the pilot has processed the alert or message like a standard stimulus, the neuroadaptive model updates its description of the situation to a missed alert. The model considers these cases as situations that require verbal reminders of the alert or message. Situations where no adequate response was observed but EEG-data show the preceding sound was processed are treated as epistemic uncertainty. For these situations, the model knows that assistance of some form other than a verbal reminder is needed.

Analysis

For this study the first reaction to the auditory events was evaluated, i.e. adjusting selected altitude in response to ATC messages or opening a checklist in response to alerts. In both models epistemic uncertainty was scored as incorrect description of pilot behavior. Both the normative and the neuroadaptive model could correctly describe situations with adequate pilot reactions to acoustic events; in addition, the neuroadaptive model was able to classify lacking responses as correct descriptions, when EEG data showed no reaction to the sound.

Correctly described responses are scored with 1, incorrect response descriptions with 0. For each participant, both models divide the sum of correct descriptions by the total number of alerts and ATC messages to quantify model accuracy. For both models, mean accuracy is computed across pilots. As the number of auditory events was not the same for all participants due to ATC speed messages, median and interquartile range had to be used as measures of central tendency and dispersion. Wilcoxon signed rank tests for pairwise comparisons are used to quantify added value of EEG-data for the neuroadaptive model.

Aleatory uncertainty in the neuroadaptive model is equal to one minus EEG classifier accuracy. As the data give no information about which situations are concerned by classifier inaccuracies, aleatory uncertainty is accepted and scored as correct. Added value of neuroadaptivity to the normative model is quantified by subtracting normative from neuroadaptive model accuracy. By multiplying added value with EEG classifier accuracy, a mean accuracy of the neuroadaptive model corrected for aleatory uncertainty can be computed.

Results

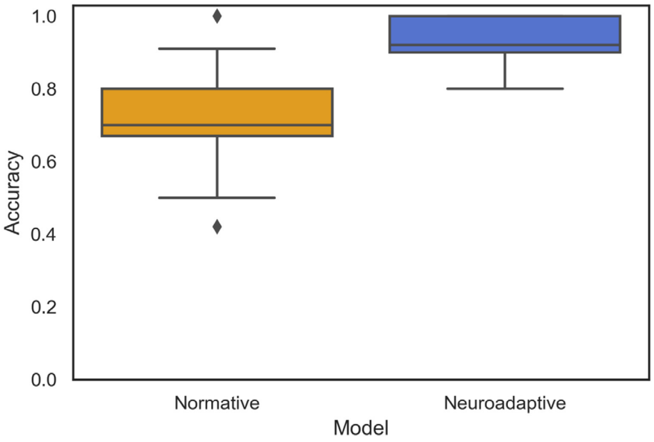


Figure 4: Median accuracy per model

In total, behavior descriptions for 225 events were generated by each model for all pilots with an average of 10.7 (*SD* = 0.9) per pilot. The normative model correctly described participant's behavior for 163 of these events ($Acc_{\text{Norm.}}$ = 0.72) with a Median model accuracy of $MDN_{\text{Norm.}}$ = 0.70 (*IQR* = 0.80 - 0.67; Figure 4). Thus, the total amount of uncertainty treated as epistemic is 0.30.

The neuroadaptive cognitive model generated correct descriptions in 213 of 225 cases ($Acc_{\text{Neuro.}}$ = 0.95) with a median accuracy of $Mdn_{\text{Neuro.}}$ = 0.92 (*IQR* = 1.0 - 0.9; Figure 4). The uncertainty treated as epistemic is therefore 0.05.

The signed rank test showed that neuroadaptive model accuracy is significantly higher compared to the normative model ($z = -4.01$, $p < 0.01$). Added value of the EEG-data is 0.23. Correcting the added value for the EEG classifier accuracy of 0.86 results in a corrected accuracy of the neuroadaptive model of 0.92 and aleatory uncertainty of 0.03.

Model accuracies per participant and model are shown in Figure 5.

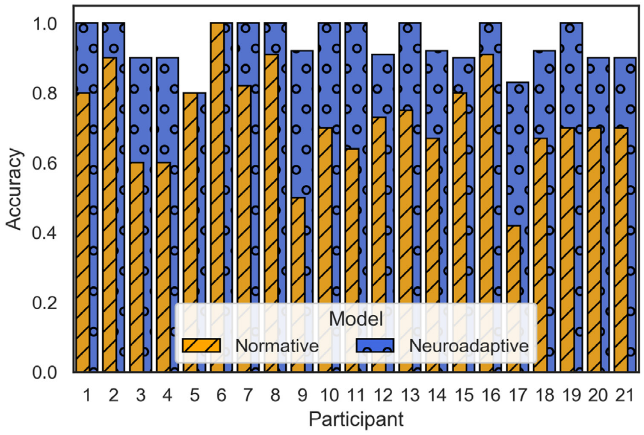


Figure 5: Mean accuracy per participant and model

Discussion

The presented concept and its application demonstrate how pilot performance can be modeled in spite of individual differences using model tracing and physiological data. The distinction between aleatory and epistemic uncertainty (Kiureghian & Ditlevsen, 2009) and their quantification was decisive for the neuroadaptive model's design and implementation. Data show how model accuracy can be significantly increased by connecting model-tracing and EEG data in line. The specification of remaining fractions of epistemic and aleatory uncertainty provide starting points for further improvement of the concept.

Whereas flight deck instrument interactions can be observed directly, unprocessed alerts can only be detected by behavioral or physiological symptoms. Due to aleatory uncertainty introduced by the EEG classifier, model tracing with instrument and EEG data had to be connected in line to maximize effectiveness in reducing epistemic uncertainty. Compared to other studies using EEG data to model effects of individual differences, the integration of EEG data was quite straightforward for the neuroadaptive model and did not require a dual model approach (Putze et al., 2015). Model tracing based on the log files proved effective in detecting deviations from normative behavior due to an increased density of acoustic events in the scenario. Real flight however contains long periods of monitoring instruments without direct input required. Deriving mental states based on model tracing (Fu et al., 2006) in such highly automated or autonomous environments could therefore require other pilot behavior data sources, e.g. unobtrusive monitoring of neurophysiological activity, speech or gaze. Cognitive models are well suited for the interpretation of such data by linking physiological phenomena to context.

Apart from measurement and classification errors, the neuroadaptive model was able to explain ~81% of the normative model's epistemic uncertainty, leaving a total of 5% of cases when the model does not know what made participants fail to react adequately. These data suggest that cognitive assistance in form of verbal reminders would suffice to help with performance recovery in all other situations lacking responses from participants.

Normative model accuracy represents the effects of individual differences on performance given the scenario. By design, the neuroadaptive model improves on the normative model; the significance of improvement with the EEG data is moderated by the effect of individual differences. Nonetheless, increased accuracy of the neuroadaptive model shows how epistemic uncertainty can be reduced with the help of physiological data. For an empirical evaluation of the concept, a comparison with alternative designs for model-based assistance is required. E.g., a wizard-of-oz setup with a human co-pilot interpreting pilot behavior could be compared to the effectiveness of the neuroadaptive model.

The neuroadaptive model tracks pilots' perception of auditory events. The fact that a piece of information has been perceived and processed by a pilot does not mean that it has been understood. Measures of pilots' situation assessment and awareness (Endsley, 1995) may help to reduce epistemic uncertainty about why a pilot may fail to respond adequately. Physiological symptoms of cognitive conflict can be used to identify when information that was perceived could not be comprehended by the pilot.

Mean accuracy of the neuroadaptive model corrected for aleatory uncertainty is 92 %. Aleatory uncertainty may be reduced with other independent physiological measures, e.g. eye tracking. EEG classification could be supported with corresponding gaze data by connecting both methods in line. E.g., when the EEG data show that a pilot has processed an alert, saccades to the warning display after the alert can reduce the uncertainty by eye tracking classification accuracy.

Further research is required on how to model individual differences with the help of behavioral and physiological measures of operators' cognitive states. Model-based assistance in human machine interaction can provide machines with an implicit feedback loop that allows to check if the information they provide is perceived and understood by the user. Ideally, this will enable machines to form a more refined model of their users and to anticipate their behavior in much the same way that humans learn to interact with a machine.

Acknowledgments

The authors would like to thank Matthias Buderath, Christoph Vernaleken, Daniel Dreyer, Laurens Krol, Thorsten Zander, Lena Andreessen, Dan Bothell, BrainProducts GmbH and Inge Wetzel for their support.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

Byrne, M. D., Kirlik, A., Fleetwood, M. D., Huss, D. G., Kosorukoff, A., Lin, R.-S., & Fick, C. S. (2004). A closed-loop, ACT-R approach to modeling approach and landing with and without synthetic vision system (SVS) Technology. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *48*(17), 2111–2115. https://doi.org/10.1177/154193120404801707

Dehais, F., Roy, R. N., & Scannella, S. (2019). Inattentional deafness to auditory alarms: Inter-individual differences, electrophysiological signature and single trial classification. *Behavioural Brain Research*, *360*, 51–59. https://doi.org/10.1016/j.bbr.2018.11.045

Dutton, J. M., & Starbuck, W. H. (1971). *Computer simulation of human behavior*. New York: Wiley.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human factors*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

Estes, S., Burns, K., Helleberg, J., Long, K., Stein, J., & Pollack, M. (2016). Digital copilot: Cognitive assistance for pilots. In *Proceedings of the AAAI Fall Symposium on Cognitive Assistance in Government and Public Sector Applications.*

Fisher, C. R., Walsh, M. M., Blaha, L. M., Gunzelmann, G., & Veksler, B. (2016). Efficient parameter estimation of cognitive models for real-time performance monitoring and adaptive interfaces. In D. Reitter & F. E. Ritter (Eds.), *Proceedings of the 14th International Conference on Cognitive Modeling* (pp. 113–118).

Fu, W.-T., Bothell, D., Douglass, S., Haimson, C., Sohn, M.-H., & Anderson, J. (2006). Toward a real-time model-based training system. *Interacting with Computers*, *18*(6), 1215–1241. https://doi.org/10.1016/j.intcom.2006.07.011

Halbrügge, M. (2013). ACT-CV - Bridging the gap between cognitive models and the outer world. In E. Brandenburg (Ed.), *Grundlagen und Anwendungen der Mensch-Maschine-Interaktion: 10. Berliner Werkstatt Mensch-Maschine-Systeme, 10.-12. Oktober 2013, Berlin = Foundations and applications of human machine interaction* (pp. 205–210). Berlin: TU Berlin.

Kiureghian, A. D., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112. https://doi.org/10.1016/j.strusafe.2008.06.020

Krol, L. R., Klaproth, O. W., Vernaleken, C., Wetzel, I., Gaertner, J., Russwinkel, N., & Zander, T. O. (2018, July). *Towards a neuroadaptive cockpit: first results.* 3rd International Mobile Brain/Body Imaging Conference, Berlin, Germany.

Putze, F., Schultz, T., & Propper, R. (2015). Dummy model based workload modeling. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 935–940). IEEE. https://doi.org/10.1109/SMC.2015.171

Rehling, J., Lovett, M., Lebiere, C., Reder, L., & Demiral, B. (2004). Modeling complex tasks: An individual difference approach. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 26).

Schoppek, W., & Boehm-Davis, D. A. (2004). Opportunities and challenges of modeling user behavior in complex real world tasks. *MMI interaktiv*, *7*, 47–60.

Somers, S., & West, R. (2013). Steering control in a flight simulator using ACT-R. In *Proceedings of the International Conference on Cognitive Modeling.*

Stanton, N. A. (2006). Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics*, *37*(1), 55–79. https://doi.org/10.1016/j.apergo.2005.06.003

Taatgen, N. A. (1999). Cognitief Modelleren: Een nieuwe kijk op individuele verschillen. *Nederlands Tijdschrift voor de Psychologie*, *54*(4), 167–176.

Taatgen, N. A., Huss, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General*, *137*(3), 548–565. https://doi.org/10.1037/0096-3445.137.3.548

Zander, T. O., & Kothe, C. (2011). Towards passive brain-computer interfaces: Applying brain-computer interface technology to human-machine systems in general. *Journal of Neural Engineering*, *8*(2), 25005. https://doi.org/10.1088/1741-2560/8/2/025005

Zander, T. O., Krol, L. R., Birbaumer, N. P., & Gramann, K. (2016). Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity. *Proceedings of the National Academy of Sciences*, *113*(52), 14898–14903. https://doi.org/10.1073/pnas.1605155114

Zhang, Z., Russwinkel, N., & Prezenski, S. (2018). Modeling Individual Strategies in Dynamic Decision-making with ACT-R: A Task toward decision-making Assistance in HCI. *Procedia Computer Science*, *145*, 668–674. https://doi.org/10.1016/j.procs.2018.11.064

Automated cognitive modeling with Bayesian active model selection

Vishal H. Lall (lall@berkeley.edu)
University of California Berkeley

Jordan W. Suchow (jws@stevens.edu)
Stevens Institute of Technology

Gustavo Malkomes (luizgustavo@wustl.edu)
Washington University in St. Louis

Thomas L. Griffiths (tom@princeton.edu)
Princeton University

Keywords: Bayesian modeling, active learning, active model selection, numerosity, cognitive modeling

Abstract

Behavioral experiments are often feed-forward: they begin with designing the experiment, and proceed by collecting the data, analyzing it, and drawing inferences from the results. Active learning is an alternative approach where partial experimental data is used to iteratively design subsequent data collection. Here, we study experimental application of Bayesian Active Model Selection (BAMS), which designs trials to discriminate between a set of candidate models. We consider a model set defined by a generative grammar of Gaussian Process kernels that can model both simple functions and complex compositions of them. To validate the method experimentally, we use BAMS to discover how factors such as contrast and number affect numerosity judgements. We compare the rate of convergence of the active-learning method to a baseline passive-learning strategy that selects trials at random. Active learning over a structured model space may increase the efficiency and robustness of behavioral data acquisition and modeling.

Predictions of a Model of Language Comprehension Compared to Brain Data

Peter Lindes (plindes@umich.edu)
University of Michigan, Computer Science and Engineering
2260 Hayward Street, Ann Arbor, MI 48109 USA

Keywords: Language comprehension; construction grammar; sentence processing; brain measurements; EEG.

A Model of Human Language Comprehension

Humans understand natural language rapidly in real time. Empirical literature supports the idea that human language comprehension involves *immediate interpretation*. Tanenhaus et al. (1995) show that humans focus their gaze on a particular object in the scene immediately upon hearing a description of that object. Bergen (2012) reviews studies that demonstrate that perceptual and motor areas of the brain are activated dynamically during sentence comprehension. These and other studies show that as soon as a word or phrase that refers to an object or event is processed, its meaning is immediately interpreted and grounded to the situational or dialog context.

Language processing is constrained by the capacity of working memory. Christiansen and Chater (2016) argue that a partial comprehension must be quickly incorporated into larger structures, or it will be lost due to working memory limits. They propose *chunk-and-pass processing*, where the analysis of a sentence is constructed in units they call *chunks*. Whenever possible, chunks already built are composed into larger ones, so that only a few chunks at a time need to be separately maintained in working memory.

Lindes and Laird (2016) have developed a computational theory of language comprehension with immediate interpretation using a chunk-and-pass-like approach. The theory has been implemented in a system called Lucia (Lindes, Mininger, Kirk, & Laird, 2017) that models form-meaning mapping using Embodied Construction Grammar (ECG; B. K. Bergen & Chang, 2013) and is built in the Soar cognitive architecture (Laird, 2012).

An analysis of this model shows that its processing depends on four basic principles:

1. A sentence is comprehended one form-meaning unit, called a *construction*, at a time.
2. Comprehension proceeds as a succession of building these units, or a series of *construction cycles*.
3. Each construction cycle is made up of three phases: *selection*, *integration*, and *grounding*.
4. Each of these phases accesses different types of memory.

In what follows we analyze implications and predictions of this model and compare them to EEG data studies.

Model Predictions

Figure 1 shows a spatiotemporal map of the processing of a simple sentence. Cognitive cycles are grouped into construction cycles, and then word cycles.

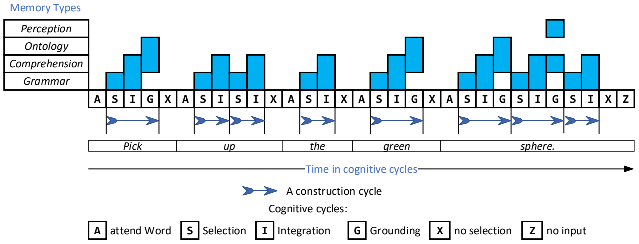


Figure 1: Time course of memory use

In each construction cycle, one construction from the available inventory in the grammar memory is selected (S) and integrated (I) into the comprehension state in working memory. Optionally, it is grounded (G) to the agent's knowledge. A, X, and Z operators perform overhead functions, such as attending to the next word.

The model uses four types of memory. Long-term memories store linguistic knowledge (Grammar) and the agent's long-term knowledge of the environment and its perception and action capabilities (Ontology). Short-term memories store the state of the comprehension process (Comprehension) and the agent's current perception, dialog, and situational states (Perception).

This model implies a time sequence in which different memories are accessed at different times, as Figure 1 shows. We suggest that this spatiotemporal pattern of memory accesses may approximate a similar pattern of activation in the brain. In accordance with standard modeling, the cognitive cycles have a 50ms time course in humans, modulated by long-term memory access. We expect that comparing these predictions to brain data will help understand both the brain and the model better.

Comparison to Brain Data

Figure 2 shows examples of the kind of data reported in the large literature on measurements of the brain during language comprehension (Left: Schwartz & Mitchell, 2019; Right: Hale, Dyer, Kuncoro, & Brennan, 2018). The images show several kinds of Event Related Potentials (ERPs) averaged over many words as they are distributed in time and space. We will compare our model to these and other related data.

Our model predicts that more time is required to process content words than function words due to grounding and the frequent need for multiple constructions. Brennan and Hale (2019) compare several simpler models to EEG data and show processing differences between these two types of words, and that less frequent content words have a stronger N400 response. Further analysis of the details is needed.

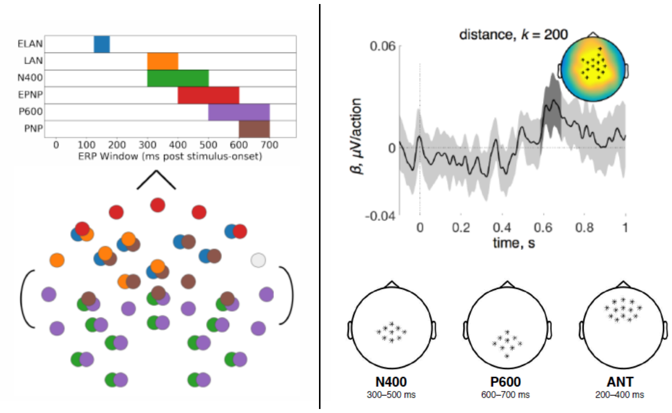


Figure 2: Examples of EEG data

Our model shows that the number of construction cycles per word varies, and the length of each cycle varies depending on whether grounding is needed. The model implies that processing of each word runs to completion before the next word is attended to. The EEG data shows word processing extending to beyond 600ms, even to almost a second (see Fig, 2). Human language input proceeds typically in a range of 150-250 words/minute, for an average time between words of 240-400ms. This implies that word processing often continues in parallel with the processing of subsequent words. This sort of parallelism is lacking in our model. Figure 3 gives a suggestion of what the processing pattern might look like. How to accomplish this within the Soar architecture is an open question.

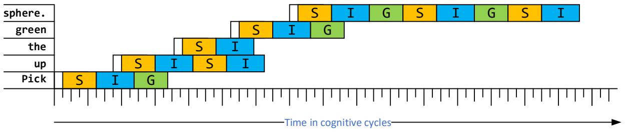


Figure 3: A possible parallel processing pattern

Bornkessel-Schlesewsky and Schlesewsky (2019) present a new, unified way of looking at the N400 response, arguing that it happens when the brain needs to modify its predictive model of the sentence. Our system restructures its sentence model each time it adds a construction, and when it performs a local repair. A general prediction function, which is lacking from our current model, will need to be added for it to better reflect the brain's processing.

Our model suggests there are different memory areas involved in language comprehension, and that there is a repetitive time sequence in their accesses. The data in Figure 2 show temporal patterns in the spatial distribution of brain activation. There is the potential here to improve both our understanding of the brain and our model by further analysis of the relationship between these two things.

This abstract suggests ways in which cognitive models of comprehension and brain measurements can be compared to improve both the models and understanding of the brain. The analysis here is very preliminary and superficial; much work is needed to explore these areas in detail.

Acknowledgments

The work described here was supported by the AFOSR under Grant Number FA9550-18-1-0168. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressly or implied, of the AFOSR or the U.S. Government.

References

Bergen, B. K. (2012). *Louder Than Words: The New Science of How the Mind Makes Meaning*. New York: Basic Books.

Bergen, B. K., & Chang, N. C. (2013). Embodied Construction Grammar. In T. Hoffman & G. Trousdale (Eds.), *The Oxford Handbook of Construction Grammar* (pp. 168–190). Oxford University Press.

Bornkessel-Schlesewsky, I., & Schlesewsky, M. (2019). Toward a Neurobiologically Plausible Model of Language-Related, Negative Event-Related Potentials. *Frontiers in Psychology*, *10*(FEB). https://doi.org/10.3389/fpsyg.2019.00298

Brennan, J. R., & Hale, J. T. (2019). Hierarchical structure guides rapid linguistic predictions during naturalistic listening. *PLoS ONE*, *14*(1). https://doi.org/10.1371/journal.pone.0207741

Christiansen, M. H., & Chater, N. (2016). The Now-or-Never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, 1–72. https://doi.org/10.1017/S0140525X1500031X

Hale, J., Dyer, C., Kuncoro, A., & Brennan, J. R. (2018). *Finding Syntax in Human Encephalography with Beam Search*. (2014). Retrieved from http://arxiv.org/abs/1806.04127

Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: The MIT Press.

Lindes, P., & Laird, J. E. (2016). Toward Integrating Cognitive Linguistics and Cognitive Language Processing. *International Conference on Cognitive Modeling (ICCM)*.

Lindes, P., Mininger, A., Kirk, J. R., & Laird, J. E. (2017). Grounding Language for Interactive Task Learning. *Proceedings of the First Workshop on Language Grounding for Robotics*, 1–9. https://doi.org/10.18653/v1/w17-2801

Schwartz, D., & Mitchell, T. (2019). *Understanding language-elicited EEG data by predicting it from a fine-tuned language model*. Retrieved from http://arxiv.org/abs/1904.01548

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K., & Sedivy, J. C. (1995). Integration of Visual and Linguistic Information in Spoken Language Comprehension. *Science*, *268*(5217), 1632.

Conceptually Plausible Bayesian Inference in Interval Timing

Sarah C. Maaß (s.c.maass@rug.nl)
Department of Experimental Psychology, University of Groningen
Groningen, The Netherlands

Leendert van Maanen (l.vanmaanen@uva.nl)
Department of Psychological Methods, University of Amsterdam
Amsterdam, The Netherlands

Hedderik van Rijn (d.h.van.rijn@rug.nl)
Department of Experimental Psychology, University of Groningen
Groningen, The Netherlands

Keywords: Bayesian Inference; Central Tendency; Conceptual Model Comparison; Interval Timing

Introduction

In a world that is uncertain and noisy, human perception makes use of optimization procedures to reduce the influence of moment-to-moment noise by incorporating statistical properties of previous experiences. This observation holds for the perception of many psychophysical quantities, ranging from light intensity to interval timing, the focus of the current study. These types of optimization procedures assume that when a specific interval needs to be reproduced, observers do not only take the current percept into account but also use their prior knowledge of previous similar incidents to form an internal estimate of the just perceived interval, yielding a central tendency effect (Hollingworth, 1910). That is, in a reproduction task in which different durations have to be reproduced, the shorter intervals will be overestimated and the longer durations underestimated yielding a regression towards the mean. A formal account of this phenomenon has only recently been proposed. In 2010, Jazayeri and Shadlen formulated a mathematical framework in which a Bayesian observer is assumed to combine the noisiness associated with time perception with a probability distribution representing the earlier observed durations. The actual reproduction is based on the posterior distribution, which consist of the integration of a Gaussian-distributed likelihood, representing the observed duration, with a uniform prior, representing the experimental history. Jazayeri and Shadlen demonstrated that the mean of the posterior distribution captures a number of important empirical phenomena, including the central tendency effect.

To account for individual differences in the magnitude of the central tendency effect, they assumed differences in the variability of the temporal percept, represented in the width of the likelihood (see https://vanrijn.shinyapps.io/MaassVanMaanenVanRijn2019/ for a simulation). Note that after a value has been sampled from the posterior distribution, Gaussian-shaped production noise is applied to map the posterior-based estimation to the actually reproduced duration. Similar Bayesian observer models have been shown to accurately reproduce human behavior in a number of timing tasks (see, e.g., Shi, Church, & Meck, 2013).

From a theoretical or conceptual perspective, however, one can question certain implementation decisions underlying this Bayesian Observer Model. Firstly, the prior with which the likelihood is convolved is assumed to be a uniform distribution precisely spanning the range of the presented durations. Even though this provides computational simplicity, its theoretical suitability can be questioned as the average of the resulting posterior distributions will, because of the central tendency, have a higher mass around the center of the distribution. Following the assumption that the prior is based on previous posteriors, the prior should reflect this bias towards the mean. This example of the central limit theorem would suggest a more Gaussian-like distributed prior which also naturally results from instance-based explanations of the role of memory processes in interval timing (for a review, see Van Rijn, 2016). Cicchini et al. (2012) addressed the issue of the uniform prior, and proposed to use a truncated normal distribution to represent the prior. Where Jazayeri and Shadlen (2010) focused on the width of the likelihoods, resembling clock precision, to account for the variability between participants in observed central tendency effects, Cicchini et al. (2012) argued that the prior might also differ on a per participant basis. To tear apart the contribution of the likelihood and prior, they estimated clock variability using a secondary task. With the likelihood fixed on a per participant basis, they demonstrated that the width of a truncated normal distribution varied over participants.

Even though a Gaussian-like distribution is theoretically more plausible than a uniform prior, its theoretical elegance is affected by the necessity to constrain its range to prevent it extending to negative values, nor does it match the heavier right tail observed in empirical data. In addition, the proposed symmetrical, Gaussian prior does not match the stronger central tendency bias for the longer compared to the shorter durations: As the mass of the average posteriors associated with the longer durations is more pulled towards the mean of all presented durations than the average of the posteriors associated with the shorter durations, a skewed Gaussian distribution would be theoretically more plausible. A second theoretical challenge for these Bayesian observer models is that they incorporate two independent sources of noise, one associated with the perceptual phase (w_m), determining the width of the likelihood, and one associated with the reproduction of a duration which is based on the posterior (w_p). Whereas w_m captures the perceptual noise associated with perceiving the onset and offset of the presented duration, as well as the clock noise associated with the actual timing of the interval, w_p captures the perceptual noise for the onset of the reproduction phase, the clock noise, and the motor noise associated with the motor movement to mark the end of the reproduction phase (by a key press). Assuming perceptual noise to be smaller than motor noise, and clock noise to be the dominant source of noise (e.g., Taatgen, Van Rijn, & Anderson, 2007), w_p should always be larger than w_m. Additionally, as clock noise can be assumed to be the largest source of variability in both w_m and w_p, it follows to estimate w_m and estimate Δw_p that expresses the difference in noise between a perception and motor action (i.e., the reproduction noise, w_p, is defined as $w_m + \Delta w_p$). As both parameters were fit independently in Jazayeri and Shadlen's Bayesian Observer model, w_p could be estimated at a smaller value than w_m and no correlation between both parameters was instantiated. In contrast, no parameters were estimated in Cicchini's et al. (2012) model. Their model incorporated an estimate for w_m based on each participant's performance on a secondary task, and w_p was fixed for all participants at a value that fell within the range of values that were determined for w_m. Thus, this model did not adhere to the notion that w_m should be larger than w_p, and it assumed that all sources of noise, including clock noise, were identical for all participants during reproduction.

Here we present Bayesian Observer models with different assumptions with respect to the source of the individual differences, by considering individual differences in clock noise and memory: We will independently estimate w_m and Δw_p assuming priors based on either a fixed uniform prior distribution, or normal and log-normal shaped prior distributions of which the variance will be estimated. To assess the goodness of fit of these models, we will estimate fit measures for 15 aged participants with the diagnosis of amnestic Mild-Cognitive Impairment (aMCI) and 44 healthy aged controls. Whereas the first group showed strong central tendency effects, the latter group showed weaker effects (Maaß, Riemer, Wolbers, & Van Rijn, submitted). will be compared. Interestingly, measures of memory functioning predicted the magnitude of the central tendency effect, even in the healthy aged control group. Additionally, we will use 1-second task data (Maaß & Van Rijn, 2018) to assess clock variability. The results suggested that neither age (cf. Paraskevoudi, Balcı, & Vatakis, 2018), nor clinical status (cf. Rueda & Schmitter-Edgecomb, 2009) influenced clock time variability, but that aMCI patients more strongly weigh prior experiences than healthy, age-matched controls, resulting in stronger central tendency effects. By fitting Bayesian Observer models to the empirical data from these (sub)populations, we aim to understand the contributions of likelihood and prior on temporal reproduction in healthy and memory-impaired individuals.

In sum, we will (1) assess whether one type of prior is preferred, (2) whether estimated values that mostly reflect clock noise (i.e., w_m) correlates to the collected clock-variability measures, and (3) whether the estimated prior parameters provide a sensible theoretical interpretation of the empirical phenomena.

Acknowledgements

This work was supported by VICI grant 453-16-005, financed by the Netherlands Organisation for Scientific Research (NWO).

References

Cicchini, G. M., Arrighi, R., Cecchetti, L., Giusti, M., & Burr, D. C. (2012). Optimal encoding of interval timing in expert percussionists. *Journal of Neuroscience*, *32*(3), 1056-1060.

Gibbon, J., Church, R. M., & Meck, W. H. (1984). Scalar timing in memory. *Annals of the New York Academy of sciences*, *423*(1), 52-77.

Hollingworth, H. L. (1910). The central tendency of judgment. *The Journal of Philosophy, Psychology and Scientific Methods*, *7*(17), 461-469.

Jazayeri, M., & Shadlen, M. N. (2010). Temporal context calibrates interval timing. *Nature Neuroscience*, *13*(8), 1020-1026.

Maaß, S. C., & Van Rijn, H. (2018). 1-Second Productions: A Validation of an Efficient Measure of Clock Variability. *Frontiers of Human Neuroscience. 12,* 519.

Maaß, S. C., Riemer, M., Wolbers, T., & Van Rijn, H. (submitted). Timing Deficiencies in Amnestic Mild Cognitive Impairment: Disentangling Clock and Memory Processes.

Paraskevoudi, N., Balcı, F., & Vatakis, A. (2018). "Walking" through the sensory, cognitive, and temporal degradations of healthy aging. *Annals of the New York Academy of Sciences*. 1426(1), 72-92.

van Rijn, H. (2016). Accounting for memory mechanisms in interval timing: a review. *Current Opinion in Behavioral Sciences*, *8*, 245-249.

Rueda, A. D., & Schmitter-Edgecombe, M. (2009). Time estimation abilities in mild cognitive impairment and Alzheimer's disease. *Neuropsychology*, *23*(2), 178-188.

Shi, Z., Church, R. M., & Meck, W. H. (2013). Bayesian optimization of time perception. *Trends in cognitive sciences*, *17*(11), 556-564.

Taatgen, N. A., Van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review*, *114*(3), 577-598.

A Computational Theory for the Model Construction, Inspection and Variation Phase in Human Spatial Reasoning

Julia Mertesdorf[1] and **Emmanuelle-Anna Dietz Saldanha**[2] and **Steffen Hölldobler**[2,3] and **Marco Ragni**[1]
[1]Cognitive Computation Lab, Technische Fakultät, Universität Freiburg, 79110 Freiburg, Germany
[2] International Center for Computational Logic, TU Dresden, 01187 Dresden, Germany
[3] North-Caucasus Federal University, Stavropol, Russian Federation

Abstract

Our long-term research goal is the development of a cognitive theory for adequately modeling human reasoning tasks. The theory should be *computational* and on the other hand *comprehensive*. The Weak Completion Semantics (WCS) seems to be a good candidate, as it has previously shown to adequately model a wide range of human reasoning tasks. By means of human spatial reasoning, we show here that the WCS can fully cover all three stages of reasoning that have been suggested by the preferred mental model theory. The contribution comprises aspects within the area of Computer Science and Psychology. Through the formal process of modeling, in particular through the computation of alternative models within the variation phase, we have gained new insights and put forward assumptions that need to be verified.

Keywords: Computational Theory, Spatial Reasoning, Preferred Mental Model Theory, Weak Completion Semantics

Introduction

Our long-term research goal is the development of a cognitive theory for adequately modeling human reasoning tasks. The theory should be *computational* in that answers to queries can be computed. The theory should be *comprehensive* in that different human reasoning tasks can be modeled by the theory without changing the theory.

Currently the *Weak Completion Semantics (WCS)* is a very good, if not the best candidate for such an comprehensive and computational cognitive theory. The WCS is based on ideas initially proposed by Stenning and van Lambalgen (2005, 2008), but is mathematically sound: As Hölldobler and Kencana Ramli (2009) have shown, the three-valued logic used in Stenning and van Lambalgen (2008) is inadequate for the suppression task. Surprisingly, the suppression task can be adequately modeled if the three-valued Łukasiewicz (1920) logic is used. Since then, the WCS has been applied to various human reasoning tasks (cf. Wason, 1968; Byrne, 1989) summarized in Hölldobler (2015), has outperformed twelve cognitive theories considered by Khemlani and Johnson-Laird (2012) in syllogistic reasoning (Oliviera da Costa, Dietz Saldanha, Hölldobler, & Ragni, 2017), and can be implemented as a neural network (Dietz Saldanha, Hölldobler, Kencana Ramli, & Palacios Medinacelli, 2018).

Given a human reasoning task, the first step within the WCS is to construct a logic program representing the task. The construction of these programs is based on several principles, some of which are well-established like using *licenses for inferences*, *existential import* (Johnson-Laird, 1983; Rips, 1994; Stenning & van Lambalgen, 2008), or *Gricean implicature* (Grice, 1975), whereas others are novel like *unknown generalization* (Oliviera da Costa et al., 2017). If interpreted under the three-valued logic of Łukasiewicz (1920), the programs have a unique supported model, which can be computed by iterating the semantic operator introduced by Stenning and van Lambalgen (2008). Reasoning is performed and answers are computed with respect to these models. Skeptical abduction is applied if some observations in the given human reasoning task can not be explained otherwise.

Human Spatial Reasoning

In this paper we apply the WCS to spatial reasoning. Suppose you were given (in this sequence) the following information:

The Audi is left of the Beetle.
The Audi is left of the Cadillac.
The Cadillac is left of the Dodge.

Given these premises, what, if anything, follows for the Beetle and the Dodge? A psychological finding by Ragni and Knauff (2013) is that many human reasoners do construct the preferred (mental) model $a \quad b \quad c \quad d$.[1]

Based on the spatial representation of this preferred model, a reasoner could infer that *the Beetle is to the left of the Dodge*. In fact, most human reasoners seem to do this. Yet, the preferred model is not the only model for the given premises. If a reasoner would construct these alternative models he/she may find a counter-example and may answer that *nothing follows.* Under First-order Logic, there might be more than one model for the given premises of the task, letting unspecified which one to choose as the preferred one. Ragni and Knauff (2013) presented an algorithmic approach – the preferred mental model theory – to construct and manipulate mental models. Based on this theory, Dietz, Hölldobler, and Höps (2015) modeled the preferred models for human spatial reasoning in the WCS. They represented the relations among objects, transitivity properties, and the first-free-fit-principle suggested by Ragni and Knauff (2013) as logic programs and showed that the supported model in the WCS corresponds to the preferred mental model.

However, the approach of Dietz et al. (2015) is restricted to computing and reasoning with respect to the preferred mental model and does not cover the inspection and variation phase

[1] a denotes Audi, b Beetle, c Cadillac, and d Dodge.

reported by Ragni and Knauff (2013). More generally, it does not cover the flesh-out process after the initial mental model has been constructed. The goal of this paper is to show that the whole process – construction of an initial mental model, inspection, and variation – can be modeled by the WCS in the context of spatial reasoning.

Programs

Here, we consider programs similar to the ones introduced by Dietz Saldanha, Hölldobler, and Pereira (2017). A *(contextual logic) program* is a finite set of (positive) *facts* of the form $A \leftarrow \top$, (negative) *assumptions* of the form $A \leftarrow \bot$ and *rules* of the form $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge (\neg)\mathsf{ctxt}\, L_{m+1} \wedge \cdots \wedge (\neg)\mathsf{ctxt}\, L_{m+p}$, where A is an atom, L_i, $1 \leq i \leq m+p$, are literals (i.e. L_i is an atom or a negated atom), $\top$ denotes truth, $\bot$ denotes falsehood, and ctxt is a unary *context* operator. The interpretation of the connectives is given in Table 1.

The ctxt operator is similar to *negation as failure* (Clark, 1978) or *default negation* locally, and helps to provide a natural formalization of defeasible rules. To explain its behavior let us return to the spatial reasoning problem in the introduction. After reading the first premise, most participants seem to assume that the space right of the Beetle is not occupied. However, (classical) logically, it can neither be proven that the space is occupied, nor that it is not. Here, the application of ctxt allows us to conclude that the space is not occupied.

The example from the introduction can be represented by the facts $left(a,b) \leftarrow \top$, $left(a,c) \leftarrow \top$, $left(c,d) \leftarrow \top$. In addition, the rule $right(X,Y) \leftarrow left(Y,X)$ denotes the symmetry of left and right. Such a rule is considered to be a schema. Ground instances of this rule are obtained by replacing the variables occurring in it by the constants occurring in the program. In this example, these are a, b, c, and d. Let $\mathcal{P}$ be a program. $\mathrm{g}\,\mathcal{P}$ denotes the set of ground instances of clauses occurring in $\mathcal{P}$.

Computation of Supported Models

The connectives in Table 1 can be read as *not* ($\neg$), *and* ($\wedge$), *or* ($\vee$), *if* ($\leftarrow$), *only if* ($\leftrightarrow$) and *not, if not true* (ctxt). It remains to specify the meaning of ground atoms. A ground atom A may be true ($\top$), false ($\bot$), or unknown (U). An interpretation I can be represented by a pair $\langle I^\top, I^\bot \rangle$, where $I^\top = \{A \mid I(A) = \top\}$ and $I^\bot = \{A \mid I(A) = \bot\}$. As interpretations are mappings, $I^\top$ and $I^\bot$ must be disjoint. Ground atoms which do not occur in $I^\top \cup I^\bot$ are mapped to U. I is a *model* for a program $\mathcal{P}$ if and only if I maps all ground instances of clauses occurring in $\mathcal{P}$ to true.

Under the WCS a program $\mathcal{P}$ may admit a unique supported model which can be computed by iterating the semantic operator $\Phi_\mathcal{P}$ on the space of interpretations provided by Stenning and van Lambalgen (2008). Let I be an interpretation, then $\Phi_\mathcal{P}(I) = \langle J^\top, J^\bot \rangle$, where

$$\begin{aligned} J^\top &= \{A \mid \text{there is } A \leftarrow body \in \mathrm{g}\,\mathcal{P} \text{ such that } I(body) = \top\}, \\ J^\bot &= \{A \mid \text{there is } A \leftarrow body \in \mathrm{g}\,\mathcal{P} \text{ and} \\ &\qquad \text{for all } A \leftarrow body \in \mathrm{g}\,\mathcal{P}, \text{ we find } I(body) = \bot\}. \end{aligned}$$

Under certain conditions $\Phi_\mathcal{P}$ has a unique fixed point which can be computed by iterating the operator starting with an arbitrary interpretation.[2] In this case, this fixed point is the supported model of the weak completion of the given program $\mathcal{P}$. For example, considering the program presented in the previous section and starting with the empty interpretation $\langle \emptyset, \emptyset \rangle$ the fixed point $\langle I^\top, \emptyset \rangle$ is reached after two iterations, where

$$\begin{aligned} I^\top &= \{left(a,b),\ left(a,c),\ left(c,d)\} \\ &\cup \{right(b,a),\ right(c,a),\ right(d,c)\}. \end{aligned}$$

All instances of *left* are added in the first iteration, whereas all instances of *right* are added in the second iteration.

A formula F *follows from* $\mathcal{P}$ *under the WCS* ($\mathcal{P} \models_{wcs} F$) if and only if the supported model of $\mathcal{P}$ maps F to true.

Construction/ Inspection for Preferred Models

Relations between objects can be easily represented in programs. However, there is no straightforward way in which we can express the order in which the premises are given. But exactly this information is crucial if we want to formalize the preferred mental model theory. For this purpose, we explicitly express phases, where each premise is read at one particular phase.

Let $\mathcal{S}$ be a spatial reasoning problem consisting of a finite sequence of premises and a conclusion. The program $\mathcal{P}_\mathcal{S}$ represents the premises of $\mathcal{S}$ and the necessary background knowledge in order to construct the preferred mental model. Within $\mathcal{P}_\mathcal{S}$ we will use the following relations, whose informal meanings are as follows:

$l(X,Y,I)$	in phase I, X is placed to the left of Y,
$nl(X,Y,I)$	in phase I, X is the left neighbor of Y,
$ol(X,I)$	in phase I, directly left of X is occupied,
$or(X,I)$	in phase I, directly right of X is occupied,

where $I \in [1,n]$, n is the number of premises, and X and Y are objects. The construction of the program $\mathcal{P}_\mathcal{S}$ is initialized by specifying all premises of $\mathcal{S}$ as facts of the form

$$l(u,v,i) \ \leftarrow \ \top, \tag{1}$$

given that the i-th premise of $\mathcal{S}$ was *object u is left of object v*. Thereafter, the following rules are added:[3]

$$\begin{aligned} & nl(X,Y,I) \leftarrow \mathsf{ctxt}\, l(X,Y,I) & (2)\\ & \qquad\qquad \wedge\ \overline{\mathsf{ctxt}}\, ol(Y,I) \wedge \overline{\mathsf{ctxt}}\, or(X,I). & \\ & nl(X,Y,J+1) \leftarrow nl(X,Y,J). & (3)\\ & ol(Y,J+1) \leftarrow nl(X,Y,J). & (4)\\ & or(X,J+1) \leftarrow nl(X,Y,J). & \\ & l(X,Z,J+1) \leftarrow l(X,Y,J+1) \wedge nl(Z,Y,J). & (5)\\ & l(Z,Y,J+1) \leftarrow l(X,Y,J+1) \wedge nl(X,Z,J). & \\ & left(X,Y) \leftarrow nl(X,Y,n). & (6)\\ & left(X,Z) \leftarrow left(X,Y) \wedge left(Y,Z). & (7)\\ & right(X,Y) \leftarrow left(Y,X). & (8) \end{aligned}$$

[2] See, Dietz Saldanha et al. (2017) for details. For each program $\mathcal{P}$ presented in this paper $\Phi_\mathcal{P}$ has a unique fixed point.

[3] Here and in the sequel, $\overline{\mathsf{ctxt}}$ is used as abbreviation for $\neg\mathsf{ctxt}$.

Table 1: Three-valued Łukasiewicz logic with ctxt. F is a formula, L a literal, and $\top$, $\bot$, and U denote *true*, *false*, and *unknown*, respectively.

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

L	ctxt L
$\top$	$\top$
$\bot$	$\bot$
U	$\bot$

These rules are schemas and need to be instantiated such that $I \in [1,n]$, $J \in [1,n-1]$, and X,Y,Z are different constants denoting the objects occurring in the premises of $\mathcal{S}$. We assume that the addition $J+1$ is computed while instantiating a rule. The rule in (2) states that if in phase I object X should be placed to the left of Y and the space to the left of X as well as the space to the right of X are empty, then X is placed as the left neighbor of Y. The rule in (3) keeps neighbors for succeeding phases. The rules in (4) ensure that neighbors take space, i.e., if X has become the left neighbor of Y in phase J, then the space to the left of Y as well as the space to the right of X are occupied in phase $J+1$. The rules in (5) implement the first free fit technique from (Ragni & Knauff, 2013), thus if X should be placed to the left of Y but there is already a left neighbor Z of Y, then X is placed to the left of Z. Likewise, if X should be placed to the left of Y but X is already the left neighbor of some other object Z, then Z should be placed to the left of Y. The final neighbors are derived by the rule in (6): If X is left neighbor of Y after processing all premises, then X is (finally) to the left of Y. The rules in (7) and (8) express that *left* is transitive and *right* is the inverse of *left*.

In each phase, one premise is processed and understood as a request to place the mentioned objects in the required order. Objects are placed in the first available space like in PRISM (see, Ragni & Knauff, 2013). Once the fixed point of $\Phi_{\mathcal{P}_\mathcal{S}}$ is computed the preferred model can be identified: Given a problem $\mathcal{S}$, X is the left neighbor of Y if and only if it holds that $\mathcal{P}_\mathcal{S} \models_{wcs} nl(X,Y,n)$. Queries involving the *left* and *right* relation can be answered with respect to the preferred model of $\mathcal{S}$.

Variation/ Inspection for Alternative Models

We now present the main result of this paper, viz. an approach to the model variation phase. Figure 1 shows the modeling process of the variation phase, which consists of several steps: First, all initial left placement requests (*il*), all positive neighborhood left relations (*nl*), and all positive *ambiguous* relations from the preferred model are extracted (*Extract relevant information from preferred mental model*). Thereafter, the program is constructed (*Create program*). Based on all extracted ambiguities, all permutations of all length are computed (*Compute all permutations*). The order of the items in each permutation is kept by the variation program through the phase-indices in the relations: The first item in a permutation is assigned the phase-index 1 and the last item the phase-index v. All different ways of swapping ambiguous objects are simulated. Considering all permutations, all alternative models of the spatial reasoning problem $\mathcal{S}$ can be found. Until all permutation have been processed, the following is done (*Iteration through list of permutations*): One trial of swapping objects is done per iteration, by swapping objects through *ambiguous relations* (*Program construction*). This is realized by adding the *ambiguous* relations of the current permutation as positive facts to the program (see (14) below). Moreover, the program needs to know which objects are affected by these swap-requests. Therefore, the implementation adds two positive facts for each added *ambiguous* fact, encoding that the objects in question need to be adjusted (see (13) below). Thereafter, it is checked whether the relation encoding the conclusion of the spatial reasoning problem is mapped to true or false in the fixed point of the semantic operator (*Compute fixed point*). The answer is saved, provided that the model is not marked as abnormal (see (21) and (22)). The program is reset, which includes deleting all facts regarding the *ambiguous* and *adjust* relations and resetting all atoms occurring in the program to unknown (*Reset program*). The iteration continues until all permutations have been processed. The final answer is given by checking whether the list of collected answers contains the answer "No" (*Final answer*). If that is the case, the final answer to the query is "No" because an alternative model has been found that does not support the conclusion. Otherwise, the final answer is "Yes".

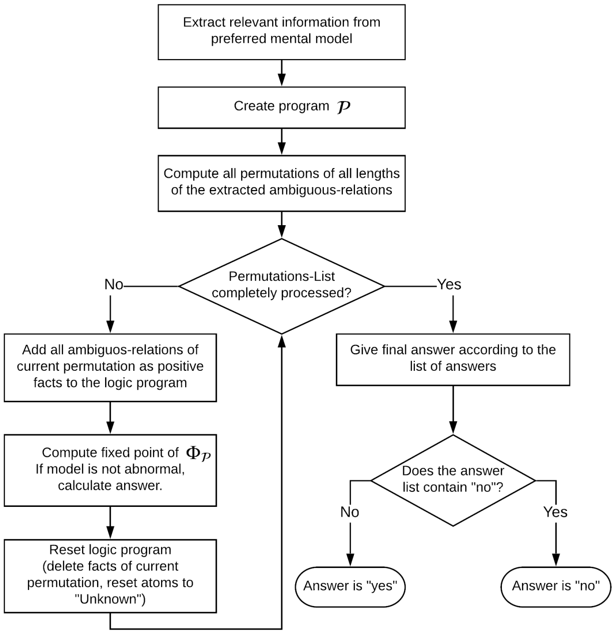

Extract relevant information from preferred mental model
Create program $\mathcal{P}$
Compute all permutations of all lengths of the extracted ambiguous-relations
Permutations-List completely processed?
No
Yes
Add all ambiguos-relations of current permutation as positive facts to the logic program
Compute fixed point of $\Phi_\mathcal{P}$
If model is not abnormal, calculate answer.
Reset logic program (delete facts of current permutation, reset atoms to "Unknown")
Give final answer according to the list of answers
Does the answer list contain "no"?
No
Yes
Answer is "yes"
Answer is "no"

Figure 1: Flowchart of the variation phase.

Ambiguity Identification

We first record the set of initial placements of the spatial reasoning problem by replacing each fact of the form (1) by

$$il(X,Y) \;\leftarrow\; \top. \tag{9}$$

We extend $\mathcal{P}_S$ to mark ambiguities in the model construction:

$$amb(Z,X,J+1) \leftarrow l(Z,Y,J+1) \wedge nl(X,Y,J). \tag{10}$$
$$amb(Z,Y,J+1) \leftarrow l(X,Z,J+1) \wedge nl(X,Y,J).$$
$$amb(Z,X,I) \leftarrow l(Z,Y,I) \wedge amb(X,Y,J+1). \tag{11}$$
$$amb(Z,Y,I) \leftarrow l(X,Z,I) \wedge amb(X,Y,J+1).$$

X, Y, Z, I, and J must be instantiated as before and we assume that $I > J$. Let $\mathcal{A}_S$ be the program consisting of all ground instances of clauses mentioned in this paragraph.

The rules in (10) record the ambiguities from neighbors: if object X is the direct left neighbor of object Y in phase I and there is a request to place a new object Z to the left of Y, then there is an ambiguity between Z and X, because both could possibly be the direct left neighbor of Y. Similarly, if X should be placed to the left of the new object Z, but X and Y are already in a direct left neighborhood, then Z and Y are ambiguous and could be swapped in order to obtain an alternative model. The rules in (11) record the inherited ambiguities: If a new object Z is requested to be set to the left of Y, but Y is already marked as ambiguous with respect to another object X, then Z is ambiguous with respect to X, too. Likewise, in case X is requested to be placed to the left of a new object Z with X and Y already being marked as ambiguous objects, then Z will also be ambiguous with respect to Y. It is important to note that these clauses need to be created for all phases I and J with $I > J$. This means that the *amb*-relation with phase index J does not necessarily need to be in the phase directly before I, but it can also be that $I = J+2$.

Program Construction for Alternative Models

Likewise to the construction of the program for the preferred model, the construction of the programs during variation uses phases as well. The neighbor left relations that have been generated by the preferred model will be used as starting point:

$$\{nl(X,Y,n) \leftarrow \top \mid \mathcal{P}_S \models_{wcs} nl(X,Y,n)\}. \tag{12}$$

First, the number of programs for the computation of alternative models (i.e. one program for one alternative model) is specified by the number of *amb* relations in the fixed point of $\Phi_{\mathcal{A}_S \cup \mathcal{P}_S}$:

$$\#\mathsf{perm} = \sum_{i=1}^{|amb|} \prod_{k=1}^{i} (|amb| - k + 1),$$

where

$$|amb| = |\{amb(X,Y,I) \mid \mathcal{A}_S \cup \mathcal{P}_S \models_{wcs} amb(X,Y,I), I \in [2,n]\}|.$$

The amount of phases v for each program depends on the number of *ambiguous*-relations with respect to the current permutation *pm*, i.e.

$$v_{pm} = |\{amb(X,Y,I) \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\}.$$

Second, the *amb* relations of the current permutation *pm* tells us which objects can be adjusted within the variation phase:

$$\{adj(X,I) \leftarrow \top \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\} \cup \tag{13}$$
$$\{adj(Y,I) \leftarrow \top \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\}.$$

In each phase I of the variation, two objects X, Y are swapped according to a swap-request of the form $amb(X,Y,I) \leftarrow \top$. Accordingly, the maximum phase index v refers to the number of phases in the variation process and the phase index n refers to the number of phases in the construction process of the preferred model. Since the model variation starts with the preferred model, that is, the last phase n of the preferred model, the last overall phase in the variation program is $n+v$. Together with the fact in (9), which will serve as constraint to prevent violating the premises of the given spatial reasoning problem, the set of neighbor relations in (12) of the preferred model, and the objects in (13) that have to be adjusted, each program $\mathsf{var}\mathcal{P}_S$ with its according considered permutation *pm*, where $I \in [1,|pm|]$, consists of the following clauses:

$$amb(X,Y,I) \leftarrow \top. \tag{14}$$
$$amb(Y,X,I) \leftarrow amb(X,Y,I). \tag{15}$$
$$left(X,Y) \leftarrow nl(X,Y,n+v). \tag{16}$$
$$left(X,Z) \leftarrow \mathsf{ctxt}\, left(X,Y) \wedge \mathsf{ctxt}\, left(Y,Z). \tag{17}$$
$$right(X,Y) \leftarrow left(Y,X). \tag{18}$$
$$nl(X,Y,Q) \leftarrow nl(X,Y,P) \wedge \overline{\mathsf{ctxt}}\, adj(X,I) \wedge \overline{\mathsf{ctxt}}\, adj(Y,I). \tag{19}$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Y,X,I) \wedge nl(Y,X,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X). \tag{20}$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Z,X,I) \wedge nl(Z,Y,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X).$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Y,Z,I) \wedge nl(X,Z,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X).$$

X, Y, Z, I, and J must be instantiated as before, n is the number of premises, v is the last phase in the variation model construction, $P = n-1+I$, and $Q = n+I$.

The fact in (14) ensures that all *amb*-relations from the current permutation are added as facts. The permutation order within a permutation *pm* is specified by the phase index I, starting from 1 to $|pm|$. These facts are requests to swap two objects in phase I. The rule in (15) expresses that the order of objects in an *ambiguous*-relation is irrelevant, as all orders lead to the same result. The rule in (16) specifies final neighbors. The rules in (17-18) are analogous to (7-8). The rule in (19) is similar to (3) except to the additional constraint that none of the concerned objects has to be adjusted in the current phase I. The rules in (20) encode the actual swapping of two objects. Altogether, there are three different cases how two objects can be swapped: Either the objects to be swapped are in the same *nl*-relation, or the left object in the *nl*-relation has to be swapped, or the right object in the *nl*-relation is requested to be swapped.

Incomplete Model or Constraint Violation

As the models constructed in the variation may be incomplete due to violated constraints, we include *abnormality* clauses in order to consider only *normal* models for computing answers with respect to the given problem query:

$$chain \leftarrow left(X_1,X_2) \wedge left(X_3,X_4) \wedge \cdots \wedge left(X_{n-1},X_n). \quad (21)$$

$$ab \leftarrow \neg chain.$$

$$ab \leftarrow left(X,Y) \wedge \mathsf{ctxt}(il(Y,X)). \quad (22)$$

$X,Y,X_1,\ldots,X_n \in con(\mathsf{init}\mathcal{P}\mathsf{rem})$ and $X,Y,X_1,\ldots,X_n$ are different to each other. The rules in (21) denote the case when the alternative model is not complete: If no chain can be constructed from the *left*-relations, then this model is marked as abnormal. The rule in (22) denotes the case when the alternative model violates some constraint. The case of constraint violation only concerns big, non-deterministic problems with five or more objects, for which the constraints contained in the bodies of the rules in (20) cannot prevent some of the violating swaps anymore.

In each phase of the model variation, two objects are swapped according to the swap-requests (by $amb(X,Y,i)$), until all requests in the current permutation have been processed. Likewise to the preferred model construction, the variation program will then proceed with mapping the *nl*-relations to *left*- and *right*-relations by the rules in (16), (17) and (18). After all *left*- and *right*-relations are determined, we can check in the alternative model whether there are any abnormalities. As soon as the fixed point of Φ with respect to the given program is computed, the alternative model can be identified, provided that the model is not abnormal, i.e. the atom *ab* is false.

Is the beetle (necessarily) left of the dodge?

Consider again the example from the introduction, where the preferred model is $a \quad b \quad c \quad d$.

This example has additionally two valid alternative models. Due to the limited space, we do not show the complete computation of the preferred mental model with marking ambiguities. The result of the computation are two marked ambiguities, one between the objects c and b and one between d and b. For a detailed explanation on what happens in each iteration when computing preferred models under the WCS, see the examples in Dietz et al. (2015).

The implementation determines four different permutations of the two *ambiguous*-relations, which are (1) $amb(c,b,1)$, (2) $amb(d,b,1)$, (3) $amb(c,b,1)$ and $amb(d,b,2)$, and (4) $amb(d,b,1)$ and $amb(c,b,2)$. We show the variation program exemplary for permutation (3) in Table 2, starting with the empty interpretation, leading to the alternative and valid model $a \quad c \quad d \quad b$.

The atoms *ambiguous* and *adjust* are abbreviated to *amb* and *adj* to fit the table. Furthermore, Table 2 only shows the atoms that appear in $I^\top$ and $I^\perp$ for the first time to maintain readability, as was done in Dietz et al. (2015). The column on the right side of the table signifies the clause which leads to the atoms shown in the respective row.

In iteration 2 and 3 in Table 2, the model obtained after processing the first swap-request is computed, which is $a \quad c \quad b \quad d$ (phase 4).

Thereafter the final alte rnative model is computed $a \quad c \quad d \quad b$, determining all *nl*-relations that hold in the model, as can be seen in iteration 3 and 4 (phase 5).

The answer to the query of the problem, $left(b,d)$ is determined in iteration 4. Since $left(b,d)$ is *False* in the fixed point of $\Phi_{\mathcal{P}}$, this relation does not hold in the alternative model. It does however hold in the preferred model $a \quad b \quad c \quad d$.

Conclusively, the final answer of our implementation is "No", because there was at least one model in which the relation described in the query did not hold.

Table 2: Alternative model computation with two swaps.

$\Phi_{\mathcal{P}}$	$I^\top$	$I^\perp$	clause nr./ program
$\uparrow 1$	$il(a,b), il(a,c),$		(9)
	$il(c,d),$		(9)
	$nl(a,b,3), nl(b,c,3),$		(12)
	$nl(c,d,3)$		(12)
	$amb(c,b,1),$		(14)
	$amb(d,b,2),$		(14)
	$adj(b,3), adj(c,3),$		(13)
	$adj(b,4), adj(d,4),$		(13)
$\uparrow 2$	$amb(b,c,1),$		(15)
	$amb(b,d,2),$		(15)
		$nl(a,b,4), nl(b,a,4), nl(c,a,4),$	(19)
		$nl(c,b,4), nl(c,d,4), nl(d,b,4),$	(19)
		$nl(d,c,4), nl(a,b,5), nl(b,a,5),$	(19)
		$nl(c,b,5), nl(d,a,5), nl(d,b,5),$	(19)
		$nl(d,c,5),$	(19)
	$nl(b,d,4),$		(20)
	$nl(a,c,4)$		(20)
$\uparrow 3$		$left(a,b), left(b,a), left(c,b),$	(16)
		$left(d,a), left(d,b), left(d,c)$	(16)
	nl(a, c, 5),	$nl(a,d,5), nl(b,c,5), \mathrm{ln}(b,d,5),$	(19)
		$nl(c,a,5),, nl(c,d,5)$	(19)
	$nl(c,b,4),$		(20)
	nl(d, b, 5)		(20)
$\uparrow 4$	$left(a,c),$	$left(a,d), left(b,c),$***left(b, d)***,	(16)
	$left(d,b),$	$left(c,a), left(c,d),$	(16)
		$right(a,b), right(a,d), right(b,a),$	(18)
		$right(b,c), left(b,d), right(c,d)$	(18)
	nl(c, d, 5)		(20)
$\uparrow 5$	$left(c,d),$		(16)
	$right(c,a),$	$right(a,c), right(c,b), right(d,a),$	(18)
	$right(b,d),$	$right(d,b), right(d,c)$	(18)
		chain	(21)
$\uparrow 6$	$left(a,d),$		(17)
	$left(c,b),$		(17)
	$right(d,c),$		(18)
	chain		(21)
	ab		(21)
$\uparrow 7$	$left(a,b),$		(17)
	$right(d,a),$		(18)
	$right(b,c)$		(18)
		ab	(21)
$\uparrow 8$	$right(b,a)$		(18)

Discussion and Conclusions

The contribution of this paper comprises various aspects within both the area of Computer Science and Psychology. Through the formal process of modeling the spatial reasoning task, we have had to put forward new hypotheses on the model variation phase which need to be verified in the future:

Cognitive complexity of alternative models The variation phase starts with the information provided on the preferred model. How is this related to the *cognitive complexity* for the construction of the individual alternative models?

List of permutations Is the list of permutations cognitively adequate? Do humans keep track of such a list, or does one permutation trigger the next one? If humans keep such a list, how likely do they make mistakes? Are these mistakes related to the *distance* of the preferred model?

Ambiguity identification We suggested to rigorously identify ambiguities within the task. Yet, humans might be sloppy in the sense that they recognize certain ambiguities more easily. If so, which are the selection criteria?

Default and explicit knowledge Two notions of negation, weak and strong negation, were necessary for modeling this task. How does this distinction relate to other tasks?

The Weak Completion Semantics has shown again to be a good candidate for a comprehensive and computational cognitive theory, as it seems to adequately model yet other aspects of human reasoning task not considered so far. The WCS can fully cover all three stages of reasoning that have been suggested by the preferred mental model theory. This is novel as the WCS has previously never been considered to model the variation phase or alternative models in such a rigorous way. In particular, it seems that only few approaches (e.g., *mReasoner* Khemlani & Johnson-Laird, 2013) deal with the processes of alternative model construction. From a cognitive point of view, this is a central step if we intend to understand actual human reasoning, as one main part of it is concerned with the construction of counter examples. Future work includes the application of the current approach to other human reasoning tasks, such as syllogistic reasoning and reasoning with (counterfactual) conditionals. Furthermore, a metric among the alternative models and with respect to the model transformation should be specified. Possibly this could depend on the cardinality of the list of permutations or, more interestingly, on the amount of steps within the fixed point computation of the Φ operator. An interesting starting point of investigation would be whether a certain experimental setup could make it possible to mimic the operator iteration, by providing participants the information sequentially.

References

Byrne, R. M. J. (1989). Suppressing valid inferences with conditionals. , *31*, 61–83.

Clark, K. L. (1978). Negation as failure. In H. Gallaire & J. Minker (Eds.), *Logic and data bases* (Vol. 1, pp. 293–322). New York, NY: Plenum Press.

Dietz, E.-A., Hölldobler, S., & Höps, R. (2015). A Computational Logic Approach to Human Spatial Reasoning. In *IEEE symposium series on computational intelligence, (SSCI 2015)* (pp. 1627–1634). IEEE.

Dietz Saldanha, E.-A., Hölldobler, S., Kencana Ramli, C., & Palacios Medinacelli, L. (2018). A Core Method for the Weak Completion Semantics with Skeptical Abduction. *J. of A. I. Res. Special Track on Deep Learning, Knowledge Representation, and Reasoning*, *63*, 51 – 86.

Dietz Saldanha, E.-A., Hölldobler, S., & Pereira, L. M. (2017). Contextual Reasoning: Usually Birds Can Abductively Fly. In M. Balduccini & T. Janhunen (Eds.), *Proceedings of 14th international conference on logic programming and nonmonotonic reasoning (lpnmr)* (Vol. 10377, pp. 64–77). Springer International Publishing.

Grice, H. P. (1975). Logic and Conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics: Vol. 3: Speech acts* (p. 41-58). New York: Academic Press.

Hölldobler, S. (2015). Weak Completion Semantics and its Applications in Human Reasoning. In U. Furbach & C. Schon (Eds.), *Proc. of workshop on bridging the gap between human and automated reasoning* (pp. 2–16). CEUR-WS.org.

Hölldobler, S., & Kencana Ramli, C. D. (2009). Logic Programs under Three-Valued Łukasiewicz Semantics. In P. M. Hill & D. S. Warren (Eds.), *Iclp* (Vol. 5649). Springer.

Johnson-Laird, P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge, MA: HARVARDUP.

Khemlani, S., & Johnson-Laird, P. N. (2012). Theories of the Syllogism: A Meta-Analysis. *Psy. Bulletin*, 427-457.

Khemlani, S., & Johnson-Laird, P. N. (2013). The processes of inference. *Argument & Computation*, *4*(1), 4-20.

Łukasiewicz, J. (1920). O logice trójwartościowej. *Ruch Filozoficzny*, *5*, 169-171.

Oliviera da Costa, A., Dietz Saldanha, E.-A., Hölldobler, S., & Ragni, M. (2017). A Computational Logic Approach to Human Syllogistic Reasoning. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. J. Davelaar (Eds.), *Proceedings of the 39th annual conference of the cognitive science society* (p. 883-888). Austin, TX: Cognitive Science Society.

Ragni, M., & Knauff, M. (2013). A Theory and a Computational Model of Spatial Reasoning With Preferred Mental Models. In *Psy. rev. 2013* (Vol. 120, pp. 561 – 588).

Rips, L. J. (1994). *The Psychology of Proof: Deductive Reasoning in Human Thinking*. Cambridge, MA: MIT Press.

Stenning, K., & van Lambalgen, M. (2005). Semantic Interpretation as Computation in Nonmonotonic Logic: The Real Meaning of the Suppression Task. *Cognitive Science*, *6*(29), 916–960.

Stenning, K., & van Lambalgen, M. (2008). *Human Reasoning and Cognitive Science*. Cambridge, MA.

Wason, P. (1968). Reasoning about a Rule. , *20*(3), 273–281.

Measuring the Influence of L1 on Learner English Errors in Content Words within Word Embedding Models

Kanishka Misra (kmisra@purdue.edu)
Hemanth Devarapalli (hdevarap@purdue.edu)
Julia Taylor Rayz (jtaylor1@purdue.edu)
Purdue University
West Lafayette, IN 47906 USA

Abstract

Recent works in Second Language Acquisition Literature and Corpus Linguistics have shown the interference of a person's first language (L1) when they process words in a new language. In this work, we build on the findings in two recent studies that explore the various differences in the lexico-semantic models of a person's L1 and L2 (English in their case), and test their hypotheses within the framework of two popular word vector models. This test is carried out by extracting erroneous content word errors from an annotated corpus of essays written by learners of English who belong to 16 different first languages. Specifically, we compare the vectors representations of the incorrect and correct-replacement word pairs in English as well as in the person's first language and find a moderate correlation between L1 and English. Additionally, we find certain inconsistencies between the two word embedding models when observed under the radar of language typology, suggesting new avenues for future work.

Keywords: L1 influence on L2; Natural Language Processing; Semantic Overlaps between L1 and L2

Introduction

While writing in a non-native language, people often make wrong word choices. For example, French speakers often use *scene* in place of *stage* when writing in English. Observations such as these are often a result of a transfer of properties from the speakers Native Language (L1) during their Second Language (L2) acquisition. In this paper, we investigate whether models for distributed representations of words capture this transfer of L1 semantic knowledge based on the errors made by learners of English; and if they do, whether the observations are similar to results from previously conducted experiments.

Patterns of lexical choice in content produced by non-native speakers have been widely studied by Second Language Acquisition (SLA) and Natural Language Processing (NLP) researchers. It has been shown that a person's native language L1 influences their L2 acquisition in morphological, phonological, syntactical and semantic aspects (Groot, 1992; Koda, 1993; De Groot & Keijzer, 2000; Hopman, Thompson, Austerweil, & Lupyan, 2018). The semantic influence of L1 over L2 has been studied by SLA researchers in behavioral studies (Prior, MacWhinney, & Kroll, 2007; Degani & Tokowicz, 2010; Bracken, Degani, Eddington, & Tokowicz, 2017) as well as corpus analysis (Gilquin, Granger, et al., 2011). Within NLP, errors in lexical choice have been analyzed based on their detection or correction (Ng et al., 2014; Rozovskaya & Roth, 2010, 2011; Chang, Chang, Chen, & Liou, 2008; Futagi, Deane, Chodorow, & Tetreault, 2008; Dahlmeier & Ng, 2011).

Word Choice by Second Language Acquisition Research

Degani and Tokowicz (2010) found that translation ambiguity occurs when there is an indirect mapping between translations of a word. Earlier works in SLA have highlighted the role of cross-lingual translation and semantic ambiguity in L2 acquisition. In an experiment with word translations from 40 English and Spanish bilinguals, Prior et al. (2007) found that the overlap between the words across the two languages was highly correlated with the translation choices made by the bilinguals. This was further confirmed by Boada, Sánchez-Casas, Gavilán, García-Albea, and Tokowicz (2013), where the presence of translation ambiguity proved to be challenging to recognize words for Spanish and Catalan bilinguals, as compared to when words only had one translation in the L2. A more recent study by Bracken et al. (2017) introduced a new metric known as Translation Semantic Variability (TSV) that measures the meaning similarity between translations, as conducted by participants who were trained to translate German-English word pairs. The TSV was found to be a predictor in measuring the learning of translation-ambiguous German words, i.e., the accuracy of learning fell when the relatedness between the German and English word was low (Bracken et al., 2017), further highlighting the importance of ambiguity in early acquisition of an L2.

Word Choice in Corpus Analysis

The influence of L1 on errors in lexical choice in learner corpora has been studied based on functional words as well as combinations of content words. Rozovskaya and Roth (2010, 2011) improved on correcting errors in preposition usage made by learners of English by inducing error-probabilities made by learners in their L1 from external corpora. Siyanova and Schmidt (Siyanova & Schmitt, 2008) showed that learning of content word combinations and collocations has also been shown to be a challenging task for non-native speakers of English. Chang et al. (2008) introduced a system to detect and correct mis-collocations of words in English content produced by Chinese speakers. Their system benefited from consulting parallel English-Chinese collocation dictionaries.

More recently, Kochmar and Shutova (2016, 2017) analyzed the L1 effects on L2 semantic knowledge using three types of content word combinations (Adjective-Noun, Verb-Direct Object, and Subject-Verb). They addressed L2 acquisition across a spectrum of proficiency, as well as within different language families of the learner L1s. We are interested in three hypotheses (out of five) that were tested in these papers: (1) L1 lexico-semantic models influence lexical choice in L2; (2) L1 lexico-semantic models are portable to other typologically similar languages; (3) typological similarity between L1 and L2 facilitates semantic acquisition of knowledge in L2. For hypothesis (1), it was found that semantic models of lexical choice derived from a learners L1 helped in improving error detection in the content word combinations. This improvement was also observed in the case of errors made by learners belonging to typologically similar L1s, as hypothesized by (2). Additionally, within language typology (hypothesis (3)), lexical distributions of content word combinations were found to be closer to native English for distant L1s, as compared to closer L1s. This contradicted the authors original assumptions that Germanic L1s would be closest to Native English. In particular, their experiment showed that the lexical distributions of Romance L1s and Asian L1s were closer to that of Native English, as compared to that of Germanic L1s. The authors speculated that this result was due to (1) the usage of prefabricated word combinations by speakers of typologically different L1s, which makes their distribution more native-like, and (2) the adventurous experimentation carried out by proficient speakers, especially observed among those that speak languages closer to English, where new (although incorrect) expressions are created.

Word Embeddings

Recent research within NLP has seen the emergence of neural network-based models of distributed word representations, also called word embeddings. Neural word embeddings were first introduced by Bengio, Ducharme, Vincent, and Jauvin (2003) and, after their reemergence due to the popularity of word2vec (Mikolov, Chen, Corrado, & Dean, 2013), have become an integral part of NLP research (Bojanowski, Grave, Joulin, & Mikolov, 2016). These word representations have found to capture semantic information of words by treating words as multi-dimensional vectors, such that words with similar contexts have similar vectors. Recent development in the intrinsic evaluation of these embeddings have highlighted their competent performance in comparison to human judgments. Specifically, word embeddings have achieved high correlation to humans in tasks involving the judgment of semantic similarity and relatedness between words such as WS-353 (Finkelstein et al., 2002), MEN (Bruni, Boleda, Baroni, & Tran, 2012), SimLex-999 (Hill, Reichart, & Korhonen, 2015). Word embeddings also exhibit the capability to solve verbal analogies, for example, king - man + woman = queen, which has attracted the attention of the Cognitive Science community. A recent study (Chen, Peterson, & Griffiths, 2017) analyzed two popular word embedding models, GloVe (Pennington, Socher, & Manning, 2014) and word2vec (Mikolov et al., 2013), as accounts of analogy to evaluate their performance in a relational similarity task. Chen et al. (2017) showed that the models capture certain forms of similarities more than others. Word embeddings have been used in SLA literature as well. Word embedding based similarity measures were successful in predicting L2 word learning accuracy (Hopman et al., 2018). Vector representations of words have been successful in improving error detection on learner corpus of essays (Kochmar & Shutova, 2016). Since word embedding models have been shown to capture certain semantic properties observed in language, we explore whether they capture patterns that were found by earlier work in the analysis of content word errors made by learners of English. Specifically, we explore the relationship of word errors in L2 and the learners L1 using distributed representations of words, following Kochmar and Shutova (2016, 2017). We are interested in the following questions:

1. Do distributed representations of words reflect L1 influence on learner English error words?
2. Does distributed representation of learner English error words exhibit similar relationships between typologically similar languages?

In order to approximate the extent of influence of L1, as represented by word embeddings, we take the incorrect-correct pairs in their present state (English), and compare them with their translated form in the learners' first language (L1). The influence is approximated by correlation between the closeness of the incorrect and correct words in each of the languages embedding spaces, i.e., a positive correlation might indicate some signal showing influence of L1 on the errors made in English. We compute the closeness of the incorrect and correct words based on their vector space neighborhood. Given the various word vectors, cosine similarities offer a good way to calculate a word's nearest neighbors, these represent words that are most related to the word (Hill et al., 2015). We assume that the closer two neighbors are in the L1 space, the easier they are to confuse in a typologically close L2 space. We introduce a metric that measures the closeness and using correlation between the closeness in L1 and L2, approximate a possible influence.

Methodology

In order to answer the questions presented above, we use an error annotated corpus where the errors are made by people whose native language is different from English. We use the Cambridge - First Certification in English (FCE) corpus (Yannakoudakis, Briscoe, & Medlock, 2011) which is a small subset of the Cambridge Learner Corpus (Nicholls, 2003). The FCE examination falls under the B2 proficiency category of the Common European Framework of Reference for Languages (CEFR). In the CEFR framework, language proficiency is organized in 6 categories, ranging from A1 (lowest) to C2 (highest). The FCE corpus contains error annotated short essay responses by learners of English taking the First Certification in English examination. There are 16 different L1 backgrounds represented in the 2488 different short essays. The errors in the corpus are annotated, including the linguistic information such as the type of error and the part of speech involved in the correction, as well as the correct replacement. The annotation follows the scheme provided by (Nicholls, 2003). We chose this corpus because it is the only freely available corpus for learner English with error annotations and suggested replacements.

We only consider the annotations involving a replacement of a content word. Based on the annotation scheme, the replacement category for content word errors have been labelled as RX where X indicates the part of speech of the word in that context. For the purposes of this research, only Nouns (N), Adjectives (J), Verbs (V), and Adverbs (Y) have been considered as content words. Furthermore, we ignore the semantic errors containing multi-word expressions or phrases, or errors counted as replacements but also containing misspellings. The incorrect-correct content word pairs were extracted based on the given criteria, resulting in a total of 5521 cases of incorrect usage of a content word, and its replacement.

Translation of Error Pairs into L1

Since each of the essays contained learners L1, the extracted incorrect words as well as the corrected words suggested by annotation we will refer to then as incorrect and correct word pairs were translated from English (L2) into the learners L1 using the Microsoft Azure Text Translator API. This was used in place of the widely used (for instance, in Hopman et al. (2018)) off-the-shelf Google Cloud Translator API, since the latter only provides one-to-one word translations, without providing much choice about the part of speech, or the confidence with which it predicts a certain translation, both of which were available in the Azure API. Translations that resulted in word utterances rather than a single word, as well as errors made by Dutch L1 speakers (only 5 cases) were discarded, resulting in a total of 4932 incorrect and correct word pairs (known as L1 and L2 pairs respectively, hereafter). Table 1 describes the number of semantic error cases for the various L1s used in the experiment.

Table 1: Number of Error Cases per language (L1).

L1	***n***	**L1**	***n***
Spanish	796	German	285
French	794	Portuguese	284
Greek	353	Turkish	272
Russian	340	Japanese	192
Italian	335	Korean	185
Catalan	325	Thai	122
Chinese (Simplified)	310	Swedish	44
Polish	295		

Distributed Representation of Words

Word embeddings provide mapping between words and their vectors in a multi-dimensional space, such that the semantic properties of the words are preserved. Since our final selection consists of content word-based errors and has a multilingual element to it, we use embeddings trained on corpora in multiple languages. Moreover, we compare different models that were produced using different parameters and different corpora. Specifically, we use:

1. **polyglot:** a word representation with embeddings for over 100 languages (Al-Rfou, Perozzi, & Skiena, 2013). This embedding learns a 64-dimension vector for each word by scoring the word's surrounding context, and a corrupted context (the selected word swapped out randomly).

2. **fasttext:** a word representation with embeddings for over 100 languages (Bojanowski et. al, 2016). In fasttext, each word vector is composed by summing up vectors of the subwords of the word (specifically, 3-6 character ngrams) and is trained using skipgrams along with negative sampling.

Error Pair Neighbor Overlap

To measure the differences between the incorrect and correct word in a given language, the semantic properties of their vectors in the distributed vector space are taken into account.

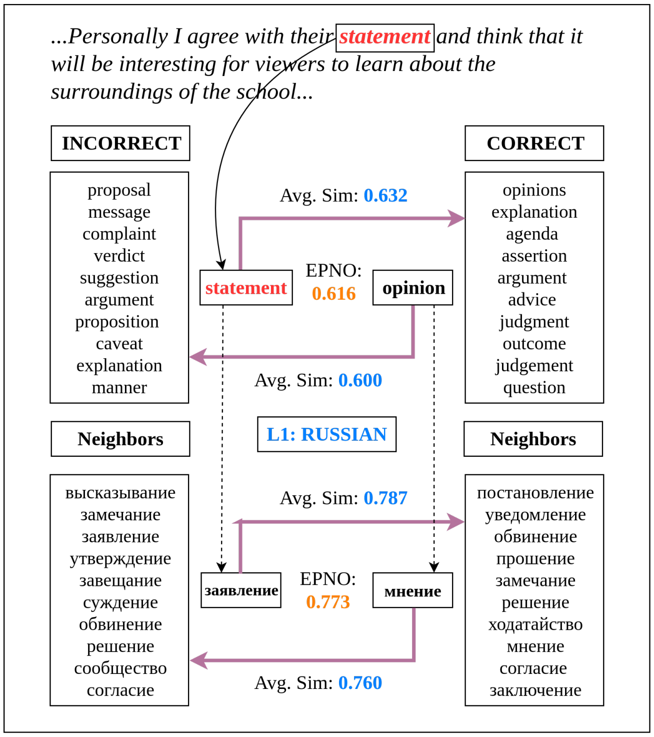

...Personally I agree with their statement and think that it will be interesting for viewers to learn about the surroundings of the school...
INCORRECT
CORRECT
proposal message complaint verdict suggestion argument proposition caveat explanation manner
Avg. Sim: 0.632
statement
EPNO: 0.616
opinion
opinions explanation agenda assertion argument advice judgment outcome judgement question
Avg. Sim: 0.600
Neighbors
L1: RUSSIAN
Neighbors
высказывание замечание заявление утверждение завещание суждение обвинение решение сообщество согласие
Avg. Sim: 0.787
заявление
EPNO: 0.773
мнение
постановление уведомление обвинение прошение замечание решение ходатайство мнение согласие заключение
Avg. Sim: 0.760

Figure 1: Visual Depiction of Computing EPNOs for (i,c) pairs in English and the person's L1 (Russian in this case). The context line is provided along with all the neighbors of the words.

More formally, given the incorrect-correct word pair, (i,c), the semantic overlap between i and c is computed. We introduced the Error Pair Neighbor Overlap (EPNO) to quantify the semantic relatedness between the incorrect word and correct word in terms of their nearest neighbors in the vector space, by relying on the idea that if the two words have a high semantic overlap, they will have related neighboring vectors. Mathematically, the *EPNO* for words *i* and *c* in language *L* is computed as:

$$EPNO_L(i,c) = \frac{1}{2k}\Big[\sum_{c' \in NN_k^L(c)} cos(i,c') + \sum_{i' \in NN_k^L(i)} cos(c,i')\Big] \tag{1}$$

where $NN_k^L(x)$ is a set of k nearest neighbors for word x in vector space for language L, and $cos(x,y)$ is the cosine similarity between vectors x and y. For our experiments, k is kept as 10. While cosine similarity shows a direct similarity between two vectors, EPNO computes the degree to which a given word (x) is related to words that are most similar to the second word (y),and vice-versa. Figure 1 shows a visual example of an error made by a native speaker of Russian, where EPNO values are calculated for *statement* (incorrect) and *opinion* (correct replacement) as well as for their respective Russian translations. The nearest neighbors, along with a context where the incorrect word occurs, are also provided for both cases.

Research Question 1

The first question that we would like to explore is: *Whether distributed representations of words reflect L1 influence on learner English error words.*

Experiment In order to approximate the influence of L1 on learner errors, EPNO values are computed for the L1 and the respective translated L2 word pairs over the fasttext as well as polyglot vector spaces. Japanese L1s were left out of the polyglot embeddings due to difficulty in feeding the text into the polyglot package. In order to check whether embeddings capture the L1 influence on learner English, the Spearman's Rank Correlation Statistic (ρ) between the overlaps in English as well as the L1 pairs was computed. Spearman's ρ calculates the monotonic relationship between the two variables. A significant correlation between the overlaps sustained across languages would indicate a potential role of L1 in influencing errors made by the learner. To test the significance for ρ for different languages, the p-values are computed along with the 95% bootstrap confidence intervals over 1000 resamples for each language. The resulting correlation estimates between the overlaps along with their p-values are shown in Table 2, while the bootstrap confidence intervals are shown in Figure 2.

Results As can be seen from Table 2 and Figure 2, the fasttext and polyglot EPNOs between L1 and English incorrect-correct word pairs have a moderately positive Spearman's ρ. In the case of Polyglot, errors committed by learners who speak Thai had a non-significant negative correlation, the rest (apart from Japanese L1) showed a significant correlation estimate between L1 and English. All languages within fasttext had significant positive correlations overall ($p < 10^{-3}$).

Discussion The results demonstrate a significant positive relationship between the EPNOs of error word pairs in English and the learners L1 for almost all languages, with the exceptions of Thai (non-significant) and Japanese (not included) in the case of Polyglot. A significant positive correlation shows that the incorrect-correct word pairs that are highly overlapping with each other in a person's L1 also highly overlap in English, indicating equal strength between the similarities in L1 and L2. These observations are consistent with findings reported by Kochmar and Shutova (2016), where L2 error detection accuracy improved when L1 lexico-semantic models were used as predictors, where their model showed improvement in differentiating error words from correctly used ones.

Table 2: Spearman's ρ between L1 and L2 overlaps in the error word pairs for fasttext and polyglot embeddings.

L1	fasttext	polyglot
Catalan	0.403 (<.001)	0.312 (<.001)
Chinese (Simplified)	0.588 (<.001)	0.322 (<.001)
French	0.477 (<.001)	0.373 (<.001)
German	0.505 (<.001)	0.384 (<.001)
Greek	0.489 (<.001)	0.351 (<.001)
Italian	0.565 (<.001)	0.355 (<.001)
Japanese	0.457 (<.001)	*NA*
Korean	0.366 (<.001)	0.281 (<.001)
Polish	0.546 (<.001)	0.356 (<.001)
Portuguese	0.543 (<.001)	0.369 (<.001)
Russian	0.552 (<.001)	0.129 (.025)
Spanish	0.539 (<.001)	0.351 (<.001)
Swedish	0.573 (<.001)	0.516 (<.001)
Thai	0.373 (<.001)	0.006 (.953)
Turkish	0.492 (<.001)	0.369 (<.001)

Note: Correlation Estimates and *p* values are listed as estimate (p-value)

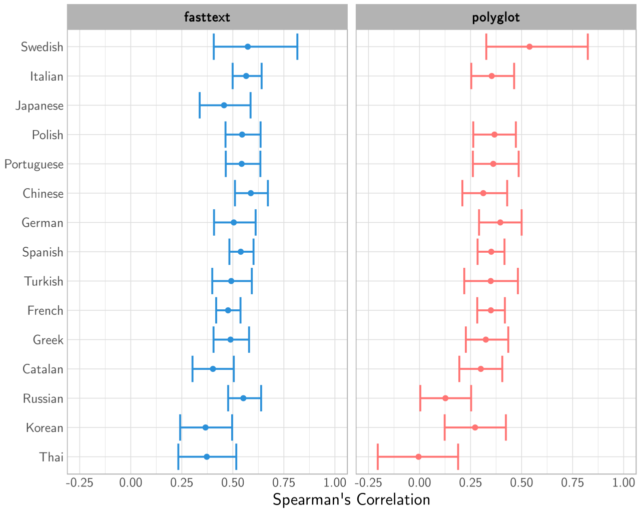


Figure 2: Spearman's ρ estimates of EPNOs computed for L1 and English incorrect-correct word pairs.

Research Question 2

The second question that we explore is: *Whether the similarity between semantic information of English and typologically closer L1s can be captured by fasttext and Polyglot.*

Experiment In this analysis, the same initial assumption made by Kochmar and Shutova (2017) was followed, i.e., L1s belonging to the same typological family will have similar EPNOs. For example, Germanic L1s should be closest to English based on their EPNO. The closeness with English is measured by the difference between the L1 and the English EPNO values computed for the fasttext and polyglot spaces. Based on our corpus, five groups of languages are considered: Germanic, Romance, Asian, Slavic, and an Other category to store the rest of the L1s. While we report the results, we will discard the Other category in the analysis since this combination is linguistically meaningless. The L1-English EPNO differences are computed as average differences over 1000 random samples (with replacement) within each group for 10,000 iterations. The notations $d_{fasttext}$ and $d_{Polyglot}$ denote these differences. Then, a one-way ANOVA is carried out to test for significance between the group L1-English differences. Table 3 lists the various languages covered in each group and their EPNO differences with English.

Results Table 3 reveals that for fasttext, the Asian family of languages in the corpus had the least difference between the EPNO values, followed by Slavic, Romance, and finally the Germanic. In contrast, for polyglot the differences observed for Germanic were the lowest, followed by Romance, Asian, and finally the Slavic. From the ANOVA results, the group L1-English differences were found to be significantly different from each other for both fasttext ($F(4, 49995) = 16539$, $p < 2 \times 10^{-16}$), and polyglot ($F(4, 49995) = 128751$, $p < 2 \times 10^{-16}$). A post-hoc Tukey HSD test revealed statistically significant pairwise difference between each of the groups except those with Slavic ($p = 0.131$).

Discussion The results observed in Table 3 reveal contrasting (although statistically significant) observations between differences in overlaps computed in fasttext and Polyglot. Based on the typology of languages, English falls under the Germanic family. However, the difference in the overlaps between the error pairs of Germanic L1s and English is the highest when computed for fasttext, with the least being the Asian L1s. In case of differences observed in the polyglot space, the opposite observation is made. The observations made in fasttext align with the findings of Kochmar and Shutova (2017), where Asian L1s were found to be closest to English in case of certain word pairs in the B2 proficiency category (same as our corpus), while Germanic L1s were found to be the farthest. On the other hand, the polyglot differences between L1 and English aligned with the initial assumptions made by Kochmar and Shutova (2017). The inconsistencies between fasttext and polyglot can be attributed to several factors. First, their dimension size and vocabulary: fasttext contains 300 dimensional vectors and an average vocabulary size is in the order of 10 million, while polyglot has 64 dimensional vectors with an average vocabulary size between 10,000 to 100,000. The difference in vocabulary size may dictate the choice in the neighbors for each overlap computation. Second, the nearest neighbors: fasttext incorporates the usage of subwords in its training along with the context of the words themselves, while polyglot follows only the contextual route. For example, the word *almost* has the following neighbors in fasttext: *nearly, practically, virtually, almsot, Almost, amost, alsmost, alomst, damn-near, pretty-much*; while in Polyglot: *nearly, once, roughly, just, equally, virtually, somewhat, less, absolutely, slightly*. The neighbors in fasttext could contribute to the noise while measuring the overlaps, thus distorting the results.

Table 3: Differences between L1 and English EPNOs for each Language Family in the Corpus.

Group	Languages	$d_{fasttext}$	$d_{Polyglot}$
Germanic	German Swedish	0.135	**0.184**
Romance	Spanish Catalan Italian French Portuguese	0.129	0.188
Slavic	Russian Polish	0.127	0.226
Asian	Chinese Japanese* Korean Thai	**0.123**	0.217
Other	Turkish Greek	0.128	0.195

* Japanese was ignored in the analysis of Polyglot. The bold formatted values highlight the minimum value in the respective column.

Conclusion

By analyzing content word errors in a corpus of learner English using two different word embedding models, we found (1) a significantly positive relationship between the error words in a learner's L1 and English, and that (2) while fasttext vector spaces emulate the results reported by Kochmar and Shutova (2016), the polyglot vector spaces are consistent with their initial assumptions. We speculate that the inconsistencies between fasttext and polyglot could be attributed to their inherent differences, namely: the dimensionality and vocabulary size, resulting in nearest neighbor choices. Due to the small size of the corpus, we unable to analyze the specific relationships within the different parts of speech used in the content word set, which could shed more light on the differences between the two embedding models.

References

Al-Rfou, R., Perozzi, B., & Skiena, S. (2013, July). Polyglot: Distributed Word Representations for Multilingual NLP. *arXiv:1307.1662 [cs]*.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Boada, R., Sánchez-Casas, R., Gavilán, J. M., García-Albea, J. E., & Tokowicz, N. (2013). Effect of multiple translations and cognate status on translation recognition performance of balanced bilinguals. *Bilingualism: Language and Cognition*, *16*(1), 183–197.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016, July). Enriching Word Vectors with Subword Information. *arXiv:1607.04606 [cs]*.

Bracken, J., Degani, T., Eddington, C., & Tokowicz, N. (2017). Translation semantic variability: How semantic relatedness affects learning of translation-ambiguous words. *Bilingualism: Language and Cognition*, *20*(4), 783–794.

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1* (pp. 136–145).

Chang, Y.-C., Chang, J. S., Chen, H.-J., & Liou, H.-C. (2008). An automatic collocation writing assistant for taiwanese efl learners: A case of corpus-based nlp technology. *Computer Assisted Language Learning*, *21*(3), 283–299.

Chen, D., Peterson, J. C., & Griffiths, T. L. (2017). Evaluating vector-space models of analogy. *arXiv preprint arXiv:1705.04416*.

Dahlmeier, D., & Ng, H. T. (2011). Correcting semantic collocation errors with l1-induced paraphrases. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 107–117).

Degani, T., & Tokowicz, N. (2010). Ambiguous words are harder to learn. *Bilingualism: Language and Cognition*, *13*(3), 299–314.

De Groot, A. M., & Keijzer, R. (2000). What is hard to learn is easy to forget: The roles of word concreteness, cognate status, and word frequency in foreign-language vocabulary learning and forgetting. *Language Learning*, *50*(1), 1–56.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., et al. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, *20*(1), 116–131.

Futagi, Y., Deane, P., Chodorow, M., & Tetreault, J. (2008). A computational approach to detecting collocation errors in the writing of non-native speakers of english. *Computer Assisted Language Learning*, *21*(4), 353–367.

Gilquin, G., Granger, S., et al. (2011). From efl to esl: evidence from the international corpus of learner english.

Groot, A. M. de. (1992). Determinants of word translation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(5), 1001.

Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665–695.

Hopman, E. W. M., Thompson, B., Austerweil, J. L., & Lupyan, G. (2018). Predictors of L2 word learning accuracy: A big data investigation. In (p. 6).

Kochmar, E., & Shutova, E. (2016, August). Cross-Lingual Lexico-Semantic Transfer in Language Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 974–983). Berlin, Germany: Association for Computational Linguistics.

Kochmar, E., & Shutova, E. (2017, September). Modelling semantic acquisition in second language learning. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 293–302). Copenhagen, Denmark: Association for Computational Linguistics.

Koda, K. (1993). Transferred l1 strategies and l2 syntactic structure in l2 sentence comprehension. *The Modern Language Journal*, *77*(4), 490–500.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In *Proceedings of the eighteenth conference on computational natural language learning: Shared task* (pp. 1–14).

Nicholls, D. (2003). The Cambridge Learner Corpus - error coding and analysis for lexicography and ELT. , 10.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Prior, A., MacWhinney, B., & Kroll, J. F. (2007). Translation norms for english and spanish: The role of lexical variables, word class, and l2 proficiency in negotiating translation ambiguity. *Behavior Research Methods*, *39*(4), 1029–1038.

Rozovskaya, A., & Roth, D. (2010). Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 961–970).

Rozovskaya, A., & Roth, D. (2011). Algorithm selection and model adaptation for esl correction tasks. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - volume 1* (pp. 924–933).

Siyanova, A., & Schmitt, N. (2008). L2 learner production and processing of collocation: A multi-study perspective. *Canadian Modern Language Review*, *64*(3), 429–458.

Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. , 10.

Method of Development of Interactive Agents Grounding the Cognitive Model to the Virtual World

Junya Morita (j-morita@inf.shizuoka.ac.jp)
Kazuma Nagashima (cs16065@s.inf.shizuoka.ac.jp)
Yugo Takeuchi (takeuchi@inf.shizuoka.ac.jp)
Faculty of Informatics, Shizuoka University,
3-5-1 Johoku, Naka-ku, Hamamatsu City, JAPAN

Abstract

Toward the realization of cognitive agents that interact with humans, this research attempts to integrate the cognitive architecture ACT-R and a 3D game engine. We built a hierarchical architecture in which ACT-R and the game engine were connected through a blackboard server, and we constructed a cognitive model for searching the 3D environment. The constructed model reproduced behavioral differences by following parameters of the cognitive model. We also made interesting errors related to the brain-body connection. From these results, it is suggested that the method of cognitive modeling is useful for constructing agents that imitate human behaviors in 3D space.

Keywords: ACT-R; virtual agent; game engine

Introduction

There are several approaches to the ultimate goal of building human or animal-like artificial agents. In the field of human-agent interaction (HAI), researchers have attempted to achieve this goal by focusing on interactions between artificial agents and human users. By considering intelligence as emergent properties of interactions, researchers have developed physical robots and virtual agents that can interact with humans, and they have conducted psychological experiments to examine human reactions to the implemented agents. Throughout these efforts, researchers have tended to emphasize visual appearance (Minato, Shimada, Ishiguro, & Itakura, 2004) or social relationships (Reeves & Nass, 1996) rather than the internal representation and internal processing of agents.

Meanwhile, the method of implementing human nature into internal representations and processing them into artificial agents has been traditionally studied in the community of cognitive modeling, which is a traditional research approach that combines artificial intelligence researches and psychological studies in the field of cognitive science. In this community, cognitive models are assumed to be hypotheses of a human's internal processing, which are represented as a computational system. Unlike other artificial intelligence researches, the study of cognitive modeling focuses on reproducing human errors, biases, and bounded rationality (Simon, 1996) found in psychological studies, which are evaluated by simulation studies reproducing the results of psychological experiments.

Despite dealing with similar topics, not much knowledge has been exchanged between the two communities. For HAI researchers dealing with human response to agents as the main data, deep internal processing might not be of interest. However, in the future, when HAI handles complicated and long-term interaction series more often, the development of agents that include internal processing, as dealt with in cognitive modeling, will be required.

From the above recognition, the authors explored the development method of an interactive agent that involved a cognitive modeling approach. In particular, this paper aims to discuss approaches toward this goal and research topics derived from the developed approach. In the following sections, we first discuss the approach of integrating HAI and cognitive modeling along with previous findings in the related fields. Based on this approach, we then present our system and a preliminarily experiment to discuss its usefulness in HAI studies.

Integrating HAI and Cognitive Modeling

Cognitive Architecture

In the cognitive modeling community, the role of cognitive architectures has become increasingly important. Cognitive architectures are the basis for integrating methods developed in individual studies of cognitive modeling. By accumulating the findings obtained from individual model development, it is thought that the structure of a universal cognitive system can be approached (Newell, 1990). Several cognitive architectures have been developed so far. In the current research, we focused on ACT-R (Anderson, 2007). ACT-R has been developed in the community, where many researchers participate. In addition, psychological and physiological studies have been conducted to associate the modules and parameters of the architecture with the brain structure (Anderson, 2007) and physiological functions (Dancy, Ritter, Berry, & Klein, 2015). Although the original ACT-R is described in Lisp, there are also implementations in multiple programming languages, including Java (Harrison, 2002) and Python (Stewart & West, 2005), making it possible for it to be developed flexibly depending on each individual developer's environment.

Connect to the Virtual World

ACT-R has several modules that are not only related to internal processing, including goal, declarative, and imaginal, but also used for interaction with the external environment, including perception and motors. However, these interactive modules do not include sensors that acquire physical signals or actuators that interact directly with the physical world. In other words, to construct an interactive agent using ACT-R, it is necessary to prepare a separate body to be connected with ACT-R. Regarding this problem, Trafton et al. (2012) implemented ACT-R on a humanoid robot that was able to interact with humans in the real world although its interactions are limited because of hardware limitations.

Considering such implementation difficulties, the current research adopts a virtual agent in a three-dimentional (3D) world instead of physical robot. To build a 3D virtual world, we used a game engine. Many game engines developed in recent years include sophisticated physical engines and body models, and they can build worlds with high reality. Recently, several studies linking these 3D environments and ACT-R have appeared. One study has developed a virtual humanoid robot that determines simple actions, such as walking and rotation, according to its perception of the 3D environment (Somers, 2016), and another study has developed a virtual robot that searches a maze environment in the virtual world while constructing a map of its environment (Smart, Scutt, Sycara, & Shadbolt, 2016). Based on the findings of the previous studies, the current research extends the scope of application while developing a novel architecture that links an ACT-R model with the virtual world.

Integrating Cognitive Architecture and the Virtual World
When connecting ACT-R to the virtual world, we need to solve a problem derived from different time scales of the two systems. In the virtual world, multiple independent events usually proceed in real time. By contrast, the process occurring within ACT-R is sequential. Therefore, for the integration of ACT-R and the virtual world, a framework such as the Subsumption Architecture (Brooks, 1986), which organizes sub-behaviors into hierarchical layers, is required to run processes of different layers in parallel. In other words, the control of body movement in the virtual world occurs in the lower layer, and decision making based on knowledge representation by ACT-R occurs in the upper layer. Both of these layers operate in parallel while communicating at regular intervals. The upper layer decides upon an action based on inference with a knowledge base while inputting the perceptual information acquired in the lower layer. The lower layer receives the decision of the upper layer as a command and transforms it to perform low-level body movement (walking, changing posture, turning around, etc.).

System

Architecture

We implemented a prototype hierarchical system that connects ACT-R (Python ACT-R) and a game engine (Unreal Engine 4) via a blackboard server (Figure 1). The server was implemented in C language, and had slots for storing action commands from agents and slots for storing visual information obtained from the environment. The value of each slot was updated via periodic socket communication from the game engine or ACT-R. With reference to past research (Somers, 2016; Smart et al., 2016), the data format used for communication was unified to JavaScript Object Notation (JSON).

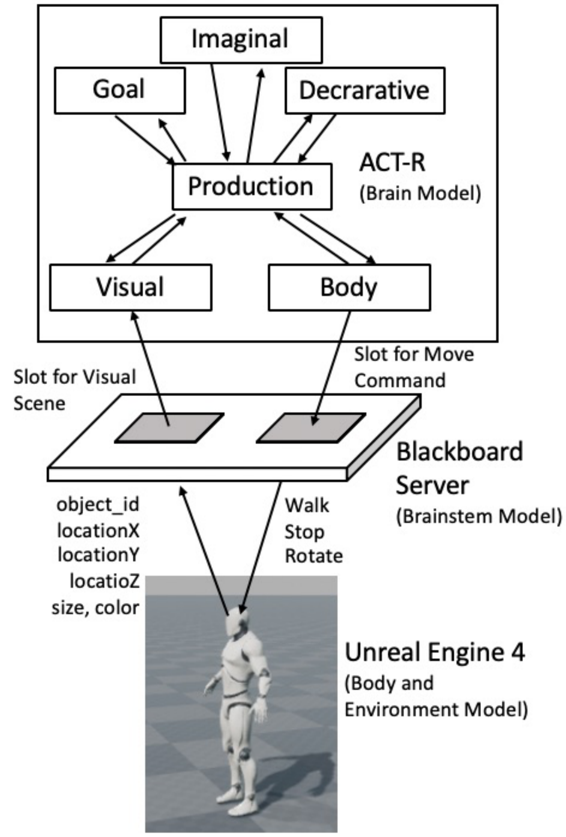


Figure 1: Architecture connecting ACT-R and the virtual world.

According to Anderson (2007), the ACT-R modules correspond to brain regions: the production module to the basal ganglia, the visual module to the visual cortex, the motor module to the motor cortex, the imaginal module to the parietal lob, the goal module to the anterior cingulate cortex, and the declarative module to the prefrontal cortex. Therefore, in this architecture, we assumed that the server corresponds to the brainstem connected to the brain model (ACT-R) with the virtual body, which have several movement patterns. The ACT-R architecture communicates with the server to monitor the state of the body, and to send a command for the next movement pattern, and to interrupt the current movement when necessary.

Task and Model

To test the above architecture, we implemented an agent that performs a simple environment search with the constructed architecture. Figure 2 shows the 3D environment in which the agent is located. A bird's-eye view is shown in the upper left, and a visual perspective of the agent is shown in the lower

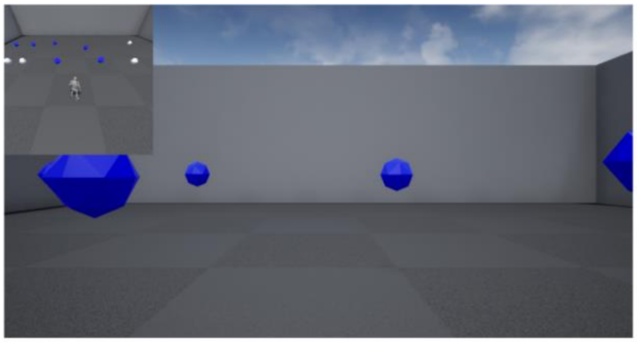


Figure 2: Task environment. The upper left window shows a view from a bird's-eye view camera, and the bottom window shows a view from the agent. The blue-colored objects are in the agent' s field of view. The white-colored objects are out of sight.

right. The task of the agent in this environment is to collect all the blue objects in as short a time as possible. However, with this agent, we did not aim to search for the shortest path connecting the positions of the objects. At each time point, the agent repeated a forward chaining search toward the nearest object.

Figure 3 is a flow chart showing the operation of the agent. Before collecting each object, the agent rotates its body and searches for objects in the environment. When the agent pays attention to one of the objects, it perceives the distance from it. When there are multiple objects in the field of view, one of the objects is selected according to the saliency values set for the object (Stewart & West, 2005). In the current agent, the saliency values were determined by the size of the object projected in the field of view, which corresponds to the distance from the agent. Based on the distance of the object to which its attention is directed, the agent updates the "nearest distance object" in the goal buffer.

At the blue triangle in Figure 3, the rule for searching for objects in the environment (*the searching rule* represented in the right-directed arrow from the triangle) conflicts with the rule for finishing the search (*the finishing rule* represented in the downward arrow from the triangle). Depending on the result of this choice, two types of errors might occur: incorrectly going to the non-nearest objects or continuing the search even after all objects were checked. In ACT-R, the frequency of these errors is controlled by conflict resolution. When the utility (priority) of the searching rule is higher than the utility of the finishing rule, the agent carefully checks the nearest object. Otherwise, the possibility of the other type of error (heading to the non-nearest object) is increased.

Experiment

We considered that one of the benefits of incorporating a cognitive modeling approach to HAI research is representing the individual difference between agents at a behavioral level.

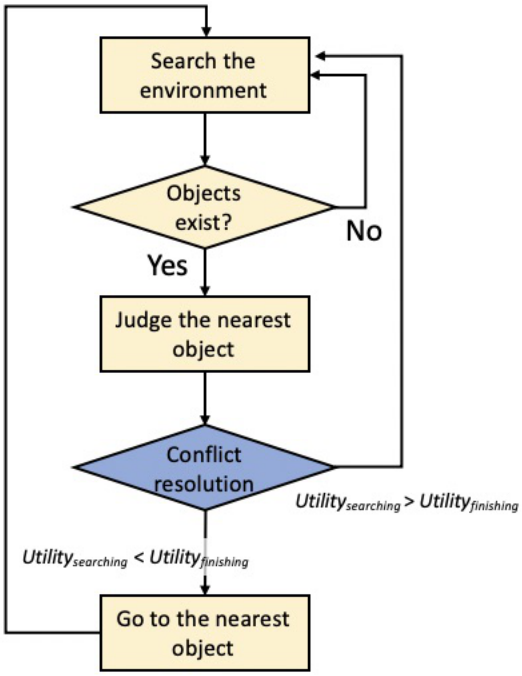


Figure 3: Flowchart of the environment search model.

Recently, in the cognitive modeling community, the exploration of model parameters that represent personal traits is a major topic (Rehling, Lovett, Lebiere, & an B. Demiral, 2004; Anderson, Bothell, Fincham, & Moon, 2016). Using parameters implemented in ACT-R, some researchers have also constructed models of atypical personal traits, such as depression (van Vugt & van der Velde, 2018) and autism (Morita et al., 2017). Utilizing these studies, it is possible to create various types of agent manipulating parameters that can be implemented in the model and architecture. In the case of our model, the agent that has a high utility value for the finishing rule can be regarded as *the reckless agent*, while the agent that has a high utility value for the searching rule can be regarded as *the careful agent*.

To demonstrate the difference between the behaviors of such agents, we conducted a simple experiment in which the utility values of the two rules in Figure 3 were varied. We prepared five conditions of 1:5, 2:4, 3:3, 4:2, and 5:1. The numbers on the left and right indicate the utility values of the searching rule and the finishing rule, respectively. In the simulation, transient noise ($s = 0.5$) was added to each utility value. The agent, whose walking speed was 450 cm/s, searched the environment presented in Figure 4 ten times for each condition. Figure 5 shows the completion time of each condition in box plots. From this figure, we can observe differences between the behaviors of each agent. Compared to the careful agents (the box plots toward the right), the reckless agents (the box plots toward the left) indicated better performance. However, we are not intending to conclude on the superiority of reckless decisions. There is a possibility that this

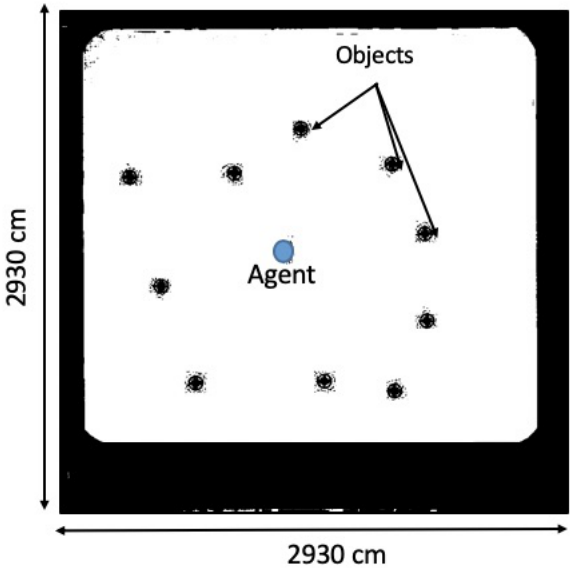


Figure 4: Arrangement of objects in the experiment.

result may change depending on the simulation settings (the map or the parameters of the agent such as walking speed). The key point is that by manipulating the parameters of the ACT-R model, we can easily represent a variability of the behavior in the 3D environment.

In addition to the above quantitative result, we found that the qualitative result indicating the potential of our architecture to replicate human-like behavior. In our architecture, the game engine and ACT-R regularly communicate via a blackboard server (Figure 1). During this process, the ACT-R model sometimes overlooked the update from the blackboard server due to mismatches between the communication rate and movement speed of the virtual agent. When such a communication error occurred, the current agent typically failed to be aware of finishing its own behavior keeping searching for the object even though it has already gotten (Figure 6). From an engineering point of view, such an error is regarded as a bug that should be fixed. However, in cognitive modeling or when building a human-like agent, we should evaluate such agent behaviors based on their correspondence to human behaviors. With regard to this error, we can find similar errors in the literature, in ecological psychology, called microslips in which an erroneous action is initiated but aborted (Reed & Schoenherr, 1992). The similarity between the human error pointed out in this psychological study and our agent shows that there is a certain validity in the structure of this architecture.

Conclusion

In this research, we constructed a mechanism to integrate cognitive modeling with the 3D virtual world. This was not the first time that it has been attempted to connect a game engine and ACT-R cognitive model. However, our architecture

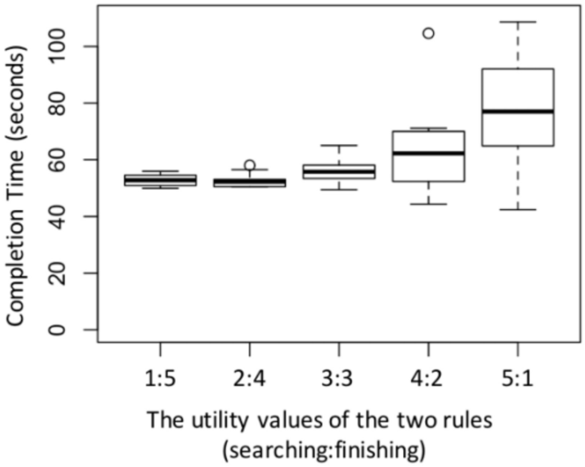


Figure 5: Results of the experiment.

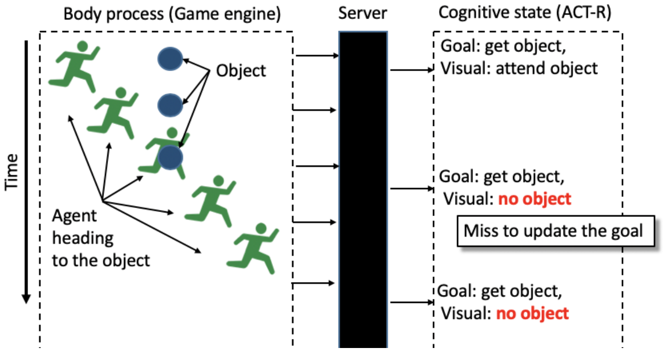


Figure 6: Schematic presentation of microslip error.

was different from the previous research in using a blackboard server (Figure 1) and not connecting the ACT-R and game engine with peer-to-peer. Due to this mechanism, novel agent behaviors, such as the microslip mentioned at the end of the previous section, emerged, and these were caused by ACT-R and the game engine operating in parallel.

Thus, the architecture constructed in this research may lead to the modeling of cognitive processes that have been overlooked in previous research. Many of the conventional cognitive models do not have a body and deal with the problems of a simple system closed in the brain. By giving a body in the virtual space to the cognitive model, there is a possibility of simulating important phenomena related to the interaction between the body and brain. In addition, the architecture of this research also has advantages in terms of being extended to a multi-agent environment. Considering this advantage, in the future, we plan to model interactions between groups with multiple embodied agents in the virtual world.

In addition, visualization of the virtual world using game engines has the advantage of making it possible to interact with agents operating using the ACT-R cognitive model and human users. The advantage of such interactive agent development with ACT-R is systematic diversion of research knowledge accumulated in cognitive modeling research. Furthermore, visualizing the behavior of interpersonal agents in the virtual world may also lead to a new methodology of validating hypotheses behind the implemented internal process in a cognitive model. In this way, the integrated approach that this research aimed for may lead to new HAI and cognitive modeling research methods.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Fincham, J. M., & Moon, J. (2016). The sequential structure of brain activation predicts skill. *Neuropsychologia*, *81*, 94-106.

Brooks, R. A. (1986). *Planning is just a way of avoiding figuring out what to do next* (Tech. Rep.). MIT Artificial Intelligence Laboratory.

Dancy, C. L., Ritter, F. E., Berry, K. A., & Klein, L. C. (2015). Using a cognitive architecture with a physiological substrate to represent effects of a psychological stressor on cognition. *Computational and Mathematical Organization Theory*, *21*(1), 90–114.

Harrison, A. (2002). jACT-R: Java ACT-R. In *Proceedings of the 8th Annual ACT-R Workshop.*

Minato, T., Shimada, M., Ishiguro, H., & Itakura, S. (2004). Development of an android robot for studying human-robot interaction. In B. Orchard, C. Yang, & M. Ali (Eds.), *Lecture Notes in Computer Science* (Vol. 3029, p. 424-434). Springer.

Morita, J., Konno, T., Okuda, J., Samejima, K., Li, G., Fujiwara, M., & Hashimoto, T. (2017). Implicit memory processing in the formation of a shared communication system. In *Proceedings of the 15th International Conference on Cognitive Modeling (ICCM 2017)* (p. 19-24).

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

Reed, E. S., & Schoenherr, D. (1992). *The neuropathology of everyday life: On the nature and significance of microslips in everyday artivities'.*

Reeves, B., & Nass, C. (1996). *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge University Press.

Rehling, J., Lovett, M., Lebiere, C., & an B. Demiral, L. M. R. (2004). Modeling complex tasks: An individual difference approach. In *In Proceedings of the 26th Annual Conference of the Cognitive Science Society* (p. 1137-1142).

Simon, H. (1996). *The Sciences of the Artificial* (Third edition ed.). Cambridge, MA: MIT Press.

Smart, P. R., Scutt, T., Sycara, K., & Shadbolt, M. R. (2016). Integrating ACT-R cognitive models with the unity game engine. In J. O. Turner, M. Nixon, U. Bernardet, & S. DiPaola (Eds.), *Integrating Cognitive Architectures into Virtual Character Design.* Hershey, Pennsylvania, USA: IGI Global.

Somers, S. (2016). ACT-R 3D: A 3d simulation environment for python act-r. In *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016)* (p. 107-112).

Stewart, T. C., & West, R. L. (2005). Python ACT-R: A new implementation and a new syntax. In *Proceedings of 12th Annual ACT-R Workshop.*

Trafton, J. G., Hiatt, L. M., Harrison, A. M., II, F. P. T., Khemlani, S. S., & Schultz, A. C. (2012). ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, *1*(1), 78-95.

van Vugt, M. K., & van der Velde, M. (2018). How does rumination impact cognition? a first mechanistic model. *Topics in Cognitive Science*, *10*(1), 175-191.

A Spiking Neural Model of Attention Effects in Memory

Marshall L. Mykietyshyn (mmykiety@edu.uwaterloo.ca)
Terrence C. Stewart (tcstewar@uwaterloo.ca)
Systems Design Engineering, University of Waterloo
200 University Avenue West, Waterloo, ON, N2L 3G1, Canada

Abstract

We present a spiking neuron model of attention-driven memory, where participants use a cue to indicate whether a word on a list is to be remembered or not. This model is fit to individual differences on mean behavioural data and produces a good match in terms of variance of performance on a recognition task, but not on a recall task. Neural activity patterns during the memorization parts of the task are also well-matched, but not during the time between when the attention cue is presented and when the word itself is presented. We believe this indicates mechanisms are involved in the recall task which are not considered as part of the current model.

Keywords: attention; memory; Neural Engineering Framework; LIF neurons

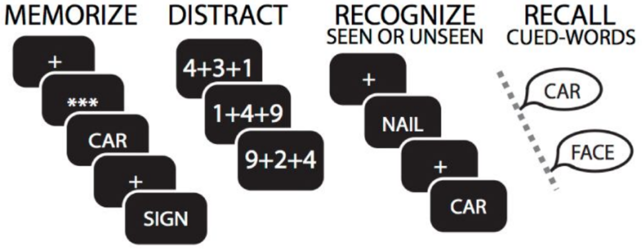


Figure 1: Task description, from (John H. Wittig et al., 2018).

Introduction

The overall goal of this work is to produce a neural-level explanation of psychological phenomena: in this case, the ability to control (via attention) what items on a word list are remembered, and which ones are not. In particular, we are interested in how low-level effects (such as how well neurons can represent, store, and transform information) can give explanations for how tasks are performed, and how people differ.

In an experiment to ”isolate the neural mechanisms of attention that lead to improved memory formation,” Wittig et al. (2018) visually presented words for one second, followed by a five second delay between words. Subjects were instructed to remember words that were preceded (cued) by a row of asterisks, with no instruction given regarding uncued words. Once the entire list was presented, subjects performed distraction tasks (simple arithmetic problems) for 20 seconds to suppress sub-vocal rehearsal and mitigate recency effects. Then the subjects memory of the word list was assessed using recognition and recall tests (John H. Wittig, Jang, Cocjin, Inati, & Zaghloul, 2018). This process is shown in Figure 1.

In the recognition (seen/unseen) test, subjects were shown a second list of words (test list) and asked to identify which words came from the list originally presented during the task (task list). The test list consisted of a mixture of cued words and uncued words from the task list, in addition to words that were not part of the task list (foil words). The recall test required subjects to verbally recite as many of the cued words as possible.

There were two recognition test criteria used to determine whether sessions would be used in the analysis 1) there must be a significant difference in recognition rates between cued and uncued words 2) there must be a significant difference in recognition rates between uncued and foil words. Significance was determined using a chi-squared contingency table (we assumed the criterion was $p<0.05$, although this is not stated in the paper). While training subjects to perform the task, the task difficulty was calibrated for each subject. Task difficulty was set primarily by altering the length of the task list such that subjects would meet both stated criteria. Other parameters (e.g. fraction of cued words) were altered if changing the list length was insufficient to meet the criteria; however, these secondary adjustments were not considered in the current work.

Adjusting the task difficulty allowed the testers to ”collect isoperformance data across participants who showed a wide range of natural aptitude for the task 1. Overall, 71/90 experimental sessions met both criteria, with training sessions excluded from the analysis. The distribution of list lengths is shown in Figure 2.

Representation

The goal of this work is to build a computational model of the memory aspects of this task using spiking Leaky Integrate-and-Fire (LIF) neurons. We use the Neural Engineering Framework (Eliasmith & Anderson, 2003) to find the connection weights between these neurons such that a) each group of neurons forms a distributed representation of a vector and b) each set of connections between groups of neurons approximates some desired function on those vectors. The neurons themselves have randomly chosen properties such as maximum firing rates, tuning curves, and preferred stimuli (a.k.a. encoders) to give a realistic heterogeneous distribution

In particular, if a population of neurons is to represent the vector $\mathbf{x}$, then each neuron i has an *encoder* $\mathbf{e}_i$ which is the value of x for which it most strongly fires. If the neuron has a randomly chosen gain α_i and bias β_i then the total current flowing into the neuron would ideally be $\alpha_i \mathbf{e}_i \cdot \mathbf{x} + \beta_i$. This will cause the population of neurons to have a different firing pattern for every value of $\mathbf{x}$.

If one group of neurons represents $\mathbf{x}$ and another represents $\mathbf{y}$ and we want $\mathbf{y}$ to be some function of $\mathbf{x}$ (i.e. $\mathbf{y} = f(\mathbf{x})$), then

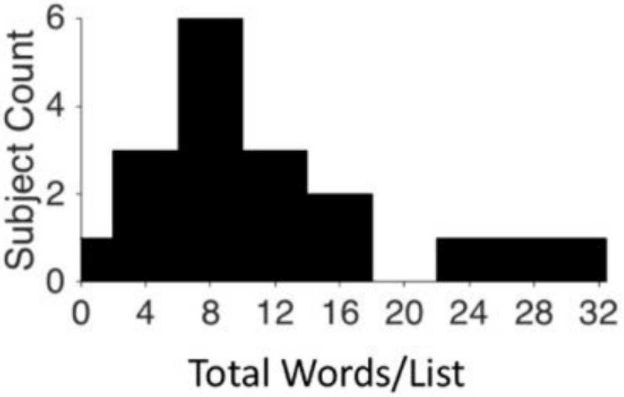


Figure 2: Distribution of list lengths among the 18 subjects, from (John H. Wittig et al., 2018).

we can form connections between the two groups of neurons. We solve for the weights ω_{ij} between neuron i in the first population and j in the second population such that the spiking activity a_i in the first population when representing x will lead to the corresponding current flowing into the second population to represent $f(x)$. In other words, we want $\sum_i a_i \omega_{ij} = \alpha_j \mathbf{e}_j \cdot f(\mathbf{x})$. Given this formulation, the connection weights ω_{ij} for any given function $f(\mathbf{x})$ can be found using least-squares minimization. We use the software package Nengo (Bekolay et al., 2014) to automate this process and run the resulting model.

While the above method can be used to create biologically plausible implementations of any computation, the neurons never perfectly approximate that computation. For example, if a group of neurons is connected back to itself with connection weights approximating the function $\mathbf{y} = \mathbf{x}$, then this would ideally be a perfect memory, storing the information $\mathbf{x}$ over time without change. However, the neural activity will, in practice, gradually change, leading to drift in the value $\mathbf{x}$ that is being represented. The purpose of this paper is to explore how that sort of low-level implementation detail affects the performance of an attention-driven memory system.

Because this is our goal, here we only present a model of the attention-driven memory. We do not include here a model of the visual recognition of cues and words, or the cognitive control needed to perform the task itself, as these aspects have been previously modelled using the Neural Engineering Framework (Eliasmith et al., 2012). We also do not model the decision-making process required to decide whether or not a particular word was in the remembered list. Instead, we directly decode out the vector $\mathbf{x}$ in the memory (also via least-squares minimization) and compare it to the vectors for different words using the dot-product to measure similarity. This was done for simplicity, and is equivalent to (but less noisy than) more detailed decision-making models that have previously been published (Sharma, Komer, Stewart, & Eliasmith, 2016; Hurzook, Trujillo, & Eliasmith, 2013).

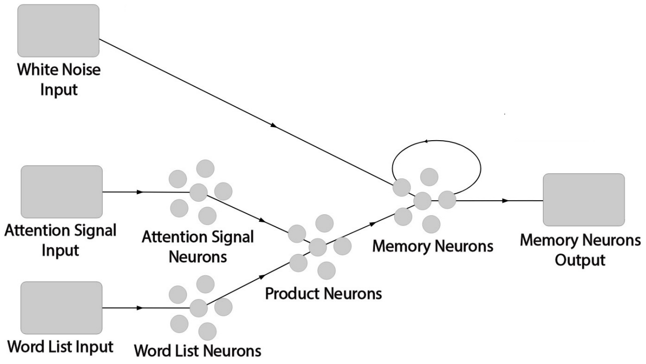


Figure 3: The spiking model of attention-driven memory.

Model

To model the memory aspects of this task, we use a group of 51,200 neurons representing a 512-dimensional vector (*Memory Neurons*). These neurons are recurrently connected to compute the function $\mathbf{y} = \mathbf{x}$. This means that, in the absence of any input, the neurons will maintain their firing pattern, creating a memory (mathematically, it will compute the integral of its input). The presented word is represented using another group of 51,200 neurons, representing a 512-dimensional vector (*Word List Neurons*). The effect of the cue is represented with an attention signal: 100 neurons representing a scalar value of how much attention to pay to the current word (*Attention Signal Neurons*. The 102,400 *Product Neurons* multiply the attention value by the word vector, sending the result into the memory.

To model the task, we present as input the (randomly chosen) vector for the current word, and a scalar value of how much attention to pay to that word. This will be larger for cued words than for uncued words, but the exact values are fit to account for individual differences between subjects. We apply varying amounts of white noise to the memory, also to account for individual differences. An illustration of this model can be seen in Figure 3.

The word list is represented by N randomly chosen 512-dimensional normalized vectors, where N is the length of the list. The attention signal is either high (cued words) or low (uncued words) for the one second the word is presented, and zero for the five second delay between words. When the attention signal is zero, the memory neurons are only affected by the white noise input and the inherent error involved in approximating a perfect memory using spiking neurons. For all simulations, exactly half the words in the list are cued, mirroring the original experiment conditions considered here.

The memory vector acts as an information compass, where the direction it points in the 512-dimension space indicates what information it represents. As the product of the inputs accumulates in memory, the memory vector gradually turns toward the direction of the word vector. How far the memory vector turns depends on the magnitude of the input (i.e. whether the attention signal is high or low), how long the word is presented for (one second in all cases), and how much information has previously been imprinted on the memory space. These effects can be seen in Figure 4. In all cases, we plot the dot product (i.e. the cosine similarity) of the vector decoded from memory and the ideal (randomly chosen) vector for each word.

The subjects memory at the time of testing is represented by the decoded memory vector at the end of the simulation. The 20 seconds of distraction tasks are represented by a 20 second period of zero input, during which no mechanism is applied to simulate rehearsal or other memory-enhancing effects. Although the recognition test was performed before the recall test in the experiment, for this work we did not consider how to model this interaction, which may be addressed in future work.

Since the model's memory vector represents the history of words imprinted on it, how strongly those words are held in memory can be calculated as the dot product of the original word vectors and the memory vector. Henceforth, we refer to this as cosine similarity. In the compass analogy, the magnitude of similarity indicates to what degree the memory and word vectors are pointing in the same direction. Cosine similarity can result in a value between 0 and 1, with 1 being perfect alignment and 0 being perfectly orthogonal. Since we are not yet building a full model of the decision-making process for extracting information out of memory, here we simply choose a threshold (different for each individual) for this similarity value.

Simulating the recognition and recall tests involved creating cosine similarity threshold values above which a word is represented strongly enough in memory to be considered ”recognized” (seen) or to be ”recalled”. These were taken as separate threshold values, as it was assumed that the mental process for recognizing a word is different than for recalling one. In order to establish threshold values that would reflect the experimental responses, experimental data was used to determine thresholds for each of the categories “seen cued”, “seen uncued”, “seen foil”, and “recalled”. Thresholds were calculated using the experimental correct response rates, which can be seen in Figure 5 below. For each category, the value above which the proper percentile of word cosine similarities lay was taken as the threshold. For example, in the experimental results, approximately 90% of cued words were correctly identified in the recognition test. Therefore, the “seen cued” threshold was calculated as the value above which 90% of cued words cosine similarities lay, see Figure 4. The calculations were performed over 20 sessions to reduce sample size error, and account for the volatility individual sessions.

Note that foil words are a separate randomly generated list of word vectors that are never presented to memory. They represent words that are not part of the task list, but are shown during the recognition tests. The rate of foil words recognized (seen) is the false-alarm rate of the test. Even though the words are never presented, and therefore are never imprinted

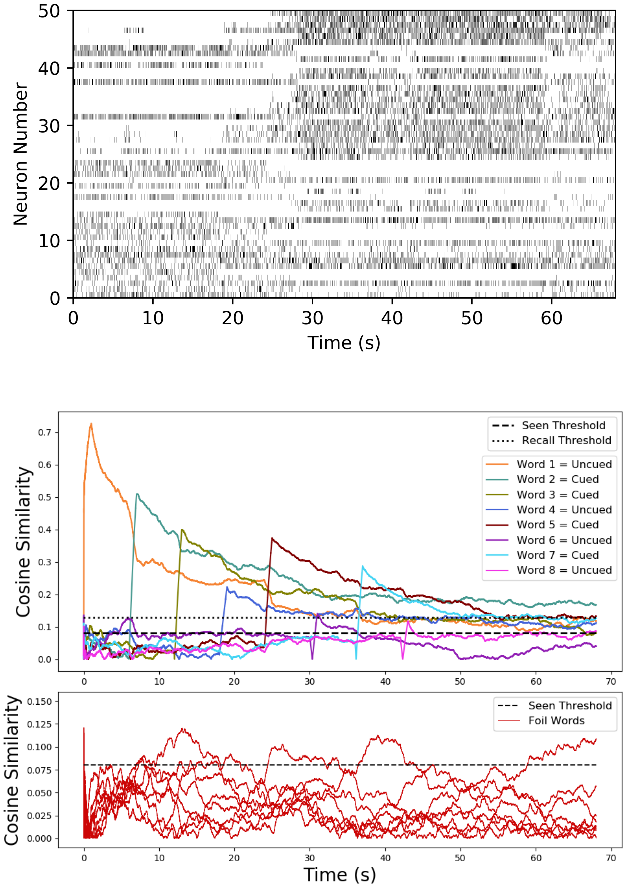


Figure 4: (Top) Spiking output from 50 randomly chosen Memory Neurons. This is the neural activity from which we decode the memory vector used for the other two parts of this figure. (Middle) Strength of task list words in memory, represented by the normalized cosine similarity of the memory vector with each studied word at each point in time. The calculated thresholds for the recognition and recall tests are shown for this subject (see below). (Bottom) Strength of foil words in memory. No foil words are imprinted on memory during simulation, so any similarity is simply due to the random foil word vectors existing in the same vector space as the memory vector. In this case, one foil word was incorrectly identified as being seen during the task. The ”Seen Word Threshold” has the same value in the upper and lower plots.

on memory, their cosine similarity values are non-zero, since they exist in the same 512-dimension vector space as memory. This allows us to calculate a false-alarm rate for the simulation, representing random chance, which is analogous to the foil word recognition rate in the experiment.

Since the recognition test involved three different categories, three separate recognition thresholds were calculated. However, in order for a word in memory to be recognized, it is irrelevant which category that word comes from. Therefore, model parameters were adjusted such that the recognition thresholds for the three seen categories were close enough that using their average would produce results similar to the individual thresholds. This allowed us to calculate a single recognition threshold representing all ”seen” categories. It was found that various combinations of parameters could produce approximately overlapping recognition thresholds for the three categories. This led to the creation of individual parameter profiles for each participant, examples of which can be seen in Table 1.

Tuning for Individual Differences in Task Difficulty

The original experiment used 18 subjects, whose corresponding word list lengths are shown in Figure 2. For the simulation, we created 18 subject profiles, tuned for list lengths matching the distribution of the original subjects. Tuning involved adjusting parameters such that the three “seen” category thresholds were similar in four out of five simulations. The parameters manipulated were the attention signal high/low values and the level of white noise added to memory. Once the subject profiles were determined, 30 simulations were run for each profile. The first 20 simulations were used to calculated recognition and recall cosine similarity thresholds, and the final 10 simulations were used as test sessions, upon which the analysis is performed.

White noise negatively affects the memory neurons ability to hold a value over time. Without the white noise, the “seen cued” and “seen uncued” thresholds were much higher than the “seen foil” threshold, which would have caused foil word recognition rates of near zero. Thus, adding this noise was necessary to accurately reflect the experimental responses. Another option for reducing the stability of the memory would have been to adjust the number of neurons. However, using white noise produced a much smoother effect on performance, making it easier to find parameter settings which matched particular subjects.

The experimental criteria described in the Task Description section were used to determine whether sessions were valid for analysis. Significance was calculated using a chi-squared contingency table, as described in the experimental methodology, using $p<0.05$.

It was found that subjects tuned for list lengths of four words resulted in a large majority of failed sessions (5/30 met the recognition test criteria). In a four word task list, two words are cued and two are uncued. Therefore, only two data points are available for calculating the ”seen cued” and ”seen uncued” thresholds. This made individual session thresholds for four word task lists quite volatile, with a large variance across the sessions used to calculate the average threshold. Consequently, the average thresholds for these subject profiles were not properly representative of the data, resulting in a high number of failed tests. Furthermore, the tests that did pass the statistical criteria, did not provide recognition rates in the expected ranges. Therefore, subject profiles with four-word task lists were removed, leaving 15 subject profiles for analysis. Additionally, Figure 2 shows only one subject with a task list length of zero. This subject was replaced with one where N = 16.

Table 1: Subject Profile Examples.

Tuned Parameter	Sub #1 (N = 8)	Sub #13 (N = 16)	Sub #15 (N = 24)
Cued (High) Attention	0.7	0.9	0.9
Uncued (Low) Attention	0.4	0.5	0.5
White Noise	0.01	0.008	0.003
Simulation Output			
”Seen Cued” Threshold	0.081	0.050	0.058
”Seen Uncued” Threshold	0.087	0.060	0.059
”Seen Foil” Threshold	0.073	0.074	0.071
Average Seen Threshold	0.080	0.061	0.063
Recall Threshold	0.128	0.106	0.106

Since the thresholds were based upon the experimental data, and the parameters were tuned such that the seen thresholds would overlap, there was some concern about overtuning the model. This is investigated in the Model Exploration section below.

Stages of Analysis

The model is analyzed in two independent stages. Any changes made to the model parameters affecting the first stage of analysis changes the simulation data used in the second stage of analysis. Therefore, second stage parameters were much easier to adjust and re-analyze than the first stage.

The first stage consists of creating the subject profiles and determining the associated recognition and recall thresholds, as described above. Once the thresholds are set for a particular subject profile, the first stage of analysis is complete. The second stage of analysis involves running test simulations. Using the thresholds from the first stage, the simulated recognition and recall test results are calculated and compared to the experimental data.

This means that the subject profiles parameters (attention signal values and white noise), and other model parameters (e.g. number of neurons) which would affect the simulation output (i.e. threshold values) cannot be altered in the second stage of analysis without repeating the first stage of analysis as well. This created time constraints on the number of parameters that could be investigated, as the first stage of analysis requires many hours of simulation for each subject profile.

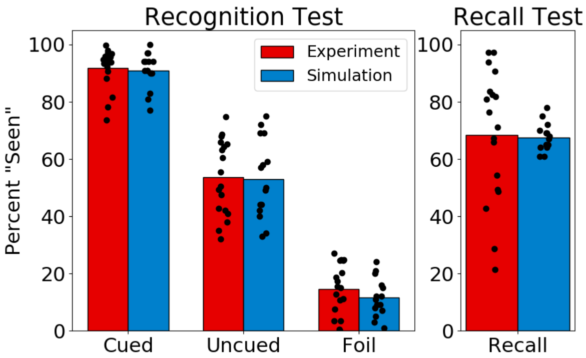


Figure 5: Comparison of experimental and simulated test results (John H. Wittig et al., 2018). In each case, only sessions that met both specified criteria were used (71/90 valid experimental sessions, 132/150 valid simulated sessions). Each data point represents the average recognition rate for an individual subject over 10 test sessions.

Table 2: Statistical Comparison of Test Data.

Category	Cued	Uncued	Foil	Recalled
Experimental Variance	50.7	174.3	79.7	534.3
Simulation Variance	39.4	190.9	44.4	23.3

Task Performance

The model's performance is compared to experiment in the rates of recognition and recall for each subject. Since the recognition and recall thresholds were generated using the experimental test data, it is expected that the simulated tests would produce similar average rates of recognition and recall. Therefore, we will focus instead on the variance of correct response rates for each of the two tests. These variances can be visually compared in Figure 5, and the calculated statistics are summarized in Table 2.

There is a large differences in variance between the experimental recognition and recall tests; however, the simulated results do not share this difference. Additionally, experimental and simulated sessions failed the performance criteria at approximately similar rates (Exp. = 79%, Sim. = 89%).

Figure 6 (Top) shows the activity of neurons measured by (John H. Wittig et al., 2018) during the experiment. It can be seen that the presence of a cue corresponds to a drop in activity immediately before presentation of the word. The red bar above the plot indicates where this effect is statistically significant. Figure 6 (Bottom) is the analogous plot showing neuron activity of our model. During the word presentation and the period after the word presentation, our model shows no difference in neural activity for cued versus uncued words. This is consistent with the empirical data. However, our model does not include the extra processing needed to *remember the cue*, and thus we do not expect to see a corresponding drop in activity in our model.

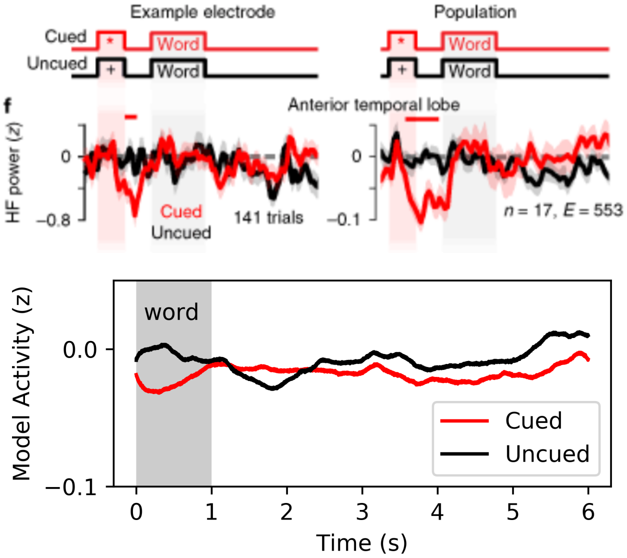


Figure 6: (TOP) Experimental neuron activity represented by high-frequency power electrode feedback, from (John H. Wittig et al., 2018). (Left) Single electrode. (Right) Population average across all electrodes. A red bar at the top of each plot indicates where there is a statistically significant difference between the cued and uncued responses. (BOTTOM) Average neuron activity from the model, for cued and uncued conditions. Note that the current model does not include the points in time between the cue and word presentation.

Model Exploration

As a test to check for extreme over-fitting in recreating the experimental correct response rates, we also ran the participant profiles to list lengths both longer and shorter than what they were tuned for. Longer lists should produce lower rates of correct responses for tests involving task list words, while shorter list lengths should produce higher rates of correct responses in the same tests. There should be no significant effect on the foil word rates as this represents random chance. The results of these modifications can be seen in Figure 7.

Discussion

In Figure 5 we see that the model behaves similarly to human subjects in the recognition test. We were able to choose a single cosine similarity threshold that produced results with significant differences in recognition rates between the cued, uncued and foil word categories, meeting the performance criteria. Table 2 illustrates that the variance in test performance is similar to experiment for the recognition test, but not for the recall test.

One possible explanation for the model's ability to match human behaviour well in the recognition test, but not the re-

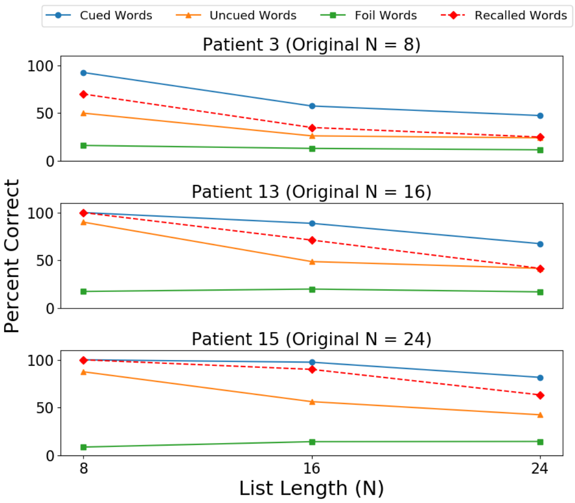


Figure 7: Effects of altering list length on task performance for three patients. The subject profile of patient 3 is calibrated for list lengths of N = 8, patient 13 for list lengths of N = 16, and patient 15 for list lengths of N = 24.

call test, is the way these tests were designed in the simulation. The simulated recognition test mirrors the experiment in that both involve presenting the subject with the word, then determining whether the subject recognizes that word. In the experiment, subjects are visually shown the word and must decide whether the word came from the original task list. In the simulation, the word vectors are compared to the final memory vector through cosine similarity. Conversely, in the experimental recall test subjects recall words from memory without a prompt, whereas in the simulated recall test cosine similarity is used again. This may indicate that using cosine similarity is a reasonable approximation of how the human brain performs the recognition test, but that humans use a different mechanism in performing the recall test.

The cosine similarity time plot (Figure 4) shows that the memory neurons of the model store information in an intuitive way. Words that are introduced to the memory space first are represented more strongly than those introduced later, regardless of whether or not they are cued. Therefore, the first word in a list will be imprinted most strongly on memory, since it is the only information stored in memory at that time. This effect is seen in Figure 4, where the first word is uncued, yet has the greatest similarity of all words when first introduced. After the fourth word of the list, the difference between high and low attention signals is more evident, as there are enough words affecting memory at this point that recency bias becomes less influential. This pattern reflects the way we would expect humans to remember words in a list.

Figure 7 illustrates that the model responds as expected when changing the difficulty of the task by altering the task list length for a subject profile. For cued and uncued words in the recognition test and cued words in the recall test, the simulation performance reduced as the task list length increased. This is the type of behaviour that we would expect from human subjects as the difficulty of the task, represented by task list length, increased. Performance for the foil words remained relatively flat, which is also as expected. This indicates that the fitting of the model parameters reasonably generalizes to other conditions, and produces predictions about individual performance on varying list lengths.

The simulation activity plot, Figure 6 (Bottom), does not show a difference between cued and uncued words. This is likely due to the design of the model. In the model, there is no analog for the cue, prior to the word being presented. The memory input simply represents a visual stimulus, multiplied by an attention value. This was done to keep the model simple, with the goal of comparing simulation and experimental results. In a further iteration of this model, a proper cue mechanism could be added examine the effects on model spiking activity.

Overall, we have presented a model showing a possible neural implementation of attention-driven memory. By taking the simple approach of representing words as vectors fed into a memory, and by scaling those vectors based on the amount of attention paid to them, we achieve a reasonable approximation of human behaviour in the recognition test, but the actual empirical data on the recall test has a much higher variance. Furthermore, we do not currently include the part of the task involving remembering the cue. Future work will address these concerns by including both the cue memory and the decision-making process to extract information out of the memory.

References

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., . . . Eliasmith, C. (2014). Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, *7*(48).

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. Cambridge, MA: MIT Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, *338*, 1202-1205.

Hurzook, A., Trujillo, O., & Eliasmith, C. (2013). Visual motion processing and perceptual decision making. In *35th annual conference of the cognitive science society* (pp. 2590–2595).

John H. Wittig, J., Jang, A. I., Cocjin, J. B., Inati, S. K., & Zaghloul, K. A. (2018). Attention improves memory by suppressing spiking-neuron activity in the human anterior temporal lobe. *Nature Neuroscience*, *21*, 808–810.

Sharma, S., Komer, B. J., Stewart, T. C., & Eliasmith, C. (2016). A neural model of context dependent decision making in the prefrontal cortex. In *38th annual meeting of the cognitive science society* (pp. 1122–1127). Austin, TX: Cognitive Science Society.

Modelling the Influence of Affect on Cognitive Processing using Chrest and Nengo

Dasaka Amarnath (amarnath.d@research.iiit.ac.in)
International Institute of Information Technology
Hyderabad – India

Bapi Raju Surampudi(raju.bapi@iiit.ac.in)
International Institute of Information Technology
Hyderabad – India

Abstract

It has long been understood that there is an interplay between affect and cognition((Kort & Reilly, 2003), but this interaction, based on the recent chess studies((Guntz, Crowley, Vaufreydaz, Balzarini, & Dessus, 2018), is much more intertwined than what the established theories postulate. To understand the underlying mechanisms in greater detail we propose an integrated model using Chrest and Nengo. We analyze the results based on simulations with data from previous empirical studies.

Keywords: Affect, Cognition , Chess, Nengo , Chrest

Introduction

The outline of our Paper is as follows: We start with Motivation for study followed by: Rational for using chess, Problem description, Research Objectives, Insights from previous research, State the Hypothesis, give a detailed view of proposed Architecture and finally implications for further research.

Motivation:

1. How do Chess players leverage emotions/Affect to deal with information overload and complexity?
2. How can interplay between emotions and cognition (information processing) be modelled?
3. How can interplay between emotions and information processing be modelled in two player chess game, where the behavioral signals from opponent serve as valuable cue? and behavioral cues from the player can serve as valuable information regarding the game play and therefore regulating these emotions (to suppress the display of behavior) have adverse effect on cognitive performance? (How can this scenario be quantified?)
4. How do chess players leverage emotions to Improve game play, complement the information processing capability and gain information about the situation based on the emotions of the opponent.?
5. What are the underlying mechanisms of this Cognitive Affective process and how can the components be modelled using available cognitive Frameworks/models?

Why Chess:

Researchers in computer science have famously referred to chess as the 'drosophila' of artificial intelligence (AI)(Lane & Gobet, 2012). What they seem to mean by this is that chess, like the common fruit fly, is an accessible, familiar, and relatively simple experimental technology that nonetheless can be used productively to produce valid knowledge about other, more complex system ((Ensmenger, 2012).

Chess is a very complex game. (Shannon showed a calculation for the lower bound of the game-tree complexity of chess, resulting in about 10120 possible games, to demonstrate the impracticality of solving chess by brute force, in his 1950 paper "Programming a Computer for Playing Chess". As a comparison, the number of atoms in the observable universe, to which it is often compared, is roughly estimated to be 1080 - Which is orders of magnitude lower) (Claude Shannon , 1950)

Problem Description:

Despite the enormous complexity, Chess players perform very well, in addition to making accurate moves under time constraints (Gobet, 2005)

Insights from Previous Research:

Even though, Chunking and template theory explain underlying mechanisms about how human mind can overcome the limitations imposed by working memory. Chunking and template theory is implemented in Chrest Framework. This framework has been used to validate the results of various chess observations. Similarly, modules for attention are also part of this framework.(Gobet, Lane, & Lloyd-Kelly, 2015)

Recent studies have highlighted the influence of emotions in dealing with this complexity(Guntz et al., 2018). The studies indicate that emotions are more tightly coupled with the information processing capability. Chess players associate previous game situations to specific emotions and use the emotions in narrowing down the potential candidate moves which are subsequently evaluated. But These studies were conducted in lab, and so far, there is no theoretical model to explain this observation.

Research Objective:

We propose to model the influence of emotions on chess playing skill using Chrest and Nengo.

We Leverage the modules already implemented in Chrest (Gobet & Jansen, 2004) for chess capabilities and leverage nengo for modulating the emotion based on the game situation. (Chrest and Nengo(Bekolay et al., 2014) work in tandem simulating a real-player scenario)

Hypothesis:

H1a: Emotions, rather than just cognitive abilities, will influence cognitive processing in chess task

H2a: Emotions, complement and assist cognitive abilities, in chess task (They are integral to Cognitive abilities)

Proposed Architecture

We have built on the existing NTIM Framework ((Degroot & Broekens, 2003). The current Architecture is limited to manipulating behavior but does not integrate emotion to complement the cognitive task. We use CHREST Framework to model the attention, heuristic search and pattern recognition, and the input from Nengo which is based on the situational parameters given (Personality Model) is used to manipulate the emotion Valance and Arousal.

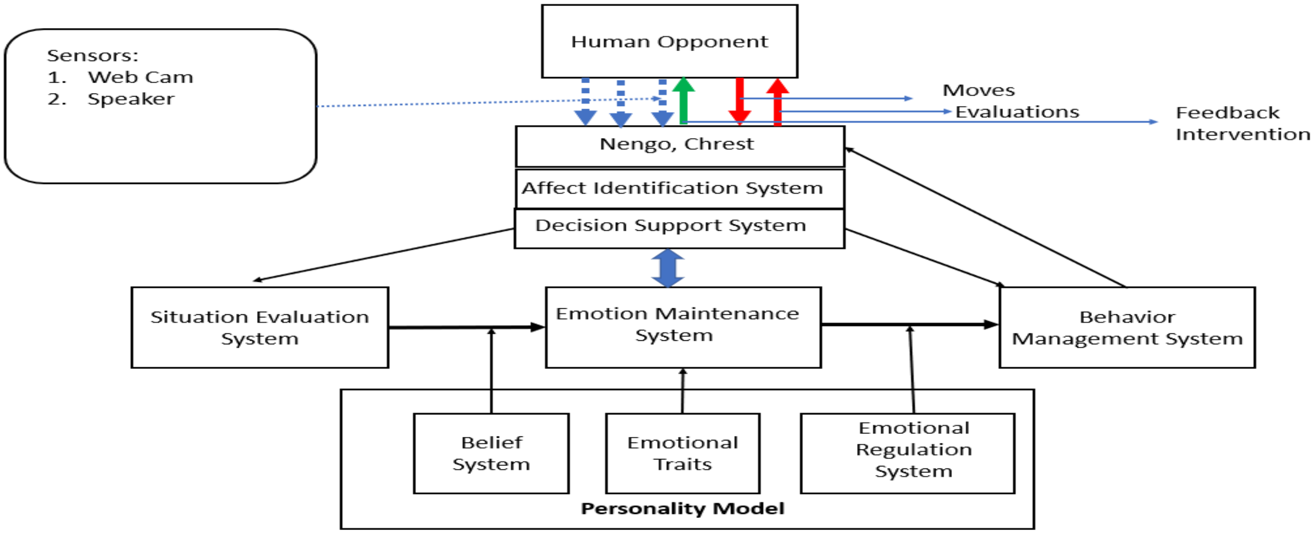


Figure 1 Proposed Architecture to integrate Affect and Cognition

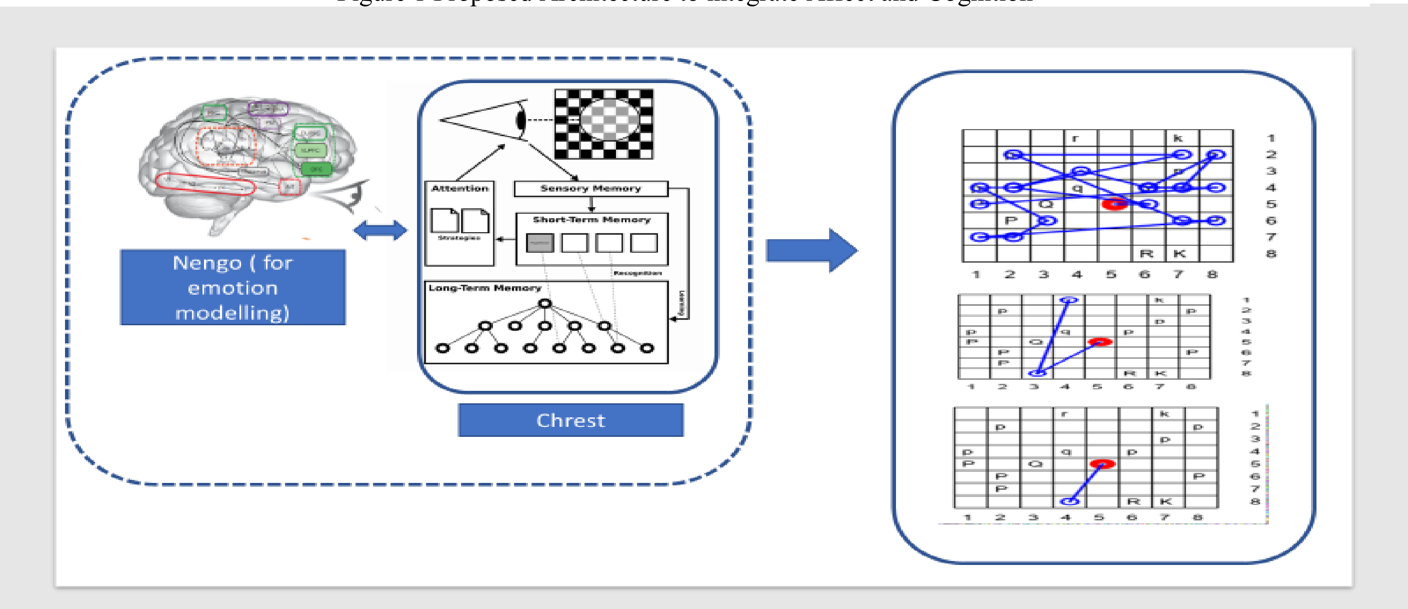


Figure 2 Integration of Nengo and Chrest and output Results (Gaze patterns)

Results

S.NO	User Profile	Chess-Score (ELO) - DV	Fixation Time, Gaze time, Fixation Duration (Seconds)
1	Cognitive (Analytical)	1900	16.1,12,5.4
2	Cognitive - Affective	2100	18.1,12,5.3.5
3	Affective (Intuition)	1700	7.3,5,6

Future Directions

The Framework can be used to model the users profile based on the affective and cognitive observations and has practical implications in developing cognitive affective learning agents.

References

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., … Eliasmith, C. (2014). Nengo: a Python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, *7*, 48. https://doi.org/10.3389/fninf.2013.00048

Degroot, D., & Broekens, J. (2003). Using Negative Emotions to Impair Game Play 2 . A Shift to Negative Emotional Behaviors. *Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2003, Nijmegen, The Netherlands)*.

Ensmenger, N. (2012). Is chess the drosophila of artificial intelligence? A social history of an algorithm. *Social Studies of Science*, *42*(1), 5–30. https://doi.org/10.1177/0306312711424596

Gobet, F. (2005). Chunking models of expertise: implications for education. *Applied Cognitive Psychology*, *19*(2), 183–204. https://doi.org/10.1002/acp.1110

Gobet, F., & Jansen, P. (2004). Training in chess : A scientific approach. *Education and Chess*, *44*(115), 1–24. Retrieved from http://chrest.info/fg/preprints/Training_in_chess.PDF

Gobet, F., Lane, P. C. R., & Lloyd-Kelly, M. (2015). Chunks, Schemata, and Retrieval Structures: Past and Current Computational Models. *Frontiers in Psychology*, *6*, 1785. https://doi.org/10.3389/fpsyg.2015.01785

Guntz, T., Crowley, J., Vaufreydaz, D., Balzarini, R., & Dessus, P. (2018). The Role of Emotion in Problem Solving: First Results from Observing Chess. *20th ACM International Conference on Multimodal Interaction*, 1–13. https://doi.org/ffhal-01886694f

Kort, B., & Reilly, R. (2003). Analytical Models of Emotions, Learning and Their Relationships. *Proceedings of the International Conference on Virtual Worlds and Simulations*. Retrieved from https://affect.media.mit.edu/projectpages/lc/vworlds.pdf

Lane, P., & Gobet, F. (2012). Research and Development in Intelligent Systems XXIX. *Research and Development in Intelligent Systems XXIX*, (June 2016). https://doi.org/10.1007/978-1-4471-4739-8

Decoy Effect and Violation of Betweenness in Risky Decision Making: A Resource-Rational Mechanistic Account

Ardavan S. Nobandegani[†,1,3], **Kevin da Silva Castanheira**[†,3], **A. Ross Otto**[3], **& Thomas R. Shultz**[2,3]
{ardavan.salehinobandegani, kevin.dasilvacastanheira}@mail.mcgill.ca
{ross.otto, thomas.shultz}@mcgill.ca
[1]Department of Electrical & Computer Engineering, McGill University
[2]School of Computer Science, McGill University
[3]Department of Psychology, McGill University
[†]**Co-primary authors**

Abstract

A wealth of experimental evidence shows that, contrary to normative models of choice, people's preferences are markedly swayed by the context in which options are presented. In this work, we present the first resource-rational, mechanistic account of the decoy effect—a major contextual effect in risky decision making. Our model additionally explains a related, well-known behavioral departure from expected utility theory: violation of betweenness. We demonstrate that, contrary to widely held views, these effects can be accounted for by a variant of normative expected-utility-maximization, which acknowledges cognitive limitations. Our work is consistent with two empirically well-supported hypotheses: (i) In probabilistic reasoning and judgment, a cognitive system accumulates information through sampling, and (ii) People engage in pairwise comparisons when choosing between multiple alternatives.

Keywords: Risky decision-making; decoy effect; violation of betweenness; rational process models; expected utility theory

1 Introduction

Expected utility theory (EU), the most prominent model of rational choice (Bernoulli, 1738/2011), maintains that people's preferences should not change depending on the context in which options are presented. More specifically, according to the von Neumann-Morgenstern axiomatization of EU (von Neumann & Morgenstern, 1947/2007), a rational decision-maker obeys the independence axiom: preferences between alternatives A and B depend only on preferences between A and B. Formally, the independence axiom prescribes the following: If A is preferred to B out of the choice set $\{A,B\}$, introducing a third option X, expanding the choice set to $\{A,B,X\}$, does not make B preferable to A.

Contrary to the independence axiom, however, a wealth of experimental evidence shows that people's preferences are markedly swayed by the context in which options are presented (e.g., Huber, Payne, & Puto, 1982; Wedell, 1991; Roe, Busemeyer, & Townsend, 2001; Soltani, De Martino, & Camerer, 2012; Tsetsos, Chater, & Usher, 2012; Noguchi & Stewart, 2014; Mohr, Heekeren, Rieskamp, 2017).

Although contextual effects are predominantly studied in the realm of multi-attribute decision making without risk (e.g., Roe et al., 2001; Noguchi & Stewart, 2014), several studies have experimentally investigated contextual effects in risky decision-making (Huber et al., 1982; Wedell, 1991; Soltani et al., 2012; Tsetsos et al., 2012; Mohr et al., 2017).

A prominent contextual effect in risky choice is the decoy effect (e.g., Mohr et al., 2017) according to which the inclusion of a third asymmetrically-dominated gamble (decoy) into the choice set leads to increased preference for the dominating gamble (target), thus clearly violating the independence axiom of EU.

In this work, we present the first resource-rational, mechanistic account of the decoy effect in risky decision-making. Concretely, we show that, contrary to widely held views, this effect can be accounted for by a variant of the normative maximizing of expected utility, *sample-based expected utility* (SbEU), which acknowledges cognitive limitations that a decision-maker is faced with (Nobandegani, da Silva Castanheira, Otto, & Shultz, 2018).

SbEU is a metacognitively-rational, process model that takes into account that people adapt their strategies depending on the amount of time available for decision-making (e.g., Maule & Svenson, 1993; Svenson, 1993). Consistent with a large body of evidence, SbEU posits that, in probabilistic reasoning and judgment, a cognitive system accumulates information through *sampling* (e.g., Vul et al., 2014; Battaglia et al., 2013; Lake et al., 2017; Gershman, Horvitz, & Tenenbaum, 2015; Hertwig & Pleskac, 2010; Griffiths et al., 2012; Gershman, Vul, & Tenenbaum, 2012; Bonawitz et al., 2014).

Additionally, our mechanistic explanation of the decoy effect relies on a key assumption: people engage in pairwise comparisons when choosing between multiple alternatives. Recent experimental work has provided mounting evidence for this assumption (e.g., Russo & Rosen, 1975; Noguchi & Stewart, 2014). Specifically, recent eye-tracking work by Noguchi and Stewart (2014) shows that, when choosing between multiple alternatives, a series of comparisons is made in each choice, with a pair of alternatives compared on a single attribute dimension in each comparison.

We furthermore show that our resource-rational, process-level account of the decoy effect can also explain another related, well-known behavioral departure from EU: violation of betweenness (e.g., Camerer & Ho, 1994; Prelec, 1990). Concretely, betweenness is a weakened form of the independence axiom, prescribing that a probability mixture of two risky gambles should lie between them in preference (Camerer & Ho, 1994). Despite being widely assumed in game theory, auction theory, macroeconomics, and dynamic choice, violations of betweenness are experimentally well-documented (e.g., Camerer & Ho, 1994; Prelec, 1990).

After presenting a brief overview of SbEU, we proceed to model the decoy effect and violation of betweenness in risky decision-making.

2 Sample-based Expected Utility Model

Extending the cognitively-rational decision-making model of Lieder, Griffiths, and Hsu (2018) to the realm of metacognition (Roberts & Erdos, 1993; Cary & Reder, 2002), SbEU is a metacognitively-rational process model of risky choice, positing that an agent rationally adapts their strategies depending on the amount of time available for decision-making (Nobandegani et al., 2018). Concretely, SbEU assumes that an agent estimates expected utility

$$\mathbb{E}[u(o)] = \int p(o)u(o)do, \tag{1}$$

using self-normalized importance sampling (Hammersley & Handscomb, 1964; Geweke, 1989), with its importance distribution q^* aiming to minimize mean-squared error (MSE):

$$\hat{E} = \frac{1}{\sum_{j=1}^{s} w_j} \sum_{i=1}^{s} w_i u(o_i), \quad \forall i: \; o_i \sim q^*, \; w_i = \frac{p(o_i)}{q^*(o_i)}, \tag{2}$$

$$q^*(o) \propto p(o)|u(o)| \sqrt{\frac{1+|u(o)|\sqrt{s}}{|u(o)|\sqrt{s}}}. \tag{3}$$

MSE is a standard normative measure of the quality of an estimator, and is widely adopted in machine learning and mathematical statistics (Poor, 2013). In Eqs. (1-3), o denotes an outcome of a risky gamble, $p(o)$ the objective probability of outcome o, $u(o)$ the subjective utility of outcome o, $\hat{E}$ the importance-sampling estimate of expected utility given in Eq. (1), q^* the importance-sampling distribution, o_i an outcome randomly sampled from q^*, and s the number of samples drawn from q^*.

While cognitively-rational agents are ignorant about adapting their importance distribution q based on time availability, a *meta*cognitively-rational agent would plausibly use such considerations in their choice of q. That is, the metacognitively-rational agent chooses a q which is normatively-justified based on time availability considerations, allowing strategy selection to be guided by time availability. In agreement with this view, a large body of psychological work on decision-making suggests that people adapt their strategies in accord with time availability (e.g., Maule & Svenson, 1993; Svenson, 1993). As evidenced by Eq. 3 explicitly depending on s, SbEU assumes that decision-makers rationally adapt their strategies depending on time availability.

SbEU posits that, when choosing between a pair of risky gambles $\{A, B\}$, people make their choice depending on whether the expected value of the utility difference $\Delta u(o)$ is negative or positive (w.p. stands for with probability):

$$A = \begin{cases} o_A & \text{w.p. } p_A \\ 0 & \text{w.p. } 1-p_A \end{cases} \tag{4}$$

$$B = \begin{cases} o_B & \text{w.p. } p_B \\ 0 & \text{w.p. } 1-p_B \end{cases} \tag{5}$$

$$\Delta u(o) = \begin{cases} u(o_A) - u(o_B) & \text{w.p. } p_A p_B \\ u(o_A) - u(0) & \text{w.p. } p_A(1-p_B) \\ u(0) - u(o_B) & \text{w.p. } (1-p_A)p_B \\ 0 & \text{w.p. } (1-p_A)(1-p_B) \end{cases} \tag{6}$$

Recent work by Nobandegani et al. (2018) showed that SbEU can account for availability bias, people's tendency to overestimate the probability of events that easily come to mind (Tversky & Kahneman, 1973), and can accurately simulate the well-known fourfold pattern of risk preferences in outcome probability (Tversky & Kahneman, 1992) and in outcome magnitude (Markovitz, 1952; Hershey & Schoemaker, 1980; Scholten & Read, 2014). Notably, SbEU is the first rational process model to score near-perfectly in optimality, economical use of limited cognitive resources, and robustness, simultaneously (Nobandegani et al., 2018; Nobandegani, da Silva Castanheira, O'Donnell, & Shultz, 2019).

Following Nobandegani et al. (2018), and consistent with prospect theory (Kahneman & Tversky, 1979) and cumulative prospect theory (Kahneman & Tversky, 1992), in this work we assume the following S-shaped, utility function:

$$u(x) = \begin{cases} x^{0.85} & \text{if } x \geq 0, \\ -|x|^{0.95} & \text{if } x < 0. \end{cases} \tag{7}$$

3 Decoy Effect in Risky Decision-Making

A prominent contextual effect in risky choice is the decoy effect (DE) according to which the inclusion of a third asymmetrically-dominated gamble (decoy D) into the choice set $\{T, C\}$ (comprising of target T and competitor C) leads to increased preference for the dominating gamble (target T), thus violating the independence axiom of EU (Arrow, 1963; Ray, 1973; Machina, 1987).

Formally, DE can be mathematically characterize by having $\mathbb{P}(T|\{T,C,D\}) > \mathbb{P}(T|\{T,C\})$ (Huber et al., 1982; Simonson, 1989; Speekenbrink & Shanks, 2013; Mohr et al., 2017), with $\mathbb{P}(T|\{T,C,D\})$ and $\mathbb{P}(T|\{T,C\})$ denoting the probability of choosing T when the choice set is $\{T,C\}$ and $\{T,C,D\}$, respectively.[1]

Consistent with mounting experimental evidence (e.g., Russo & Rosen, 1975; Noguchi & Stewart, 2014), we assume that the decision-maker engages in pairwise comparisons when choosing from the choice set $\{T,C,D\}$, with parameters p_{tc}, p_{cd}, p_{td} denoting the probability of starting with the pairs $(T,C), (C,D), (T,D)$, respectively. The winner of the first pairwise comparison will then compete against the remaining risky gamble. Ultimately, the winner of the final

[1] DE is also a violation of the regularity axiom—a weakened form of the independence axiom—according to which the addition of an option to the choice set can never increase the probability of choosing an option relative to the original set (Speekenbrink & Shanks, 2013). More formally, for options X,Y,Z, the regularity principle prescribes the following: $\mathbb{P}(X|\{X,Y\}) > \mathbb{P}(X|\{X,Y,Z\})$.

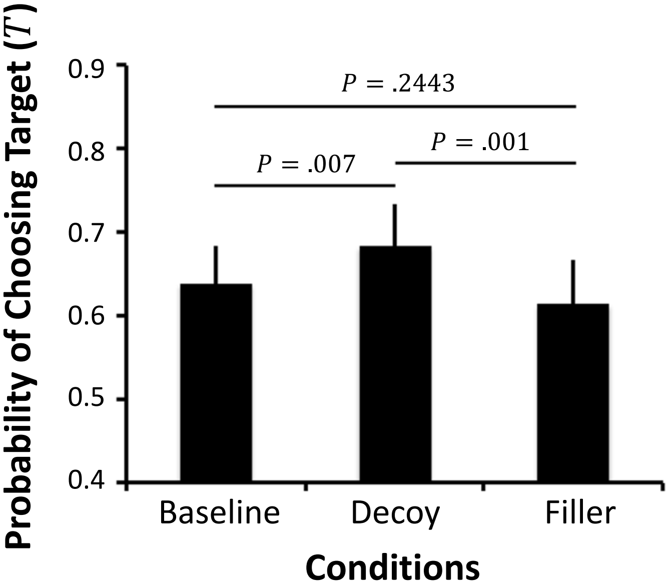


Figure 1: Mohr et al.'s (2017) experimental data. Once decoy (D) is added to the choice set (middle bar, Decoy Condition), people's preference for the target (T) significantly increases. People's preference for the target in the Decoy Condition is also significantly higher than it is in the Filler Condition. However, people's preference for the target is not significantly different between the Baseline Condition (wherein the choice set is $\{C,T\}$) and the Filler Condition (wherein the choice set is $\{C,T,F\}$). Adapted from Mohr et al.'s (2017, Fig. 2A).

pairwise comparison will determine the explicit choice that the agent makes from the original choice set $\{T,C,D\}$.

Recently, Mohr et al. (2017) investigated the neural underpinnings of DE in risky decision making using fMRI. Concretely, Mohr et al. (2017) showed that specific brain regions (e.g., the medial orbitofrontal cortex and the anterior insula) not only code the value or risk of a single choice option but also code the evidence in favor of the best option compared with other available choice options.

In their behavioral experiment, Mohr et al. (2017) showed that $\mathbb{P}(T|\{T,C,D\}) > \mathbb{P}(T|\{T,C\})$ when D is asymmetrically dominated (i.e., D is dominated by T but not C), while $\mathbb{P}(T|\{T,C\}) \approx \mathbb{P}(T|\{T,C,F\})$ when a gamble F (called filler) is dominated by both T and D, thus experimentally confirming DE in risky decision-making. Mohr et al.'s (2017) experimental data are shown in Fig. 1.

Next, we show that SbEU, together with the experimentally well-supported assumption of pairwise comparison, can provide a resource-rational mechanistic explanation of the behavioral finding by Mohr et al. (2017) discussed above. For our simulation of risky DE, we adopt a representative stimulus from Mohr et al. (2017, Fig. 1), involving four gambles

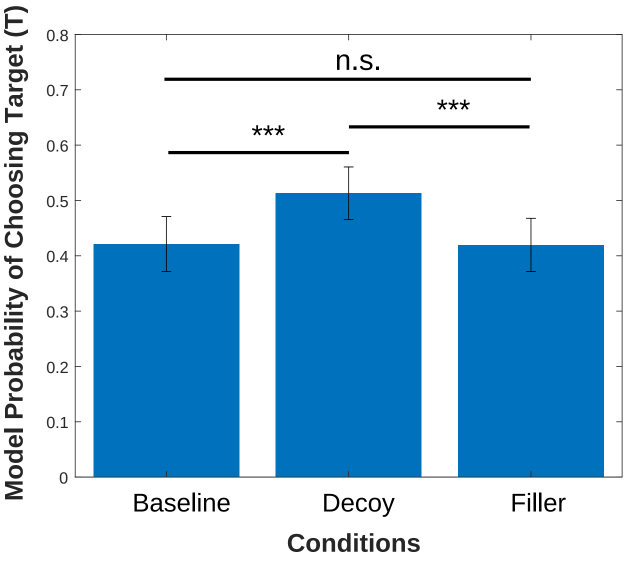


Figure 2: Model simulation of Mohr et al.'s (2017) experimental data reported in Fig. 1. Error bars indicate standard deviation (SD). $^{***}P < .001$, n.s. not significant.

(€ denotes the Euro sign):

$$C = \begin{cases} 80€ & \text{w.p. } 20\% \\ 0 & \text{w.p. } 80\% \end{cases}$$

$$T = \begin{cases} 20€ & \text{w.p. } 80\% \\ 0 & \text{w.p. } 20\% \end{cases}$$

$$D = \begin{cases} 20€ & \text{w.p. } 70\% \\ 0 & \text{w.p. } 30\% \end{cases}$$

$$F = \begin{cases} 20€ & \text{w.p. } 20\% \\ 0 & \text{w.p. } 80\% \end{cases}$$

where C,T,D,F denote the competitor, target, decoy, and filler gambles, respectively.

A la Mohr et al. (2017), we consider three experimental conditions, with the Baseline Condition, Decoy Condition and Filler Condition corresponding to participants choosing from the choice sets $\{T,C\},\{T,C,D\},\{T,C,F\}$, respectively.

We simulate $N = 1000$ participants, with every participant performing 100 trials of each of the experimental conditions. Model predictions for a few samples ($s = 4$) are shown in Fig. 2. This choice of s is supported by recent work providing mounting evidence that people often use only a few samples in probabilistic judgments and reasoning under uncertainty (e.g., Vul et al., 2014; Battaglia et al. 2013; Lake et al., 2017; Gershman, Horvitz, & Tenenbaum, 2015; Hertwig & Pleskac, 2010; Griffiths et al., 2012; Gershman, Vul, & Tenenbaum, 2012; Bonawitz et al., 2014; Nobandegani et al., 2018).

Fully consistent with Mohr et al.'s (2017) experimental results (see Fig.1), SbEU predicts that $\mathbb{P}(T|\{T,C,D\}) > \mathbb{P}(T|\{T,C\})$ ($t(999) = 42.2177$, $P < .001$, Cohen's $d = 1.3350$) and $\mathbb{P}(T|\{T,C,D\}) > \mathbb{P}(T|\{T,C,F\})$ ($t(999) =$

43.9820, $P < .001$, Cohen's $d = 1.3908$), while predicting that $\mathbb{P}(T|\{T,C\}) \approx \mathbb{P}(T|\{T,C,F\})$ ($t(999) = 0.7550$, $P = 0.4504$, Cohen's $d = 0.0239$). Model predictions are shown in Fig. 2.

In Fig. 2, we set $p_{tc} = 0.01, p_{cd} = 0.98, p_{td} = 0.01$. Recall that the parameters p_{tc}, p_{cd}, p_{td} denote the probability of starting with the pairs $(T,C), (C,D), (T,D)$, respectively. The relatively high value of p_{cd} receives strong theoretical supports from Theorem 1. Specifically, Theorem 1 provides a general, theoretical foundation for risky DE, under the experimentally well-supported hypothesis that people engage in pairwise comparisons when choosing between multiple alternatives (the pairwise-comparison hypothesis).

Theorem 1. *Let $S = \{T,C,D\}$ be the choice set, with T,C,D denoting the target, competitor, and decoy, respectively. Assuming that a decision-maker is to always start with a particular pair, then the following holds true: Starting only with the pair (C,D) can potentially produce risky DE. That is, starting with the pair (C,D) can potentially lead to having $\mathbb{P}(T|\{T,C,D\}) > \mathbb{P}(T|\{T,C\})$, while starting with the pair (T,C) or (T,D) grants $\mathbb{P}(T|\{T,C,D\}) \not> \mathbb{P}(T|\{T,C\})$.*

Proof of Theorem 1 is given in the Appendix. Theorem 1 has an important implication which can be articulated in simple terms as follows: Assuming that people perform pairwise comparisons when choosing between multiple alternatives, the only reliable way of producing risky DE is for people to significantly direct their attention to the pair (C,D) at the outset of their decision-making process. (Recall that the amount of attention directed at the pair (C,D) at the outset of decision-making is controlled by the parameter p_{cd}.) This provides a mathematically-rigorous, formal basis for our choice of $p_{tc} = 0.01, p_{cd} = .98, p_{td} = 0.01$.

4 Violation of Betweenness in Risky Choice

Betweenness is a relaxation of the independence axiom, prescribing that a probability mixture of two risky gambles should lie between them in preference (Camerer & Ho, 1994). Despite being widely assumed in game theory, auction theory, macroeconomics, and dynamic choice, experimental violations of betweenness are well-documented (e.g., Camerer & Ho, 1994; Prelec, 1990).

Formally, betweenness can be characterized as follows (Camerer & Ho, 1994): If risky gamble A is preferred to risky gamble B (i.e., $A \succ B$), then the following should hold: $\forall p \in (0,1): \;\; A \succ pA + (1-p)B \succ B$, where $pA + (1-p)B$ denotes a probabilistic mixture of A and B with probabilities p and $1-p$, respectively. In simple terms, betweenness requires that every probabilistic mixture of two gambles A and B lie between them in preference (hence the term "betweenness").

Next, we show that SbEU can additionally account for an experimentally-documented violation of betweenness (Prelec, 1990; Camerer & Ho, 1994).

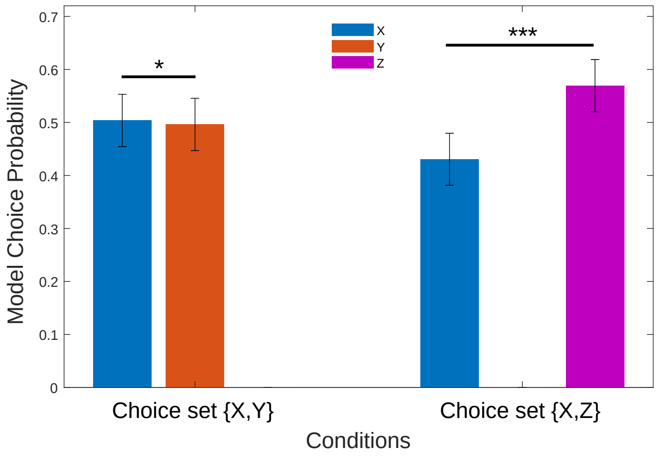


Figure 3: Model simulation of Prelec's (1990) experimental results. Consistent with the experimental data, the model predicts that, when choosing from the choice set $\{X,Y\}$, people prefer X over Y. However, when choosing from the choice set $\{X,Z\}$, the model predicts that people prefer Z over X, thus violating the betweenness property. Error bars indicate standard deviations.$^*P < .05$, $^{***}P < .001$.

An experiment by Prelec (1990), and replicated by Camerer and Ho (1994), revealed that people preferred X to Y, but preferred Z (a probabilistic mixture of X and Y) to X, hence violating the betweenness property. The gambles X,Y,Z are given by (Prelec, 1990):

$$X = \begin{cases} \$20{,}000 & \text{w.p. } 34\% \\ 0 & \text{w.p. } 66\% \end{cases} \quad Y = \begin{cases} \$30{,}000 & \text{w.p. } 17\% \\ 0 & \text{w.p. } 83\% \end{cases}$$

$$Z = \begin{cases} \$30{,}000 & \text{w.p. } 1\% \\ \$20{,}000 & \text{w.p. } 32\% \\ 0 & \text{w.p. } 67\% \end{cases}$$

where $Z = \frac{16}{17}X + \frac{1}{17}Y$, i.e., Z is a probabilistic mixture of the gambles X and Y with probabilities $\frac{16}{17}$ and $\frac{1}{17}$, respectively.

As we did in our simulation of risky DE, we simulate $N = 1000$ participants, with each participant performing 100 trials of each of the experimental conditions (Condition I: choosing between the gamble X and Y; Condition II: choosing between the gambles X and Z).

Model predictions for a few samples ($s = 4$) are shown in Fig. 3. Consistent with the Prelec's (1990) experimental data, SbEU predicts that, in Condition I, $\mathbb{P}(X|\{X,Y\}) > \mathbb{P}(Y|\{X,Y\})$ ($t(999) = 2.4160$, $P = .0159$, Cohen's $d = 0.0764$) while predicting that, in Condition II, $\mathbb{P}(Z|\{X,Z\}) > \mathbb{P}(X|\{X,Z\})$ ($t(999) = 44.8260$, $P < .001$, Cohen's $d = 1.4175$).

5 General Discussion

Expected utility theory (EU), the most prominent model of rational choice, maintains that people's preferences should not change depending on the context in which options are presented. Contrary to a widely held view, in this work we show that a variant of normative expected-utility-maximization which acknowledges cognitive limitations, SbEU (Nobandegani et al., 2018), can provide a metacognitively-rational process-level account of a prominent contextual effect in risky decision-making: the decoy effect (e.g., Mohr et al., 2017). Additionally, our explanation provides a resource-rational mechanistic account of another behavioral departure from EU: violations of betweenness (e.g., Camerer & Ho, 1994; Prelec, 1990). Betweenness, a relaxation of the independence axiom, has played a prominent role in developing generalizations of EU and their applications to game theory and macroeconomics (Camerer & Ho, 1994).

Recent work has shown that SbEU can account for the St. Petersburg paradox, a centuries-old paradox in human decision-making (Nobandegani, da Silva Castanheira, Shultz, & Otto, 2019a), and can provide a resource-rational mechanistic account of (ostensibly irrational) cooperation in one-shot Prisoner's Dilemma games, thus successfully bridging between game-theoretic decision-making and risky decision-making (Nobandegani, da Silva Castanheira, Shultz, & Otto, 2019b). There is also experimental confirmation of a counterintuitive prediction of SbEU: Deliberation leads people to move from one well-known bias, framing effect, to another well-known bias, the fourfold pattern of risk preferences (da Silva Castanheira; Nobandegani, & Otto, 2019).

Notably, the present study is simultaneously guided by, and consistent with, two empirically well-supported assumptions: (1) In probabilistic reasoning and judgment, a cognitive system accumulates information through sampling (e.g., Vul et al., 2014; Battaglia et al. 2013; Lake et al., 2017; Gershman, Horvitz, & Tenenbaum, 2015; Hertwig & Pleskac, 2010; Griffiths et al., 2012; Gershman, Vul, & Tenenbaum, 2012; Bonawitz et al., 2014), and (2) People engage in pairwise comparisons when choosing between multiple alternatives (e.g., Russo & Rosen, 1975; Noguchi & Stewart, 2014).

Camerer and Ho (1994) provide evidence suggesting that people are more likely to violate the independence and the betweenness axioms when presented with single-stage gambles than with multi-stage gambles (wherein, with some probability, the agent is presented with one risky gamble, and, with some other probability, with a different risky gamble). The present study particularly focused on single-stage gambles. Future work should investigate if the resource-rational process-level explanation presented in this work could also account for the forgoing tendency experimentally documented by Camerer and Ho (1994).

There have been several recent studies (see Lieder & Griffiths, 2018, for a review) attempting to show that many well-known (purportedly irrational) behavioral effects and cognitive biases can be understood as optimal behavior subject to computational and cognitive limitations (e.g., Griffiths, Lieder, & Goodman, 2015; Nobandegani, 2018; Lieder, Griffiths, Huys, & Goodman, 2018). The present study contributes to this line of work by providing a resource-rational process-level explanation of two (purportedly irrational) effects in risky decision-making. As such, our work suggests an alternative perspective on evaluating human rationality: To judge human rationality not by whether human behavior respects a set of axioms and/or principles (e.g., the independence axiom, the betweenness axiom, etc.), but by the extent to which human judgment and decision-making is in accord with rational process models acknowledging computational and cognitive limitations (e.g., Lieder & Griffiths, 2018; Nobandegani, 2018).

While the foregoing approach to evaluating human rationality is still in its infancy, and much work is needed to investigate the efficacy of this perspective, we hope to have made some progress in this direction.

Appendix

Proof of Theorem 1

There are three possible pairs $(T,C),(T,D),(C,D)$ that the decision-maker can begin with. In what follows we consider each possibility in turn, showing that only starting with the pair (C,D) grants the occurrence of risky DE.
Case 1. Starting with the pair (T,C):

$$\mathbb{P}(T|\{T,C,D\}) = \mathbb{P}(T|\{T,C\})\mathbb{P}(T|\{T,D\}) \leq \mathbb{P}(T|\{T,C\}).$$

The right-hand side inequality immediately follows from the basic axiom in probability that $\mathbb{P}(T|\{T,D\}) \leq 1$.
Case 2. Starting with the pair (T,D):

$$\mathbb{P}(T|\{T,C,D\}) = \mathbb{P}(T|\{T,D\})\mathbb{P}(T|\{T,C\}) \leq \mathbb{P}(T|\{T,C\}).$$

The right-hand side inequality immediately follows from the basic axiom in probability that $\mathbb{P}(T|\{T,D\}) \leq 1$.
Case 3. Starting with the pair (C,D):

$$\begin{aligned}\mathbb{P}(T|\{T,C,D\}) &= \mathbb{P}(C|\{C,D\})\mathbb{P}(T|\{T,C\}) + \\ &\quad \mathbb{P}(D|\{C,D\})\mathbb{P}(T|\{T,D\}) \qquad (8)\end{aligned}$$

Under the plausible assumption that $\mathbb{P}(T|\{T,D\}) > \mathbb{P}(T|\{T,C\})$ (due to the fact that T dominates D), it follows that the right-hand side of Eq. 8 is greater than $\mathbb{P}(T|\{T,C\})$, hence granting the occurrence of risky DE.

This completes the proof of Theorem 1. ■

References

Arrow, K. J. (1963). *Social Choice and Individual Values*. Yale university press.

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, *110*(45), 18327–18332.

Bernoulli, D. (1738/2011). Exposition of a new theory on the measurement of risk. In *The kelly capital growth investment criterion: Theory and practice* (pp. 11–24). World Scientific.

Bonawitz, E., Denison, S., Griffiths, T. L., & Gopnik, A. (2014). Probabilistic models, learning algorithms, and response variability: sampling in cognitive development. *Trends in Cognitive Sciences*, *18*(10), 497–500.

Camerer, C. F., & Ho, T.-H. (1994). Violations of the betweenness axiom and nonlinearity in probability. *Journal of Risk and Uncertainty*, *8*(2), 167–196.

Cary, M., & Reder, L. M. (2002). Metacognition in strategy selection. In *In: Metacognition: Process, function and use* (pp. 63–77). Springer.

da Silva Castanheira, K., Nobandegani, A. S., & Otto, A. R. (2019). Sample-based variant of expected utility explains effects of time pressure and individual differences in processing speed on risk preferences. In: *Proceedings of the* 41*st* *Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gershman, S. J., Vul, E., & Tenenbaum, J. B. (2012). Multistability and perceptual inference. *Neural Computation*, *24*(1), 1–24.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, 1317–1339.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, *7*(2), 217–229.

Griffiths, T. L., Vul, E., & Sanborn, A. N. (2012). Bridging levels of analysis for probabilistic models of cognition. *Current Directions in Psychological Science*, *21*(4), 263–268.

Hammersley, J., & Handscomb, D. (1964). *Monte carlo methods.* London: Methuen & Co Ltd.

Hershey, J. C., & Schoemaker, P. J. (1980). Prospect theory's reflection hypothesis: A critical examination. *Organizational Behavior and Human Performance*, *25*(3), 395–418.

Hertwig, R., & Pleskac, T. J. (2010). Decisions from experience: Why small samples? *Cognition*, *115*(2), 225–237.

Huber, J., Payne, J. W., & Puto, C. (1982). Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of Consumer Research*, *9*(1), 90–98.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–291.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*.

Lieder, F., & Griffiths, T. L. (2018). Resource-rational analysis: understanding human cognition as the optimal use of limited. Available on Researchgate.

Lieder, F., Griffiths, T. L., Huys, Q. J., & Goodman, N. D. (2018). Empirical evidence for resource-rational anchoring and adjustment. *Psychonomic Bulletin & Review*, *25*(2), 775–784.

Machina, M. J. (1987). Choice under uncertainty: Problems solved and unsolved. *Journal of Economic Perspectives*, *1*(1), 121–154.

Markowitz, H. (1952). The utility of wealth. *Journal of Political Economy*, *60*(2), 151–158.

Maule, A. J., & Svenson, O. (1993). Theoretical and empirical approaches to behavioral decision making and their relation to time constraints. In *Time Pressure and Stress in Human Judgment and Decision Making* (pp. 3–25). Springer.

Mohr, P. N., Heekeren, H. R., & Rieskamp, J. (2017). Attraction effect in risky choice can be explained by subjective distance between choice alternatives. *Scientific Reports*, *7*(1), 8942.

Nobandegani, A. S. (2018). The Minimalist Mind: On Mininality in Learning, Reasoning, Action, & Imagination. McGill University, PhD Dissertation.

Nobandegani, A. S., da Silva Castanheira, K., O'Donnell, T. J., & Shultz, T. R. (2019). On robustness: An undervalued dimension of human rationality. In *Proceedings of the* 17*th* *International Conference on Cognitive Modeling.* Montreal, QC.

Nobandegani, A. S., da Silva Castanheira, K., Otto, A. R., & Shultz, T. R. (2018). Over-representation of extreme events in decision-making: A rational metacognitive account. In: *Proceedings of the* 40*th* *Annual Conference of the Cognitive Science Society* (pp. 2391-2396). Austin, TX: Cognitive Science Society.

Nobandegani, A. S., da Silva Castanheira, K., Otto, A. R., & Shultz, T. R. (2019a). A resource-rational process-level account the St. Petersburg paradox. In *Proceedings of the* 41*st* *Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Nobandegani, A. S., da Silva Castanheira, K., Otto, A. R., & Shultz, T. R. (2019b). A resource-rational mechanistic approach to one-shot non-cooperative games: The case of prisoner's dilemma. In: *Proceedings of the* 41*st* *Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Noguchi, T., & Stewart, N. (2014). In the attraction, compromise, and similarity effects, alternatives are repeatedly compared in pairs on single dimensions. *Cognition*, *132*(1), 44–56.

Poor, H. V. (2013). *An Introduction to Signal Detection and Estimation.* Springer Science & Business Media.

Prelec, D. (1990). A "pseudo-endowment" effect, and its implications for some recent nonexpected utility models. *Journal of Risk and Uncertainty*, *3*(3), 247–259.

Roberts, M. J., & Erdos, G. (1993). Strategy selection and metacognition. *Educational Psychology*, *13*(3-4), 259–266.

Russo, J. E., & Rosen, L. D. (1975). An eye fixation analysis of multialternative choice. *Memory & Cognition*, *3*(3), 267–276.

Scholten, M., & Read, D. (2014). Prospect theory and the "forgotten" fourfold pattern of risk preferences. *Journal of Risk and Uncertainty*, *48*(1), 67–83.

Simonson, I. (1989). Choice based on reasons: The case of attraction and compromise effects. *Journal of Consumer Research*, *16*(2), 158–174.

Soltani, A., De Martino, B., & Camerer, C. (2012). A range-normalization model of context-dependent choice: a new model and evidence. *PLoS Computational Biology*, *8*(7), e1002607.

Speekenbrink, M., & Shanks, D. R. (2013). *Decision Making.* In *The Oxford Handbook of Cognitive Psychology.* Reisberg (Ed.). Oxford University Press.

Svenson, O. (1993). *Time Pressure and Stress in Human Judgment and Decision Making*. Springer Science & Business Media.

Tsetsos, K., Chater, N., & Usher, M. (2012). Salience driven value integration explains decision biases and preference reversal. *Proceedings of the National Academy of Sciences*, *109*(24), 9659–9664.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, *5*(2), 207–232.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, *5*(4), 297–323.

Von Neumann, J., & Morgenstern, O. (1947/2007). *Theory of games and economic behavior (commemorative edition).* Princeton University Press.

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and done? optimal decisions from very few samples. *Cognitive Science*, *38*(4), 599–637.

Wedell, D. H. (1991). Distinguishing among models of contextually induced preference reversals. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(4), 767.

On Robustness: An Undervalued Dimension of Human Rationality

Ardavan S. Nobandegani^{†,1,4}**, Kevin da Silva Castanheira**^{†,4}**, Timothy J. O'Donnell**^{3}**, & Thomas R. Shultz**^{2,4}
{ardavan.salehinobandegani, kevin.dasilvacastanheira}@mail.mcgill.ca
{timothy.odonnell,thomas.shultz}@mcgill.ca
[1]Department of Electrical & Computer Engineering, [2]School of Computer Science, [3]Department of Linguistics, [4]Department of Psychology
McGill University
†**Co-primary authors**

Abstract

Human rationality is predominantly evaluated by the extent to which the mind respects the tenets of normative formalisms like logic and probability theory, and is often invoked by appealing to the notion of optimality. Drawing mainly on Simon's bounded rationality principle, there has been a surge in the understanding of human rationality with respect to the limited computational and cognitive resources the mind is faced with. In this work, we focus on another fairly underappreciated yet crucial facet of rationality, robustness: insensitivity of a model's performance to miscalculations of its parameters. We argue that an integrative pursuit of three facets (optimality, efficient use of limited resources, and robustness) would be a fruitful approach to understanding the extent of human rationality. We present several novel formalizations of robustness and discuss a recently proposed metacognitively-rational model of risky choice which is surprisingly robust to under- and over-estimation of its focal parameter, nicely accounting for well-known framing effects in human decision-making under risk. We close by highlighting the ubiquitous presence of robustness in natural as well as artificial realms, and the implications of our work for rationalistic approaches to understanding human cognition at the algorithmic level of analysis.

Keywords: bounded rationality; robustness; rational process models; heuristics; metacognition

1 Introduction

Practical applications of complex algorithms to solve problems may not always prove to be the ideal approach to real world problems. Indeed, there are circumstances in which simple heuristics outperform optimal process models (Gigerenzer, 2008, 2010). A good example is that of Harry Markowitz previously outlined by Gigerenzer (2010). Markowitz is best known for his optimal asset-allocation model known as mean-variance portfolio, for which he won a Nobel prize in economics. However, when it came to his investments for retirement, he relied on a simpler intuitive heuristic known as the $1/N$ heuristic: allocate your resources equally to each of N alternatives (Gigerenzer, 2007). In fact, it has been shown that $1/N$ heuristic outperforms mean-variance portfolio which is sensitive to sampling error unless there are sufficiently many samples. In contrast, except for N which can be trivially set based on the number of investment options available to the agent, $1/N$ does not have any free parameters to estimate (Gigerenzer, 2010). Even when N is relatively small ($N = 50$), one needs a large amount of data, approximately 500 years of stock data, in order to outperform the simple $1/N$ heuristic. This is far more data than are available to the average investment firm (DeMiguel, Garlappi, & Uppal, 2009). Surely, a model like this cannot reasonably be considered as truly rational as attempted implementations would prove to be impractical. Process models which require gargantuan amounts of data to provide accurate parameter estimates do not possess the robustness necessary to be considered as rational in uncertain environments (Gigerenzer, 2008). Considerations of robustness when evaluating rationality of process models are far too scarce in the psychological literature.

There have been many attempts to define human rationality with respect to normative formalisms like logic and probability theory. In doing so, the notion of optimality is often invoked. Anderson's (1991) rational analysis approach specifically characterizes rationality as the extent to which a model approximates or attains optimality with respect to some reasonable objective (see also Chater & Oaksford, 1999). However, recent work has drawn on Simon's (1957, 1972) principle of bounded rationality to temper rationality by placing limitations on this model (e.g., Icard, 2014; Griffiths, Lieder, & Goodman, 2015, Nobandegani, 2018).

In this paper, we focus on an often overlooked, yet, crucial, factor in understanding human rationality: robustness. To corroborate this view, we discuss a recent metacognitively-rational model of Availability bias which is surprisingly robust to under- and over-estimation of its focal parameter, and which accounts for well-known framing effects in human decision-making under risk: the fourfold pattern of risk preferences in outcome probability (Kahneman & Tversky, 1992) and magnitude (Markovitz, 1952; Scholten & Read, 2014). We further elaborate on the key role of robustness at the cognitive and meta-cognitive levels, and articulate how robustness, along with principles of optimality and efficient use of limited resources, naturally leads to a key, yet, often overlooked, cognitive level: *meta*-metacognition. We present several formalizations of the notion of robustness, and close by discussing how various recent rationalistic approaches to cognition at the algorithmic level (*rational process models*, Griffiths et al., 2009, 2012) could be integrated with robustness, simultaneously enabling the pursuit of optimality, efficient use of limited resources, and robustness.

2 Facets of Human Rationality

In what follows, we first overview the two main facets of rationality predominately discussed in the psychological literature, and then turn our attention to a key, yet often overlooked, dimension of human rationality: robustness.

Optimality Perhaps the best characterized and extensively discussed facet of rationality is optimality. Optimality has been portrayed as the extent to which a model satisfies some objective (see Anderson, 1991, and Chater & Oaksford 1999). Models generally have as their objective the minimization or maximization of some objective function, or a combination thereof. For example, minimizing sum-of-squared error or cross entropy in training neural networks, minimizing probability of error in decision-making as in the Bayesian decision rule (Poor, 2013), maximizing expected utility as in expected utility theory (Von Neumann & Morgenstern, 1955), or minimizing the maximum probability of error as in the minimax decision rule (Poor, 2013). A model is considered optimal to the extent that it attains the set objectives. Thus, this facet of rationality depends on both the objective and the outcome, without regards to the context in which the cognitive system is operating. Surely, this cannot be taken as a comprehensive evaluation of rationality as it ignores many important factors affecting a cognitive system's performance, e.g., environmental uncertainty, lack of information, resource limitations, etc.

The importance of optimality—in evaluating what it means to be rational—is unquestionable. However we argue, like many others before us (Icard, 2014; Griffith et al., 2015; Nobandegani, 2018; Gigerenzer, 1998, Lewis, Howes, & Singh, 2014; Howes, Lewis, & Vera, 2009; Russell, Stuart & Subramanian, 1995; Russel 1997, *inter alia*), that there are other factors to take into consideration.

Economy In recent years, many have taken inspiration from Simon's (1957) bounded rationality to expand our understanding of human rationality. This concept is heavily based on the limitations of cognitive and computational resources imposed on the model when considering rationality. A boundedly rational agent need not fully optimize but find a solution which only *satisfices* certain criteria given the limitations at hand—both environmental and internal (Simon, 1957). The emphasis here is primarily on the circumstances and conditions under which the cognitive system operates,[1] highlighting the importance of the cognitive system's quest for *economy*: the economical use of limited computational and cognitive resources (e.g., time, memory, information). As opposed to optimality, which is predominately invoked with a disregard for such contextual limitations, the concept of economy is context-dependent. The concept of economy and its role in theorizing about human cognition are mainly pursued under titles like *ecological rationality* (Gigerenzer, 1998; Gigerenzer & Todd, 2012), *algorithmic rationality* (Halpern & Pass, 2011), *bounded-optimality* (Russell, Stuart & Subramanian, 1995; Russel 1997), *boundedly rational analysis* (Icard, 2014), *resource-rationality* (Griffiths et al., 2015), *computational rationality* (Lewis, Howes, & Singh, 2014), and *rational minimalist program* (Nobandegani, 2018).

Importantly, here there are broadly two approaches to economy. One assumes there is a necessary trade-off to be made between the two facets (e.g., Icard, 2014; Griffiths et al., 2015, Russell, Stuart & Subramanian, 1995; Russel 1997), while the other views the facets as largely independent (e.g., Gigerenzer, 2010; Nobandegani, 2018). For example, it has been surprisingly demonstrated that economical process models—often referred to as heuristics (*fast-and-frugal*, Gigerenzer, 2008)—can outperform optimal process models (Gigerenzer, 2010), thereby establishing that, at least in some settings, optimality and economy need not trade off.

Also interestingly, using limited knowledge, some algorithms can outperform or match algorithms which integrate all information available (i.e., multiple regression) (Gigerenzer, 2010). Drawing on the previously discussed example of investment, superior performance of the heuristic is chiefly due to its robustness with respect to uncertainty in parameter estimates (Gigerenzer, 2008). Only under extraordinary circumstances can the optimal, mean-variance portfolio model outperform the simple $1/N$ heuristic.

In the following sections, we shed light on another aspect of rationality which is not extensively discussed in the literature: robustness. Examples of robustness as an objective criterion are provided as well as several formalizations of it, providing formal and precise characterizations of this aspect and facilitating future evaluations of human rationality.

3 On Robustness

Although the concept of robustness is not new in the literature, it has been largely overlooked in discussions of rationality. Robustness has appeared previously in academic writing in a specialized and narrow sense (e.g., Gigerenzer, 2008; Lempert & Collins, 2007), largely without precise formal characterizations. In the field of decision-making, where attempts have been made to tackle the issue of uncertainty in model specifications (specifically the probability distributions of the parameters), robustness has been discussed (Lempert & Collins, 2007). There, importance is placed on not achieving the optimal solution, but dealing with uncertainty—trading off optimality for less sensitivity to violated assumptions (Lempert & Collins, 2007). We propose a similar view when evaluating process models of cognition in general. Robust models should be insensitive to inaccuracies of their parameters, with little or no decline in their performance. An agent should use models allowing them to perform optimally or near-optimally, regardless of the limitations imposed on them and possible miscalculations of model parameters.

At first, it may seem that robustness and economy are addressing the same concerns. However, further investigation of the implications of robustness as an independent facet of rationality reveals that these two facets are indeed distinct.

In fact, we can force a model to be economical (i.e., frugal) by restricting its use of resources (e.g., by limiting the amount of information the model is allowed to process). Nevertheless, this does not make the model robust with respect to miscalculations of its parameters. Let us elucidate this understanding in the context of a recent model by Piantadosi

[1] A reader familiar with Minimalist Program in linguistics (Chomsky, 1993), could see clear connections between the concept of *virtual conceptual necessity* and the topic under discussion here.

(2018). Surprisingly, Piantadosi (2018) presents a single-parameter model capable of fitting any scatter plot, on any number of points, to arbitrary precision. Despite having only a single parameter, this model is overly sensitive to parameter imprecision. We can have this model estimate its single parameter using only one randomly chosen point from the target scatter plot, thereby forcing the model to highly respect economy. Nonetheless, this does not alleviate the oversensitivity of the model with respect to parameter imprecision: Robustness is an intrinsic property of a model (either a model is sensitive to inaccuracies in parameter estimation or not), and it is independent of whether a model is economical.

Economy is primarily concerned with the strategic use of limited resources (e.g., computational, cognitive, etc.). In contrast, robustness is about insensitivity to inaccuracies in parameter estimation; the sources of these inaccuracies often boil down to the agent's incomplete knowledge of, and uncertainty about, its environment. However, incomplete knowledge and uncertainty are not the only factors responsible for an agent's inability to precisely estimate parameters.

There are several sources of uncertainty. First, uncertainty can come from changes in the environment. If one is attempting to estimate a value which changes over time, an estimate would be likely erroneous. Second, uncertainty can come from limited knowledge. An agent may not know all relevant information for the task at hand.[2] Third, even if an agent has all the relevant knowledge at their disposal, the computational power needed to accurately estimate parameters may be outside the agent's computational capacity.

Thus, miscalculations of parameters may be due to external (e.g., environmental changes) as well as internal (e.g., limited computational power) constraints. In that light, robustness can be characterized as preserved performance despite these constraints. Much like optimality and economy, robustness serves as a meta-level objective criterion for an intended cognitive level of analysis, reflecting on the quality of the model developed at that cognitive level.

4 Robustness as an Objective Criterion

In the following, we discuss in greater detail how robustness can serve as a meta-level objective criterion for human cognition at two distinct cognitive levels of analysis: the cognitive and meta-cognitive levels.

4.1 On the Cognitive Level

Reflecting on the robustness of cognitive models leads to a key level of analysis: metacognition. This level of analysis is analogous to the considerations afforded to the economy of cognitive models. To elaborate on the use of robustness as a meta-level objective criterion, we return to the investment example. The example of Harry Markowitz has been used to illustrate the success of heuristics over rational process models (Gigerenzer, 2010). Here, the optimal strategy is impractical to use as it requires a sizable amount of data (about 500 years of stock data) to accurately estimate parameter values. In other words, the optimal asset allocation strategy proposed by Markowitz (1952) would only result in the best outcome if the parameter values were known near-perfectly, as in a small world, but is inferior to heuristics in a large world, where parameter values need to be estimated from limited samples of information. The success of heuristics is largely due to the robustness of their performance afforded by insensitivity to imperfections in parameter estimates (see Gigerenzer, 2008).

Although the literature emphasizes limited number of samples as the main source of inaccuracies in parameter estimation, this account is incomplete. Gigerenzer (2010) argues that the optimality of mean-variance portfolio hinges on accurate parameter estimation using only a limited number of available samples. However, even if the samples would abound, one would need an extraordinary amount of computational power to estimate parameters adequately. Processing 500 years of stock data is no trivial task.

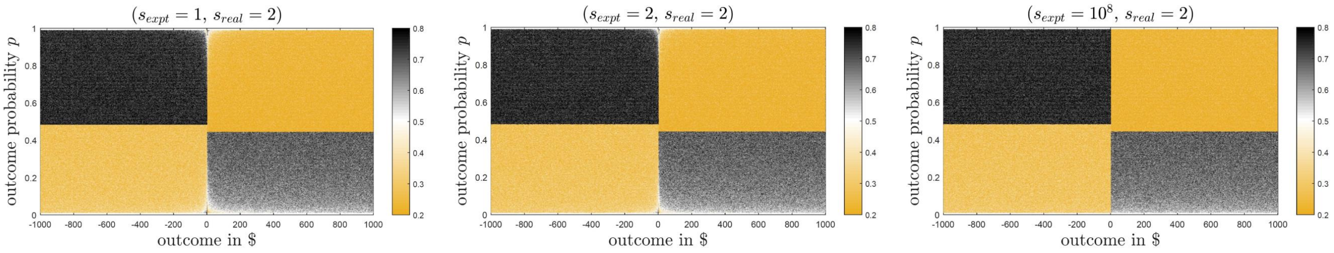
A noteworthy example of computational intractability being the primary source of miscalculations (as opposed to incomplete knowledge) eminently features in the problem of finding Nash equilibria in game theory. Even when everything about the game is known (aka complete-knowledge games), finding a (mixed) Nash equilibrium is computationally intractable (more precisely, it is **PPAD**-complete, Daskalakis et al., 2009), attesting to the claim that miscalculations may sometimes result from computational complexity barriers, not lack of information.

4.2 On Meta-Cognitive Level

Following the logic of the previous section, reflections on robustness can be applied to metacognitive models leading to another key level of analysis: meta-metacognition. "Meta-metacognition" is scarcely used in the literature. Previous uses have either been in a narrow sense (e.g., Arnold, 2013; Buratti & Allwood, 2012), or as a term whose content is not concretely specified, characterized broadly as "reflection" on the metacognitive level without articulating precisely what that reflection means (e.g., Renkl et al., 1996; Efklides & Misailidi, 2010). In what follows, we seek to clarify what meta-metacognitive considerations entail and provide concrete examples.

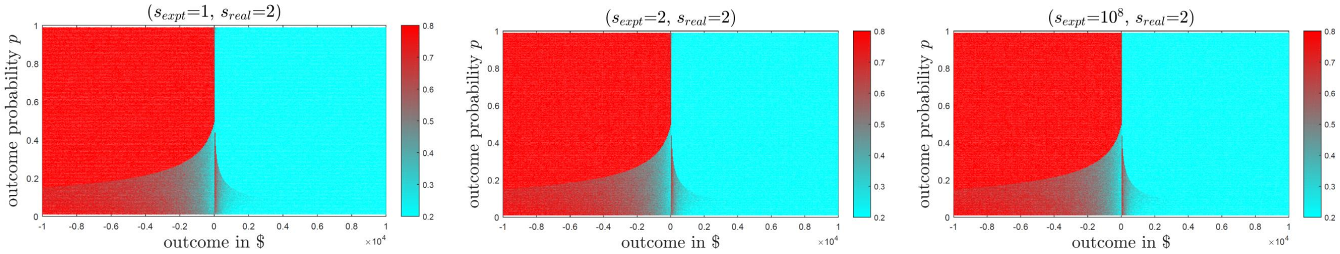
Research on metacognitively-rational models is still in its infancy, and little work is done on this exciting topic (e.g., Lieder & Griffiths, 2017; Nobandegani, da Silva Castanheira, Otto, & Shultz, 2018). A good example of such models is the recent work by Lieder and Griffiths (2017) on rational models of strategy selection. Despite its great performance, a pre-theoretic evaluation of this model suggests that it would not score high on robustness as its performance largely hinges on accurate parameter estimations. In this model, strategies (e.g., heuristics) are evaluated based on their previous performance on problems which share similar features (Lieder

[2] More precisely, uncertainty due to unanticipated environmental changes can be seen as an instance of incomplete knowledge with respect to future states of the environment. In that sense, the first source of uncertainty mentioned above is a special case of the second source.



(a) Under-estimation (b) Exact (c) Severe over-estimation

Figure 1: Accounting for the fourfold pattern of risk preferences in outcome probability (Kahnemnan & Tversky, 1992) using Nobandegani et al.'s (2018) metacognitively-rational model. Each figure plots the probability of choosing the risky choice, depending on the probability of outcome involved in the risky choice (p) and the amount of outcome in dollars; see Nobandegani et al. (2018) for details. A striking resemblance can be observed between **(a,b,c)**.



(a) Under-estimation (b) Exact (c) Severe over-estimation

Figure 2: Accounting for the fourfold pattern of risk preferences in outcome magnitude (Markowitz, 1952) using Nobandegani et al.'s (2018) metacognitively-rational model. Each figure plots the probability of choosing the risky choice, depending on the probability of outcome involved in the risky choice (p) and the amount of outcome in dollars; see Nobandegani et al. (2018) for details. A striking resemblance can be observed between **(a,b,c)**.

& Griffiths, 2017), and the perceived benefits from applying a given strategy (e.g., rewards) are contrasted with the costs (e.g., opportunity costs of the strategy's execution time) to supply a "value of computation" (Lieder & Griffiths, 2017; Horvitz, 1990). To evaluate the performance of a given strategy, accurate estimates of pivotal features of the problem are needed. However, in complex environments where these estimates are not readily available or easily computable, such a metacognitively-rational model for strategy selection would likely fail to satisfy robustness considerations. Therefore, one could say that such metacognitively-rational models are not *meta*-metacognitively rational.

Recent work on rational process models has proven to successfully capture all the three facets of optimality, economy, and robustness. In particular, drawing on the work by Lieder, Griffiths, & Hsu (2017) which applied importance sampling to estimate the expected utility of an action, recent work by Nobandegani et al. (2018) provides a metacognitively-rational process model of Availability Bias (Tversky & Kahneman, 1972), and the fourfold pattern of risk preferences in probability outcome (Tversky & Kahneman, 1992) and in outcome magnitude (Markowitz, 1952), by rationally taking into consideration the amount of time available for decision-making. Concretely, the model takes into account the number of samples the decision-maker draws before making a decision (Nobandegani et al., 2018). This model is in accord with a recent, empirically supported line of research suggesting that people draw very few samples (i.e., few mental simulations) in reasoning and decision-making (Vul et al., 2014; Battaglia et al. 2013; Lake et al., 2017; Gershman, Horvitz, & Tenenbaum, 2015; Hertwig & Pleskac, 2010; Griffiths et al., 2012; Gershman, Vul, & Tenenbaum, 2012; Bonawitz et al., 2014). Further investigation of this metacognitive model reveals that the performance of the model depends on the actual number of samples the model gets to draw (denoted by s_{real}), and not on the number of samples it anticipates drawing (denoted by s_{expt}, with s_{expt} serving as a priori estimate of s_{real}). In other words, the model is robust with respect to inaccurate estimation of the number of samples it gets to draw, both under- and over-estimations. Sensitivity analysis shows that the model is indeed surprisingly robust with respect to its focal parameter s_{expt}. Consistent with the past literature providing evidence for people drawing very few samples in reasoning and decision-making, when s_{real} assumes a value of 2, the fourfold pattern of risk preferences in outcome probability persistently emerges for exact, under- and severely over-underestimated values of s_{expt} (i.e., $s_{\text{expt}} = 2$, $s_{\text{expt}} = 1$, and $s_{\text{expt}} = 10^8$, respectively); see Fig. 1.

The model is also strikingly robust when producing more nuanced patterns of behavior like Markowitz's (1952) fourfold pattern of risk preference in outcome magnitude; see Fig. 2. In fact, it was this model which inspired our considerations of the importance of robustness in evaluating rational cognitive models. (Thus, process models satisfying near-perfectly all three facets of rationality are indeed possible.) Nevertheless, Nobandegani et al. (2018) failed to provide a formal characterization of the robustness of their model.

In the following section, we formalize robustness to provide precise characterizations of this notion and facilitate future evaluations of rationality. As we show, our work additionally allows us to formally characterize the robustness of the Nobandegani et al.'s (2018) model and the $1/N$ heuristic.

5 Formalization of Robustness

We first consider robustness with respect to real-valued parameters, and finally show how these formalizations can be adapted to the discrete-valued parameters case.

Def. 1 (i^{th}-order locally-robustness) Model M_θ parameterized by θ is i^{th}-order locally-robust at $\theta = \theta_0$ iff M_θ's performance measure $T[M_\theta]$ is insensitive to infinitesimal deviations of θ from θ_0, all the way up to the i^{th}-order. That is, formally, $\forall j \leq i$, $\nabla^{(j)}_{\theta=\theta_0} T[M_\theta] = 0$, where $\nabla^{(j)}_{\theta=\theta_0}(\cdot)$ denotes the j^{th}-order gradient w.r.t. θ and evaluated at $\theta = \theta_0$.

Def. 2 (i^{th}-order ε-locally-robustness) Model M_θ parameterized by θ is i^{th}-order ε-locally-robust at $\theta = \theta_0$ iff M_θ's performance measure $T[M_\theta]$ satisfies: $\forall j \leq i$, $|\nabla^{(j)}_{\theta=\theta_0} T[M_\theta]| \leq \varepsilon$.

Definitions 1 and 2 are founded on an important understanding based on the concept of Taylor series in calculus: The more number of higher-order derivatives of function $f(x)$ are zero (or nearly-zero) at $x = x_0$, the wider and flatter $f(x)$ is in the neighborhood of $x = x_0$. Extending Definitions 1 and 2 to the case of multi-parameter models (as opposed to a single-parameter model M_θ), we arrive at the following:

Def. 3 (i^{th}-order singly-locally-robustness) Model $M_{\theta,\theta'}$ parameterized by $\{\theta, \theta^{'}\}$ is i^{th}-order singly-locally-robust at $(\theta = \theta_0, \theta^{'} = \theta^{'}_0)$ iff $M_{\theta,\theta'}$'s performance measure $T[M_{\theta,\theta'}]$ is insensitive to infinitesimal deviations of θ from θ_0, all the way up to the i^{th}-order, when $\theta^{'}$ is held fixed at $\theta^{'}_0$ (denoted by $\theta^{'} := \theta^{'}_0$). That is, formally, $\forall j \leq i$, $\nabla^{(j)}_{\theta=\theta_0|\theta':=\theta'_0} T[M_{\theta,\theta'}] = 0$, where $\nabla^{(j)}_{\theta=\theta_0|\theta':=\theta'_0}(\cdot)$ denotes the j^{th}-order gradient w.r.t. $\theta^{'}$ and evaluated at $\theta^{'} = \theta^{'}_0$, when $\theta^{'}$ is held fixed at $\theta^{'}_0$.

Def. 4 (i^{th}-order ε-singly-locally-robustness) Model $M_{\theta,\theta'}$ parameterized by $\{\theta, \theta^{'}\}$ is i^{th}-order ε-singly-locally-robust at $(\theta = \theta_0, \theta^{'} = \theta^{'}_0)$ iff $M_{\theta,\theta'}$'s performance measure $T[M_{\theta,\theta'}]$ satisfies: $\forall j \leq i$, $|\nabla^{(j)}_{\theta=\theta_0|\theta':=\theta'_0} T[M_{\theta,\theta'}]| \leq \varepsilon$.

Definitions 1 to 4 can be straightforwardly adapted to to the case of discrete-valued parameters, with operations $\nabla^{(j)}_{\theta=\theta_0}(\cdot)$ and $\nabla^{(j)}_{\theta=\theta_0|\theta':=\theta'_0}(\cdot)$ being replaced, respectively, with the operations $D^{(j)}_{\theta=\theta_0}(\cdot)$ and $D^{(j)}_{\theta=\theta_0|\theta':=\theta'_0}(\cdot)$ defined as follows:

$$D^{(j)}_{\theta=\theta_0} f(\theta) \triangleq \big(f(\theta_0 + i) - f(\theta_0)\big)/i, \tag{1}$$

$$D^{(j)}_{\theta=\theta_0|\theta':=\theta'_0} g(\theta, \theta^{'}) \triangleq \big(g(\theta_0 + i, \theta^{'}_0) - g(\theta_0, \theta^{'}_0)\big)/i. \tag{2}$$

Eyeballing Figs. 1 and 2 reveals that Nobandegani et al.'s (2018) metacognitively-rational model is approximately[3] $(10^8 - 2)^{\text{th}}$-order singly-locally-robust at $(s_{\text{expc}} = 2, s_{\text{real}} = 2)$, with the performance measure being the most probable choice suggested by the model (i.e., the preference for the risky choice vs. the safe one). Note that, given that Definition 4 is a relaxation of Definition 3, Nobandegani et al.'s (2018) model is also approximately $(10^8 - 2)^{\text{th}}$-order ε-singly-locally-robust at $(s_{\text{expc}} = 2, s_{\text{real}} = 2)$, $\forall \varepsilon \in \mathbb{R}_+$.

Our formalism also allows us to provide a formal characterization of the robustness of $1/N$ heuristic. Using Def. 2, it is easy to mathematically show that, for any $N_0, i \in \mathbb{N}$, the $1/N$ heuristic is i^{th}-order 0.5-locally-robust at $N = N_0$, with the performance measure being the portion of resources to be allocated to each of N investment alternatives.

6 General Discussion

Examples of robustness in natural and man-made artifacts are abundant and often ensured by adding redundancy. In biological systems, robustness can be characterized as the maintenance of some functionality (e.g., phenotype) despite perturbations (e.g., genetic variation) and achieved through many means, one being redundancy (Kitano, 2004; Felix, 2015). At the genetic level, this can be seen as genes with overlapping products or at the network level with different mechanisms serving the same purpose: glycolysis and oxidative phosphorylation both produce ATP under different conditions (anaerobic and aerobic respectively) (Kitano, 2004). Modularity and decoupling of low-level variations from high-level functionality (e.g., genotype and phenotype) are also seen as sources of robustness in a biological system (Felix, 2015). Furthermore, a modular view of the brain fits nicely with the concept of robustness: Locally perturbations of a module should leave other modules unaffected. In fact, Fodor's (1983) view of low-level system modularity (e.g., perception and language) provides another example of modularity of biological systems. Similarly, decoupling of higher-level systems from lower-level systems is in accordance with the proposed view of robustness. The ubiquitous presence of robustness in biological systems suggests its importance in successful fulfillment of a system's goals (Felix, 2015).

Similarly, in artificial systems robustness is engineered into systems by particularly capitalizing on the benefits of adding redundancy to systems. For example, network architecture comprises several modules with overlapping functionalities, as opposed to a single integrative module (Kurose & Ross, 2009). In information theory, robustness is featured in error detection/correction codes for communicating informa-

[3] The rationale behind using the term "approximately" is that there could be some (x,y)-coordinates whose values are not perfectly invariant across Fig. 1(a-c) (and, likewise, across Fig. 2(a-c)). However, note that even if such (x,y)-coordinates do exist, they are very scarce, as evidenced by the striking resemblance of Fig. 1(a-c) (and, likewise, Fig. 1(a-c)). We could have provided a more rigorous characterization of this possibility using notions analogous to *almost-everywhere* in measure theory. However, for the sake of keeping the formalism simple, we refrained from that.

tion over a noisy medium, which by introducing redundancy into the transmitted code ensures that possible errors can be detected/corrected at the receiver (Cover & Thomas, 2012).

6.1 Toward Robust Models

Outperforming optimal models when accurate-enough parameter estimates cannot be obtained is evidence for their lack of robustness (Gigerenzer, 2008). The success of recent models at capturing and providing metacognitively-rational bases for intricate behavior patterns (Nobandegani et al., 2018) suggests that many of the findings in the psychology literature which are often considered "irrational" may be well-explained by appealing to metacognition or *meta*-metacognition. Indeed, the modeling work which inspired these reflections did not explicitly consider robustness. However, considerations of robustness should not be left up to serendipity. Rather, we believe that robustness of process models should be another factor in the modeler's objective set. Unlike considerations of sensitivity analysis, optimality and economy are not treated as an after-thought. Why should robustness be any different? Nobandegani et al.'s (2018) metacognitively-rational model achieves all three facets of rationality near-perfectly: optimality, economy and robustness.

Several frameworks for theorizing about cognitive process models have been proposed to simultaneously attain optimality and economy (e.g., Icard, 2014; Griffiths et al., 2015; Nobandegani, 2018). An important question is whether and how robustness can be integrated into these frameworks? Drawing on statistical learning theory and machine learning, we proposed a possible solution.

But first it is important to highlight a key connection between the concept of robustness discussed here and that of over-fitting in statistical learning theory and machine learning. If models (or theories) are selected from overly complex hypothesis sets, the learned model would likely overfit the observed data and would not generalize well. Importantly, an over-fitting model would also be fragile (as opposed to robust), since slight perturbations of the training patterns would lead to the selection of a radically different model. In that light, over-fitting models are fundamentally unrobust.

Inspired by these understandings, we suggest that current modeling frameworks should search for algorithms that satisfy some general characteristics ensuring robustness. A prominent such characteristic is for hypothesis sets to be not overly complicated, to avoid over-fitting. Importantly, several important theoretical measures of complexity of a hypothesis set have been already extensively studied in the statistical leaning theory, e.g., VC-dimentsion (Vapnik & Chervonenkis, 1971), Natarajan dimension (Natarajan, 1989), and Rademacher complexity (see Bartlett & Mendelson, 2002).

Although previous work has largely focused on the aspects of optimality and economy, underplaying the role of robustness in rationality, we hope to have given robustness the attention it truly deserves.

References

Anderson, J. R. (1991). Is human cognition adaptive? *Behavioral and Brain Sciences*, *14*(3), 471–485.

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*(Nov), 463–482.

Battaglia, P. W., Hamrick, J. B., & Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, *110*(45), 18327–18332.

Bonawitz, E., Denison, S., Griffiths, T. L., & Gopnik, A. (2014). Probabilistic models, learning algorithms, and response variability: sampling in cognitive development. *Trends in cognitive sciences*, *18*(10), 497–500.

Buratti, S., & Allwood, C. M. (2012). The accuracy of meta-metacognitive judgments: Regulating the realism of confidence. *Cognitive processing*, *13*(3), 243–253.

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in cognitive sciences*, *3*(2), 57–65.

Chomsky, N. (1993). A minimalist program for linguistic theory. *The view from Building*, *20*, 1–52.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. (2009). The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, *39*(1), 195–259.

DeMiguel, V., Garlappi, L., Nogales, F. J., & Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, *55*(5), 798–812.

Efklides, A., & Misailidi, P. (2010). *Trends and prospects in metacognition research*. Springer Science & Business Media.

Félix, M.-A., & Barkoulas, M. (2015). Pervasive robustness in biological systems. *Nature Reviews Genetics*, *16*(8), 483.

Foder, J. A. (1983). *The modularity of mind: an essay on faculty psychology.* MIT Press.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gershman, S. J., Vul, E., & Tenenbaum, J. B. (2012). Multistability and perceptual inference. *Neural computation*, *24*(1), 1–24.

Gigerenzer, G. (1998). Ecological intelligence: An adaptation for frequencies. In *The evolution of mind* (pp. 9–29). Oxford University Press.

Gigerenzer, G. (2008). *Rationality for mortals: How people cope with uncertainty*. Oxford University Press.

Gigerenzer, G. (2010). Moral satisficing: Rethinking moral behavior as bounded rationality. *Topics in cognitive science*, *2*(3), 528–554.

Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. *Topics in cognitive science*, *1*(1), 107–143.

Gigerenzer, G., & Todd, P. M. (2012). Ecological rationality: the normative study of heuristics. In *Ecological rationality: Intelligence in the world* (pp. 487–497). Oxford University Press.

Griffiths, T., Levy, R., McKenzie, C. R., Steyvers, M., Tenenbaum, J., & Vul, E. (2009). Rational process models. In *Proceedings of the cognitive science society* (Vol. 31).

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, *7*(2), 217–229.

Griffiths, T. L., Vul, E., & Sanborn, A. N. (2012). Bridging levels of analysis for probabilistic models of cognition. *Current Directions in Psychological Science*, *21*(4), 263–268.

Halpern, J. Y., & Pass, R. (2011). Algorithmic rationality: Adding cost of computation to game theory. *ACM SIGecom Exchanges*, *10*(2), 9–15.

Hertwig, R., & Pleskac, T. J. (2010). Decisions from experience: Why small samples? *Cognition*, *115*(2), 225–237.

Horvitz, E. J. (1990). *Computation and Action under Bounded Resources*. PhD Dissertation, Stanford University.

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: implications for testing theories of cognition and action. *Psychological review*, *116*(4), 717.

Icard, T. (2014). Toward boundedly rational analysis. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 36).

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive psychology*, *3*(3), 430–454.

Kitano, H. (2004). Biological robustness. *Nature Reviews Genetics*, *5*(11), 826.

Kurose, J. F., & Ross, K. W. (2009). *Computer networking: a top-down approach* (Vol. 4). Addison Wesley Boston, USA.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*.

Lempert, R. J., & Collins, M. T. (2007). Managing the risk of uncertain threshold responses: comparison of robust, optimum, and precautionary approaches. *Risk analysis*, *27*(4), 1009–1026.

Lewis, R. L., Howes, A., & Singh, S. (2014). Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science*, *6*(2), 279–311.

Lieder, F., Griffiths, T., & Hsu, M. (2017). Overrepresentation of extreme events in decision making reflects rational use of cognitive resources. *Psychological review*.

Lieder, F., & Griffiths, T. L. (2017). Strategy selection as rational metareasoning. *Psychological review*, *124*(6), 762.

Markowitz, H. (1952). The utility of wealth. *Journal of political Economy*, *60*(2), 151–158.

Natarajan, B. K. (1989). On learning sets and functions. *Machine Learning*, *4*(1), 67–97.

Nobandegani, A. S. (2018). The Minimalist Mind: On Mininality in Learning, Reasoning, Action, & Imagination. McGill University, PhD Dissertation.

Nobandegani, A. S., da Silva Castanheira, K., Otto, A. R., & Shultz, T. R. (2018). Over-representation of extreme events in decision-making: A rational metacognitive account. In: *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 2391-2396). Austin, TX: Cognitive Science Society.

Piantadosi, S. T. (2018). One parameter is always enough. *AIP Advances*, *8*(9), 095118.

Pollard, D. (2012). *Convergence of stochastic processes*. Springer Science & Business Media.

Poor, H. V. (2013). *An Introduction to Signal Detection and Estimation*. Springer Science & Business Media.

Renkl, A., Mandl, H., & Gruber, H. (1996). Inert knowledge: Analyses and remedies. *Educational Psychologist*, *31*(2), 115–121.

Russell, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, *94*(1-2), 57–77.

Russell, S. J., & Subramanian, D. (1995). Provably bounded-optimal agents. *Journal of Artificial Intelligence Research*, *2*, 575–609.

Scholten, M., & Read, D. (2014). Prospect theory and the forgotten fourfold pattern of risk preferences. *Journal of Risk and Uncertainty*, *48*(1), 67–83.

Simon, H. A. (1957). *Models of Man*. Wiley.

Simon, H. A. (1972). Theories of bounded rationality. *Decision and organization*, *1*(1), 161–176.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, *5*(4), 297–323.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity* (pp. 11–30). Springer.

von Von Neumann, J., & Morgenstern, O. (1955). *The theory of games and economic behavior*. Princeton University Press.

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and done? optimal decisions from very few samples. *Cognitive science*, *38*(4), 599–637.

Bringing Order to the Cognitive Fallacy Zoo

Ardavan S. Nobandegani[1,3], William Campoli[2], & Thomas R. Shultz[2,3]
{ardavan.salehinobandegani, william.campoli}@mail.mcgill.ca
{thomas.shultz}@mcgill.ca
[1]Department of Electrical & Computer Engineering, McGill University
[2]School of Computer Science, McGill University
[3]Department of Psychology, McGill University

Abstract

Investigations into human reasoning, judgment and decision-making have led to the finding of numerous cognitive biases and fallacies, with new ones continually emerging, leading to a state of affairs which can fairly be characterized as the cognitive fallacy zoo! In this work, we formally present a principled way to bring order to this zoo. We introduce the idea of establishing implication relationships (IRs) between cognitive fallacies, formally characterizing how one fallacy implies another. IR is analogous to, and partly inspired by, the fundamental concept of reduction in computational complexity theory. We present several examples of IRs involving experimentally well-documented cognitive fallacies: base-rate neglect, availability bias, conjunction fallacy, decoy effect, framing effect, and Allais paradox. We conclude by discussing how our work: (i) allows for identifying those pivotal cognitive fallacies whose investigation would be the most rewarding research agenda, and importantly (ii) permits a systematized, guided research program on cognitive fallacies, motivating influential theoretical as well as experimental avenues of future research.

Keywords: Cognitive biases and fallacies; cognitive fallacy map; cognitive processes;

1 Introduction

Over the past few decades, empirical investigations into human reasoning, judgment and decision-making have led to the discovery of new cognitive fallacies, giving rise to a large, ever-growing number of documented biases and fallacies, a state of affairs which can fairly be characterized as a zoo[1] of cognitive fallacies (e.g., Tversky & Kahneman, 1973, 1981b). A glance at over a hundred cognitive fallacies listed on Wikipedia attests to this claim (see Fig. 1).

In this work, we formally present a principled way to bring order to the cognitive fallacy zoo, allowing for a precise characterization of how various fallacies relate to one another. We introduce the idea of establishing an *implication relationship* (IR) (denoted by $\rightsquigarrow$) between a pair of cognitive fallacies, formally characterizing how the occurrence of one fallacy implies another. More formally, for two cognitive fallacies A, B, the expression $A \rightsquigarrow B$ denotes that A leads to B, i.e., the occurrence of A logically implies the occurrence of B. As a proof-of-concept, we present several examples of IRs involving experimentally well-documented cognitive fallacies: base-rate neglect (Tversky & Kahneman, 1981a), availability bias (Kahneman & Tversky, 1973), conjunction fallacy (Kahneman & Tversky, 1983), decoy effect (Huber, Joel, & Puto, 1982), framing effect (Tversky & Kahneman, 1981b) and Allais paradox (Allais, 1953).

The idea of establishing IRs between pairs of cognitive fallacies is analogous to, and partly inspired by, the foundational concept of reduction in computational complexity theory (see Karp, 1972; Papadimitriou, 2003; Arora & Barak, 2009; Sipser 2006), which has played a profound role in theoretical computer science, allowing to formally establish how various computational problems relate to each other and how the solution to one sheds light on that of another. After a brief discussion on the role of reduction in computational complexity, we return to the formal characterization of the notion of IR and subsequently present several examples of IRs involving experimentally well-documented cognitive fallacies. But first, some historical notes on reduction in computational complexity.

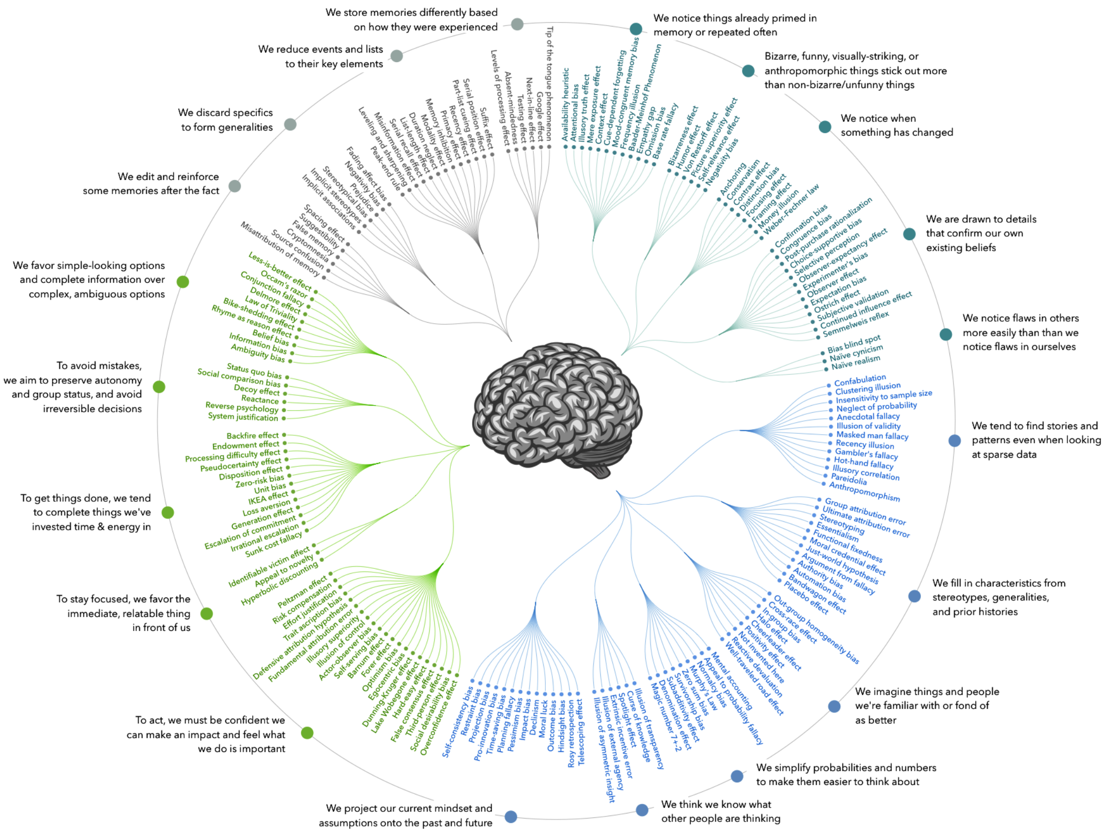
2 Reduction in Computational Complexity

The notion of reduction plays a fundamental role in computational complexity theory, and in theoretical computer science more generally. Informally put, a computational problem A is *reducible* to computational problem B, if every instance of A can be transformed into an instance of B. Therefore, the reduction of A to B offers an indirect way of solving A, by first reducing A to B, and then solving B.

To further clarify the idea of reduction, we provide two examples. As a first example, consider two well-known computational problems, namely, HAMILTONIAN-PATH and HAMILTONIAN-CYCLE. The HAMILTONIAN-PATH problem is defined as follows: given a (directed) graph G, is there a path which visits every node of G exactly once? The HAMILTONIAN-CYCLE is defined as follows: given a (directed) graph G, is there a cycle which visits every node of G exactly once? It turns out that HAMILTONIAN-CYCLE is reducible to HAMILTONIAN-PATH. Given that the definitions HAMILTONIAN-CYCLE and HAMILTONIAN-PATH are closely related (since a cycle is a path with its endpoints coinciding), this reduction is not especially surprising.

As a second example, let us consider HAMILTONIAN-PATH together with the 3-COLORABILITY problem, defined as follows: given a graph G and 3 distinct colors, can you color the nodes of G such that the endpoints of every edge are colored differently? At fist glance, the HAMILTONIAN-PATH and 3-COLORABILITY appear to have no connection with one another whatsoever. Surprisingly, however, it

[1] The term 'cognitive fallacy zoo' is inspired by an analogous terminology in the computational complexity literature, called 'complexity zoo.' For details, visit: `https://complexityzoo.uwaterloo.ca/Complexity_Zoo`



DESIGNHACKS.CO · CATEGORIZATION BY BUSTER BENSON · ALGORITHMIC DESIGN BY JOHN MANOOGIAN III (JM3) · DATA BY WIKIPEDIA

Figure 1: Cognitive fallacies listed on Wikipedia. Besides qualitatively categorizing them into classes depending on the context in which they occur, to date there exists no principled way of bringing order to these fallacies, allowing for formally characterizing how one fallacy relates to another.

turns out that HAMILTONIAN-PATH can be reduced to 3-COLORABILITY.[2] Thus, the question of whether a graph G has a Hamiltonian path can be resolved by answering if a corresponding graph G' is 3-colorable.

The idea of reduction has had profound implications for theoretical computer science, allowing for formally connecting seemingly unrelated computational problems to one another (see Fig. 2(a)). Had reduction not been introduced into theoretical computer science, every computational problem would have had to be investigated on its own, because the solution to one would not have shed any light on the solution to others. It was a major breakthrough when Karp (1972) showed that a key computational problem called SATISFIABILITY could be reduced to a number of other well-known computational problems, a contribution for which he was eventually awarded the Turing award in 1985. It is also worth noting that the (in)famous $\mathcal{P}$ vs. $\mathcal{NP}$ problem in theoretical computer science, at its core, concerns the possibility or impossibility of establishing a particular form of reduction.

One might wonder if an idea broadly analogous to reduction in computational complexity could be introduced into cognitive science, allowing for formally connecting seemingly different cognitive fallacies with one another, and, hence, bring order to the cognitive fallacy zoo in a principled manner. Primarily motivated by this, and, by analogy with the notion of reduction in theoretical computer science, we introduce the idea of establishing IRs between cognitive fallacies, formally characterizing how one fallacy would imply another.

3 Implication Relationships: Formalization

In what follows, we first formally introduce the idea of establishing an *implication relationship* (IR) (denoted by $\rightsquigarrow$) between a pair of cognitive fallacies, followed by several examples of IRs involving experimentally well-documented cognitive fallacies.

Definition (Implication Relationship). For two cognitive fallacies/biases A, B, the fallacy A is said to *implicate* the fallacy B (denoted by $A \rightsquigarrow B$) if and only if the occurrence of A *logically implies* the occurrence of B.

[2] The reduction can be established by a chain of straightforward reductions: HAMILTONIAN-PATH to SAT, SAT to 3SAT, and finally, 3SAT to 3-COLORING.

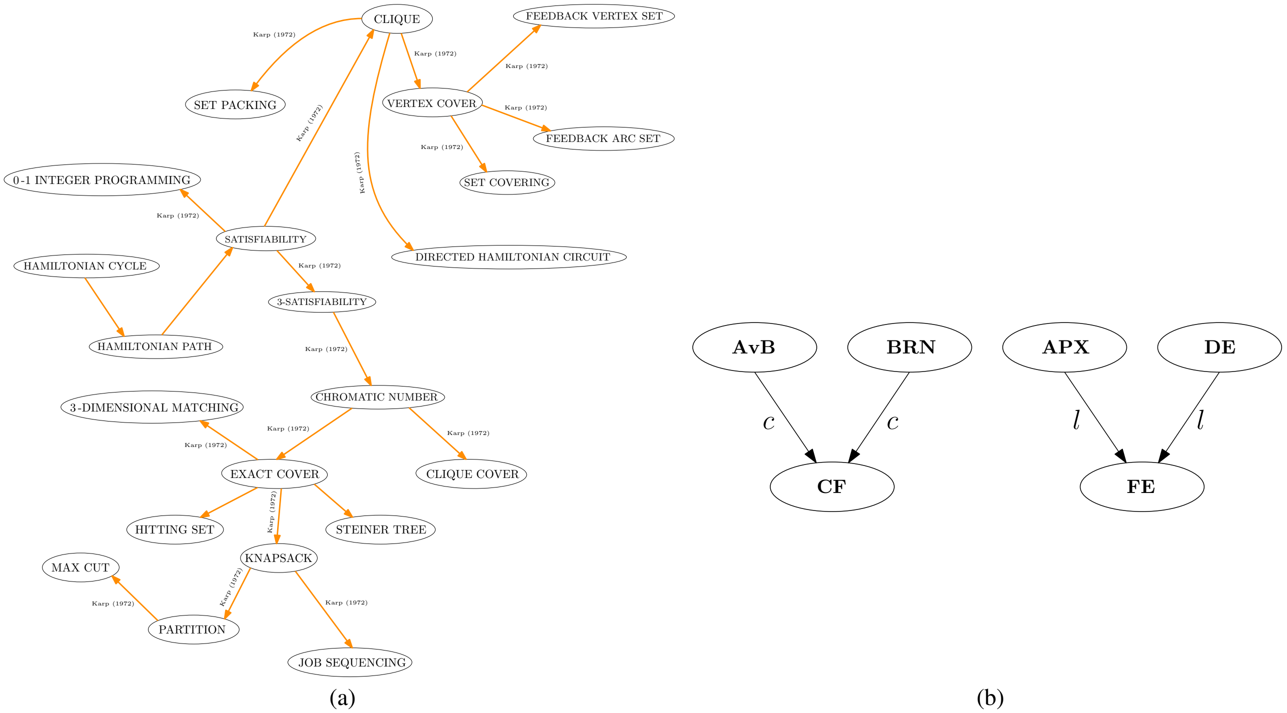


Figure 2: **(a)** A map showing reductions (directed gold lines) between a set of important computational problems (ovals) in theoretical computer science, formally characterizing how one problem is related to another. The first demonstration of a particular reduction from one problem to another is listed on the corresponding arrow between the two. **(b)** The IRs formally established in this paper, among well-known cognitive fallacies (AvB: Availability bias, BRN: base-rate neglect, CF: conjunction fallacy, APX: Allais paradox, DE: Decoy effect, FE: framing effect). Letters c (for causal) and l (for logical) on the arrows specify the type of an IR; see the Discussion section for details.

4 Examples on Implication Relationships

As a proof-of-concept, We next present several examples of IRs involving experimentally well-documented cognitive fallacies, namely, base-rate neglect (Tversky & Kahneman, 1981a), availability bias (Kahneman & Tversky, 1973), conjunction fallacy (Kahneman & Tversky, 1983), decoy effect (Huber et al., 1982), framing effect (Tversky & Kahneman, 1981b), and Allais paradox (Allais, 1953).

4.1 Case Study 1: Decoy Effect $\rightsquigarrow$ Framing Effect

As our first example, we formally establish an IR between two well-documented cognitive fallacies, namely, the decoy effect (DE) and the framing effect (FE).

The Framing Effect (FE): If people produce different responses for two equivalent tasks, the framing effect (FE) has occurred (Tversky & Kahneman, 1981b; Kahneman & Tversky, 1984). In that light, FE is a violation of the extensionality principle (Bourgeois-Gironde & Giraud, 2009), which prescribes that two equivalent tasks should elicit the same response.

FE is well captured by Tversky & Kahneman (1981b): Subjects were asked to “imagine that the U.S. is preparing for the outbreak of an unusual Asian disease, which is expected to kill 600 people. Two alternative programs to combat the disease have been proposed. Assume the exact scientific estimate of the consequences of the programs are as follows.” In one condition, subjects were presented with a choice between Programs A and B, while in another condition, subjects were asked to choose between Programs C and D:

Program A: 200 people will be saved.

Program B: There is a $\frac{1}{3}$ probability that 600 people will be saved, and a $\frac{2}{3}$ probability that no people will be saved.

Program C: 400 people will die.

Program D: there is a $\frac{1}{3}$ probability that nobody will die, and a $\frac{2}{3}$ probability that 600 people will die.

Despite the equivalence of these Programs pairs, a majority of the first group preferred Program A (= C), while a majority of the second group preferring Program D (= B).

The Decoy Effect (DE): The decoy effect (DE) refers to a change in people’s preference between two options, when presented with a third *asymmetrically-dominated* option, i.e., an option which is inferior to one option in all respects, but, in comparison to the other option, it is inferior in some respects and superior in others. In that light, DE is a violation of the independence of irrelevant alternative axiom of rational choice theory (Ray, 1973), which prescribes the following: If A is preferred to B out of the choice set $\{A,B\}$, introducing a third option X, hence expanding the choice set to $\{A,B,X\}$, should not make B preferable to A.

We are now well-positioned to formally present our result.

Proposition 1. *The following holds:*

$$\text{DE} \rightsquigarrow \text{FE}.$$

Proof. According to normative principles, preference for the choice sets $\{A,B\}$ and $\{A,B,X\}$ should be the same, with X being an asymmetrically-dominated option. The rationale is the following: Since X is inferior to one option in *all* respects, rationally X should never be chosen; hence, the preference pattern for the choice sets $\{A,B\}$ and $\{A,B,X\}$ should be identical. Therefore, whenever people's preference pattern for the choice sets $\{A,B\}$ and $\{A,B,X\}$ differs (which is the case for DE), it logically implies the violation of the extensionality principle, hence granting the occurrence of FE. This concludes the proof. ■

The message of Proposition 1 is simple: From the standpoint of normative principles, the two choice sets $\{A,B\}$ and $\{A,B,X\}$ (with X being an asymmetrically-dominated option) are equivalent, therefore people's showing different preference patterns for the two choice sets, as is the case in DE, is a clear indication of FE. Proposition 1, therefore, formally establishes that the occurrence of DE leads to the occurrence of FE.

4.2 Case Study 2: Base-Rate Neglect $\rightsquigarrow$ Conjunction Fallacy

As our second example, we formally establish an IR between another pair of well-documented cognitive fallacies, namely, the base-rate neglect (BRN) and the conjunction fallacy (CF). BRN and CF can be characterized as follows.

The Base-Rate Neglect (BRN): Base-rate neglect (BRN) (Tversky & Kahneman, 1981a) refers to people not considering prior probabilities in their judgments under uncertainty.

The Conjunction Fallacy (CF): For two events A,B and presented with evidence e, people judge the probability of the event $A\cap B$ to be greater than that of A (or B), in isolation. That is, more formally, people judge: $\mathbb{P}(A\cap B|e) > \mathbb{P}(A|e)$. In that light, CF is a clear violation of the axioms of probability (since $\forall A,B,\ A\cap B\subseteq A \Rightarrow \mathbb{P}(A\cap B|e)\leq \mathbb{P}(A|e)\ \forall e\neq\varnothing$; that is, the probability of a subset of Y, in principle, cannot be greater than that of Y).

CF is well captured in the famous Linda experiment by Tversky and Kahneman (1981). Presented with a description (e) of Linda, a politically active, single, outspoken, and very bright 31-year-old female, people overwhelmingly judge that Linda is more likely to be a feminist bankteller ($A\cap B$) than to be a bankteller (A).

We are now well-positioned to formally present our result.

Proposition 2. *The following holds.*

$$\text{BRN} \rightsquigarrow \text{CF}.$$

Proof. Since $\mathbb{P}(A\cap B|e) = \mathbb{P}(e|A\cap B)\mathbb{P}(A\cap B)$ and $\mathbb{P}(A|e) = \mathbb{P}(e|A)\mathbb{P}(A)$, we have:

$$\frac{\mathbb{P}(A\cap B|e)}{\mathbb{P}(A|e)} = \frac{\mathbb{P}(e|A\cap B)}{\mathbb{P}(e|A)}\frac{\mathbb{P}(A\cap B)}{\mathbb{P}(A)},$$

where the term $\frac{\mathbb{P}(A\cap B)}{\mathbb{P}(A)}$ indicates the ratio between priors $\mathbb{P}(A\cap B)$ and $\mathbb{P}(A)$. If BRN occurs (which results in the term $\frac{\mathbb{P}(A\cap B)}{\mathbb{P}(A)}$ being dropped), it follows that:

$$\frac{\mathbb{P}(A\cap B|e)}{\mathbb{P}(A|e)} = \frac{\mathbb{P}(e|A\cap B)}{\mathbb{P}(e|A)}.$$

Assuming that $\mathbb{P}(e|A\cap B) > \mathbb{P}(e|A)$, which is the case in the context of CF (see the Linda experiment discussed above), it follows that:

$$\frac{\mathbb{P}(A\cap B|e)}{\mathbb{P}(A|e)} = \frac{\mathbb{P}(e|A\cap B)}{\mathbb{P}(e|A)} > 1 \Rightarrow \mathbb{P}(A\cap B|e) > \mathbb{P}(A|e),$$

hence CF occurs. This completes the proof. ■

In simple terms, Proposition 2 shows that the occurrence of BRN leads to the occurrence of CF.

4.3 Case Study 3: Allais Paradox $\rightsquigarrow$ Framing Effect

As our third example, we formally establish an IR between the Allais paradox (APX) and FE. APX can be characterized as follows. (See Sec. 4.1 for the characterization of FE.)

The Allais Paradox (APX): The Allais paradox refers to an observed reversal in participants' choices in two different experiments, each of which consists of a choice between two gambles, A and B, while in fact, according to the independence axiom of rational decision-making (Von Neumann & Morgenstern, 1953), no such a reversal should occur. That is, although the independence axiom grants the equivalence of the two experiments, the pattern of people's preference nevertheless reverses from the first experiment to the second.[3]

Proposition 3. *The following holds:*

$$\text{APX} \rightsquigarrow \text{FE}.$$

Proof. The proof is evident from the characterization of APX given above. Although the independence axiom of rational decision-making (Von Neumann & Morgenstern, 1953) grants the equivalence of the two experiments entertained in APX, the pattern of people's preference nevertheless reverses from one to the other. That is, in the case of APX, people produce different responses for two equivalent experiments. Therefore, the occurrence of the Allais paradox logically implies the occurrence of the framing effect. This concludes the proof. ■

4.4 Case Study 4: Availability Bias $\rightsquigarrow$ Conjunction Fallacy

As our final example, we formally establish an IR between the well-documented Availability bias (AvB) and CF. AvB can be concisely characterized as follows: (See Sec. 4.2 for the characterization of CF.)

The Availability Bias (AvB): Extreme events come to mind easily, people overestimate their probabilities, and over-represent them in decision-making (Tversky & Kahneman,

[3] The reader is referred to Allais (1953) for a clear description of the two experiments.

1973; Lieder et al., 2018; Nobandegani et al., 2018). Formally, people overestimate the probability of an event o, $p(o)$, proportional to the absolute value of its subjective utility $u(o)$ (Lieder et al., 2018; Bordalo, Gennaioli, & Shleifer, 2012). That is, people's subjective probability of event o, $q(o)$, is given by[4] $q(o) \propto p(o)|u(o)|$.

Proposition 4. *Let o_1 and o_2 be two events, and let $o_\wedge$ denote the event corresponding to the occurrence of o_1 and o_2 together, i.e., the one corresponding to the conjunction of the two events o_1 and o_2. Assuming that $\forall i = 1,2,\ |u(o_\wedge)| \gg |u(o_i)|$, the following holds:*

$$\text{AvB} \rightsquigarrow \text{CF}.$$

Proof. According to the characterization of AvB given above, $q(o_\wedge) \propto p(o_\wedge)|u(o_\wedge)|$ and $\forall i = 1,2,\ q(o_i) \propto p(o_i)|u(o_i)|$. We have,

$$\forall i = 1,2, \quad \frac{q(o_\wedge)}{q(o_i)} = \frac{p(o_\wedge)}{p(o_i)} \frac{|u(o_\wedge)|}{|u(o_i)|}.$$

It follows from the axioms of probability that $\forall i = 1,2,\ p(o_\wedge) \leq p(o_i)$; hence, $\forall i = 1,2,\ \frac{p(o_\wedge)}{p(o_i)} \leq 1$. However, since $\forall i = 1,2,\ |u(o_\wedge)| \gg |u(o_i)|$, it follows that $\frac{|u(o_\wedge)|}{|u(o_i)|} \gg 1,\ \forall i = 1,2$. Therefore, altogether, $\forall i = 1,2,\ \frac{q(o_\wedge)}{q(o_i)} > 1$ which implies $\forall i = 1,2,\ q(o_\wedge) > q(o_i)$, granting the validity of the conjunction fallacy (CF). This concludes the proof. ■

The message of Proposition 4 is simple. If people judge the conjunction of two events to be much more extreme than each of them individually (i.e., $\forall i = 1,2,\ |u(o_\wedge)| \gg |u(o_i)|$), then the occurrence of AvB leads to the occurrence of CF.

5 General Discussion

In this work, we introduce the notion of implication relation (IR) between a pair of cognitive fallacies, formally characterizing how one would logically imply the other.

A crucial initial step in establishing IRs between cognitive fallacies is to provide a characterization of the cognitive fallacies involved in those IRs, i.e., to specify, for each cognitive fallacy, what instances and/or circumstances belong to the class of that cognitive fallacy. In Sec. 4, we first provide a characterization of the cognitive fallacies of interest, followed by formally establishing IRs. Particularly, we provide a broad characterization of the cognitive fallacies we are interested in, with those characterizations being primarily guided by experimental findings. As such, these characterizations could be arguably made more precise and/or broadened as future research deepens our understanding of the cognitive fallacies involved. Accordingly, we see the characterizations provided in the current study as work in progress and, very likely, subject to change.

A closer examination of Propositions 1 to 4 and their proofs reveals that IRs can be categorized into two broad types: *logical-IRs* (denoted by $\overset{l}{\rightsquigarrow}$) and *causal-IRs* (denoted by $\overset{c}{\rightsquigarrow}$). Establishing a logical-IR, $\overset{l}{\rightsquigarrow}$, from a fallacy F_1 to another fallacy F_2 implies that F_1 is a special case of F_2, with every instance of F_1 being an instance of F_2. For example, a closer examination of Proposition 1 and its proof reveals that DE is a special case of FE, with every instance of DE being an instance of FE in disguise. The same understanding holds for Proposition 3 and its proof, indicating that APX is simply a special case of FE, with every instance of APX being an instance of FE in disguise. Hence, using our newly introduced notation: DE $\overset{l}{\rightsquigarrow}$ FE and APX $\overset{l}{\rightsquigarrow}$ FE. Establishing a causal-IR, $\overset{c}{\rightsquigarrow}$, from a fallacy F_1 to another fallacy F_2 implies that the occurrence of F_1 brings about (i.e., causes) the occurrence of F_2. For example, a closer examination of Proposition 2 and its proof reveals that the occurrence of BRN brings about the occurrence of CF, i.e., there is a cause-effect relationship between BRN and CF, with BRN being the cause and CF the effect. The same understanding holds for Proposition 4 and its proof, indicating that the occurrence of AvB brings about the occurrence of CF, i.e., there is a cause-effect relationship between AvB and CF, with AvB being the cause and CF the effect. Hence, using our newly introduced notation: BRN $\overset{c}{\rightsquigarrow}$ CF and AvB $\overset{c}{\rightsquigarrow}$ CF. Drawing further on the analogy between IR and reduction in computational complexity, it is worth noting that there also exist several types of reduction in computational complexity, namely, Karp's reduction, Cook's reduction, truth-table reduction, L-reduction, A-reduction, P-reduction, E-reduction, AP-reduction, PTAS-reduction, etc.

Importantly, logical-IRs and causal-IRs have quite different implications. If $F_1 \overset{l}{\rightsquigarrow} F_2$ holds (implying that F_1 is a special case of F_2 as discussed above), it then follows that a *complete* account of F_2 should also account for F_1, and, in that sense, accounting for F_2 is more demanding[5] than accounting for F_1. For example, since DE is a spacial case of FE (see Proposition 1 and its proof), that is, DE is nothing but FE in disguise, any complete account for FE inevitably should also account for DE, implying that accounting for FE is more demanding than accounting solely for a special case of FE, DE. However, if $F_1 \overset{c}{\rightsquigarrow} F_2$ holds (implying that the occurrence of F_1 brings about F_2), it then follows that an account of F_1 naturally serves as an account of F_2 due to the following rationale: If X causes F_1, and F_1 causes F_2, it then follows that X causes F_2, with F_1 serving as a mediator. In that light, establishing causal-IRs between various cognitive biases/fallacies has an intriguing implication: For any chain of causal-IRs $F_1 \overset{c}{\rightsquigarrow} F_2 \overset{c}{\rightsquigarrow} F_3 \overset{c}{\rightsquigarrow} \ldots \overset{c}{\rightsquigarrow} F_{n-1} \overset{c}{\rightsquigarrow} F_n$, any mechanistic account

[4] We must emphasize that our establishing of the IR between AvB and CF only depends on the broad assumption that the more extreme an event is, the more people overestimate its probability, and holds for any $q(o)$ which satisfies this condition, e.g., $q(o) \propto p(o)|u(o)|\sqrt{\frac{1+|u(o)|\sqrt{s}}{|u(o)|\sqrt{s}}}$ (Nobandegani et al., 2018). Therefore, the assumption $q(o) \propto p(o)|u(o)|$ made in the characterization of AvB is only one choice out of infinitely-many possibilities satisfying the said condition, and hence, is not necessary.

[5] Accounting for F_2 is "more demanding" than for F_1, as a complete account of F_2 would necessarily have to explain a wider range of cases, including all instances of F_1 as a subset.

of F_i naturally serves as an account of $F_{i+1}, F_{i+2}, \ldots, F_n$. For example, since the occurrence of BRN causes the occurrence of CF (see Proposition 2 and its proof), it then follows that any mechanistic account of BRN naturally serves as an account of CF, with BRN serving as a mediator. This understanding has a very intriguing implications for studies of cognitive fallacies in general: Establishing a chain of causal-IRs $F_1 \overset{c}{\rightsquigarrow} F_2 \overset{c}{\rightsquigarrow} F_3 \overset{c}{\rightsquigarrow} \ldots \overset{c}{\rightsquigarrow} F_{n-1} \overset{c}{\rightsquigarrow} F_n$, clearly reveals which of the fallacies $F_1, \cdots, F_n$ is more pivotal or fundamental to account for; the answer is of course the left-most fallacy in the chain, i.e., F_1. This strongly suggests that, directing efforts toward finding a comprehensive, satisfying account of F_1 would be the most rewarding research agenda, because, thanks to the established chain of causal-IRs, we would get a set of comprehensive, satisfying accounts of all $F_2, F_3, \cdots, F_n$ for free! Therefore, identifying IRs could systematize and guide a research agenda, with a huge increase in research efficiency.

Suppose we have established a causal IR between two biases A and B (i.e., $A \overset{c}{\rightsquigarrow} B$). Here is a question worth considering. (Q1) Does a mechanistic account of A also serve as a mechanistic account of B? As we argue above, it does. But it is crucial to note that this is just a theoretical possibility. That is, upon empirical investigations (e.g., using advanced neuroimaging techniques), we may come to realize that the mechanistic underpinnings of B, after all, have nothing to do with that of A. Just because some process model can simulate B does not necessarily imply that that process model is *the* cognitive process responsible for the occurrence of B in the brain. Thus, identifiability remains an issue.

Another question worth considering is the following. (Q2) Assuming we have established $A \overset{c}{\rightsquigarrow} B$, is a participant who commits bias A more likely to commit bias B? The answer to this question is a bit subtle, and is related to our elaboration on (Q1) presented above. If the occurrence of A is indeed what mechanistically drives the occurrence of B (through mechanisms specified in our establishing of the IR between A and B), then the answer to (Q2) is positive. Otherwise, solely based on the fact that we have theoretically established $A \overset{c}{\rightsquigarrow} B$, no decisive answer can be given to (Q2), as there is no *real* mechanistic connection between A and B. Note that just because we have theoretically shown $A \overset{c}{\rightsquigarrow} B$ (i.e., A *can* bring about B, hence a purely theoretical possibility), it does not necessarily imply that A *does* bring about B in reality—the latter claim can be only shown empirically.

Proposition 4, establishing AvB $\rightsquigarrow$ CF, demonstrates an interesting possibility wherein, under a set of auxiliary assumptions (e.g. $\forall i = 1, 2,\ |u(o_\wedge)| \gg |u(o_i)|$ in this case), an IR can be established between two fallacies. The idea of establishing IRs under a set of assumptions widens the applicability of the notion of IR, allowing it to link together pairs of cognitive fallacies that would have little connections unless further assumptions are invoked. Drawing again on the analogy between IR and reduction in computational complexity, it is worth noting that in establishing reductions it is common practice to evoke various assumptions/constraints on the characterization of computational problems (e.g. 3-SAT instead of SAT) and/or on the forms of reductions themselves (e.g. *polynomial-time* reductions or *linear-time* reductions). Importantly, these auxiliary assumptions should be empirically confirmed, motivating new and exciting experimental avenues of research. Empirical confirmations of such auxiliary assumptions, empirically justifies the validity of invoking such assumptions. Importantly, empirical disconfirmation of such assumptions, of course, discredit the said established IR, inviting attempts for establishing other IRs (in the hope that they would survive empirical tests), or for invoking other empirically validated assumptions which would save the established IR, motivating new theoretical and empirical work.

In this work, as a proof-of-concept, we establish IRs between several well-documented cognitive biases; see Fig.2(b). Future work should investigate the possibility of establishing IRs between a wider range cognitive biases/fallacies, with the ultimate goal of developing a principled, comprehensive map of cognitive biases/fallacies, broadly resembling what is shown in Fig. 2(a) in the context of computational complexity. As it is conceivable, and in our view very likely, that a single mechanism would act as the common cause of several biases, that mechanism would then serve as a common parent node (in the yet-to-be-developed comprehensive map of biases) having those biases as children. As such, ultimately, the comprehensive map of biases would have (at least) two types of nodes, one to denote biases and one to denote mechanisms.

While many questions remain open, and much work is left to be done in this direction, we hope to have made some progress toward systematically bringing order to the cognitive fallacy zoo. We see our work as a first step in this direction.

Acknowledgments This work is supported by an operating grant to TRS from the Natural Sciences and Engineering Research Council of Canada.

References

Allais, M. (1953). Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'ecole americaine. *Econometrica*, *21*(4), 503-546. Retrieved from http://www.jstor.org/stable/1907921

Arora, S., & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press.

Bourgeois-Gironde, S., & Giraud, R. (2009). Framing effects as violations of extensionality. *Theory and Decision*, *67*(4), 385–404.

Huber, J., Payne, J. W., & Puto, C. (1982). Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of Consumer Research*, *9*(1), 90–98.

Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

Kahneman, D., & Tversky, A. (1984). Choices, values, and frames. *American Psychologist*, 341–350.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85–103). Springer.

Lieder, F., Griffiths, T. L., & Hsu, M. (2018). Overrepresentation of extreme events in decision making reflects rational use of cognitive resources. *Psychological Review*.

Nobandegani, A. S., da Silva Castanheira, K., Otto, A. R., & Shultz, T. R. (2018). Over-representation of extreme events in decision-making: A rational metacognitive account. In *Proceedings of the* 40^{th} *Annual Meeting of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Papadimitriou, C. H. (2003). *Computational Complexity*. John Wiley and Sons Ltd.

Ray, P. (1973). Independence of irrelevant alternatives. *Econometrica: Journal of the Econometric Society*, 987–991.

Simon, H. A. (1972). Theories of bounded rationality. *Decision and Organization*, *1*(1), 161–176.

Sipser, M. (2006). *Introduction to the Theory of Computation* (Vol. 2). Thomson Course Technology Boston.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, *5*(2), 207–232.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*(4157), 1124–1131.

Tversky, A., & Kahneman, D. (1981a). *Evidential impact of base rates* (Tech. Rep.). Stanford University, Department of Psychology.

Tversky, A., & Kahneman, D. (1981b). The framing of decisions and the psychology of choice. *Science*, *211*(4481), 453–458.

Tversky, A., & Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review*, *90*(4), 293.

Modelling alternative strategies for mental rotation

David Peebles (d.peebles@hud.ac.uk)
Department of Psychology, University of Huddersfield
Queensgate, Huddersfield, HD1 3DH, UK

Abstract

I present two models of mental rotation created within the ACT-R theory of cognition, each of which implements one of the two main strategies identified in the literature. A *holistic* strategy rotates mental images as a whole unit whereas *piecemeal* strategy decomposes the mental image into pieces and rotates them individually. Both models provide a close fit to human response time data from a recent study of mental rotation strategies conducted by Khooshabeh, Hegarty, and Shipley (2013). This work provides an account of human mental rotation data and in so doing, tests a new proposal for representing and processing spatial information to model mental imagery in ACT-R.

Keywords: Mental imagery; Mental rotation; ACT-R; Cognitive architectures.

Models of mental imagery

There have been various attempts to provide formal computational accounts of mental imagery phenomena (e.g., Glasgow & Papadias, 1992; Kunda, McGreggor, & Goel, 2013; Tabachneck-Schijf, Leonardo, & Simon, 1997; Just & Carpenter, 1985) and these have often sought to address the issue of whether imagery requires some form of array based representation or can be accomplished by more abstract, amodal representations and processes.

An early and influential cognitive model that combined pixel array based representations and more abstract representations is the CaMeRa model of expert problem solving with multiple representations (Tabachneck-Schijf et al., 1997). A more recent example is a model of problem solving on the Raven's Progressive Matrices test by Kunda et al. (2013) using 2D arrays of grayscale pixels and associated transformation operations.

In recent years there have been a number of attempts to develop computational accounts of mental imagery from within the assumptions and constraints of *cognitive architectures* (e.g., Rosenbloom, 2012; Wintermute, 2012). Cognitive architectures are theories of the core memory and control structures, learning mechanisms, and perception-action processes required for general intelligence and how they are integrated into a "system of systems" to enable human cognition and autonomous, human-level artificial cognitive agents.

The cognitive architecture with one of the most well developed and comprehensive set of representations for spatial reasoning and visual imagery is Soar (Laird, 2012) and its *Spatial/Visual System* (SVS) (Lathrop, Wintermute, & Laird, 2011; Wintermute, 2012). The SVS system contains two layers of representation: a *visual depictive* layer (a bitmap array representation of space and the topological structure of objects), and a *quantitative spatial* layer (an amodal symbolic/numerical representation of objects and their spatial coordinates, location, rotation and scaling)[1]. SVS also contains operations to transform the continuous information in the quantitative spatial layer into symbolic information that can be used by Soar for reasoning. These processes allow Soar agents to perform mental imagery operations that can manipulate the representations and then extract spatial relationships from the modified states.

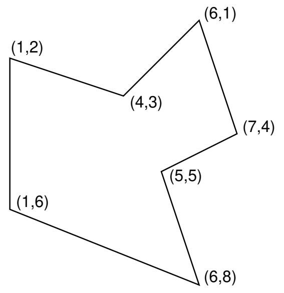
Several proposals have been put forward to endow the ACT-R cognitive architecture (Anderson, 2007) with spatial abilities. For example Gunzelmann and Lyon (2007) outlined an extensive proposal for modelling a range of spatial behaviour (including imagery) by augmenting the architecture with a spatial module and several additional buffers and processes for transforming spatial information. These proposals have, as yet, not been implemented however and so it remains to be seen whether the suggested changes would be able to account for human spatial competence.

An alternative approach to providing ACT-R with spatial capacities is the ACT-R/E project to embody ACT-R in robots (Trafton et al., 2013). ACT-R/E incorporates the *Specialized Egocentrically Coordinated Spaces* (SECS) framework (Trafton & Harrison, 2011; Harrison & Schunn, 2002) which adds modules for three aspects of spatial processing: 2D-retinotopic space, configural space for navigation and localisation, and manipulative space for the region that can be grasped by the robot.

Both of these approaches are broad in the sense that they propose extensive changes to the architecture (i.e., new modules and buffers) and seek to endow ACT-R with a wide range of spatial capabilities related to different spaces (Montello, 1993). Neither approach has modelled spatial imagery however. The aim of the study reported here is to fill this gap by developing ACT-R models of human spatial imagery behaviour. The approach adopted here is more limited and focussed than those discussed above in that it does not propose new modules or buffers but seeks to determine whether the phenomena can be accounted for with only minor adjustments to the existing structures and assumptions of ACT-R.

In the following sections I describe the relevant structures and assumptions of ACT-R and the adaptations required to allow the architecture to model spatial imagery. I then test the approach by using it to develop models of two proposed strategies for mental rotation. Finally I discuss the implications, strengths and weakness of the approach and consider further applications.

[1] In the current (9.6.0) version of Soar, the visual depictive level has been omitted from SVS.



(a)

```
VISUAL:    POLYGON0-0 [POLYGON0]
   POLYGON0-0
   SCREEN-POS    POLYGON-FEATURE8-0
   VALUE         "poly4"
   COLOR         WHITE
   HEIGHT        8
   WIDTH         8
   POINTS        ((1 2) (4 3) (6 1) (7 4) (5 5) (6 8) (1 6))
   CENTRE-X      4
   CENTRE-Y      4
   REGULAR       FALSE
   SIDES         7
   POLYGON       T
```

(b)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(c)

Figure 1: Components of ACT-R's mental imagery mechanism: (a) explicit representation of object vertex coordinate locations, (b) encoding of the vertex locations in the visual buffer, (c) manipulation of the coordinates via matrix transformations.

An ACT-R approach to mental imagery

The two components of ACT-R most relevant to this work are the *vision* module which allows ACT-R to perceive objects in external task environments and the *imaginal* module which functions as ACT-R's limited capacity working memory store in which information is represented and manipulated during problem solving.

ACT-R's perceptual and motor systems were designed to support interaction with computer-based psychology experiments and therefore typically works within a screen-based 2D coordinate space. ACT-R's visual module doesn't interact with the computer interface directly but via a *visual icon*, an intermediate symbolic representation of the objects in the visual environment. When ACT-R's visual attention is directed towards an object in the visual icon, information about the object enters two buffers: a *visual* buffer containing information about the object's features (type, shape, colour etc.), and a *visual-location* buffer representing the object's coordinate location.

Once information has entered the buffers, it is available for further processing, for example as a cue to retrieve further information from declarative memory or to create a new problem state representation in the imaginal module. Compared to other modules, the imaginal module has a greater degree of flexibility in that, in addition having standard buffer for creating and holding information, it also has an *imaginal-action* buffer to allow the module to be extended with novel capabilities by enabling arbitrary actions to be performed on information in the imaginal buffer. This feature is crucial for modelling mental imagery.

Modifications required to model imagery

Many spatial imagery phenomena involve mental representations of the shape, location, orientation and spatial extent of the imagined objects and a set of processes that are able to transform and compare objects according to these characteristics. While the representational and processing assumptions of ACT-R outlined above impose strict but valuable constraints on methods for modelling mental imagery, in this regard, the discrete symbolic representations of ACT-R's visual module (e.g., shape = 'square') with only one x-y coordinate location for each object are currently inadequate.

In light of this, the approach adopted here augments ACT-R with the addition of a new feature slot in the visual object chunk to represent information regarding the outline shape of environmental objects. This requires objects in the task environment to be defined so that the coordinate locations of their vertices are represented explicitly (see Figure 1a). When ACT-R's visual module attends to an object, the vertex coordinates are encoded (Figure 1b) and then transferred to the imaginal buffer.

The second extension to ACT-R adds the ability to perform various imagery operations (e.g., translation, scanning, scaling, zooming, reflection, rotation and composition functions such as intersection, union and subtraction) using a set of linear and affine matrix transformation functions which act upon the vertex coordinates in the imaginal module via the imaginal-action buffer. For example, to rotate each coordinate counter-clockwise by a particular angle θ, it is multiplied by the transformation matrix shown in Figure 1c.

Mental imagery and mental rotation

Mental imagery plays a crucial role in many aspects of cognition, from problem solving, creativity and scientific discovery to psychological disorders such as post-traumatic stress disorder, social phobia and depression (Kosslyn, Thompson, & Ganis, 2006; Pearson, Deprose, Wallace-Hadrill, Burnett Heyes, & Holmes, 2013). Mental imagery has also been the subject of one of the longest running and fiercest debates in cognitive science (Kosslyn & Pomerantz, 1977; Pylyshyn, 1973; Anderson, 1978; Tye, 2000) and the nature of the mental representations and processes underlying mental imagery is still a subject of contention.

The study of mental rotation has been a cornerstone of research into mental imagery since the original experiments of Shepard and Metzler (1971). In the typical form of the mental rotation task, participants are presented with pairs of similar images, one of which has been rotated around its centre, and then required to decide whether the images are identical or not (Figure 2 shows a widely used stimulus from (Shepard & Metzler, 1971)). The key finding of mental rotation tasks is that RT typically increases monotonically with the degree of angular rotation between the images.

Mental rotation has been studied extensively over the last half century in a wide variety of forms and a range of strategies and underlying processes have been proposed. For example, some have suggested that mental rotation is carried out using a *holistic* strategy in which the rotated figure is mentally manipulated as a single, whole unit (e.g., Shepard & Metzler, 1971; Cooper, 1975). Others have argued that rotated figures are subdivided and the component pieces mentally manipulated separately in a *piecemeal* fashion.

The latter strategy was advanced by Just and Carpenter (1976, 1985) who used eye tracking data to support the identification of three distinct stages in the mental rotation task. In the first *search* stage, people look for correspondences between regions of the target and rotated figures in order to select candidate pieces for transformation. In the second *transform and compare* stage, the piece from the rotated image is mentally re-rotated towards its corresponding piece in the target image. Crucially, this process is not a single ballistic rotation but consists of a series of discrete steps in which the mental image is repeatedly manipulated and then compared to the target image to determine whether they are sufficiently congruent to stop.

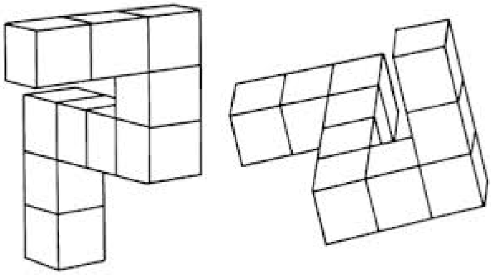


Figure 2: Stimuli used by Shepard and Metzler (1971).

If the second stage is successful and the two pieces are found to be congruent, a third *confirmation* stage is conducted to determine whether the same degree of rotation will also bring other corresponding pieces of the two figures into congruence. This involves a repeat of the three stages until it is judged that the two figures are in fact the same.

In contrast, a holistic strategy involves different stages of processing. The first consists of a process by which representations of—and correspondences between—the two images are constructed. The second consists of a whole-figure rotation process which continues until the two figures are aligned.

In addition to eye movements, response time data are also used to infer the nature of the processes and strategies being employed in mental rotation. A common assumption is that the linear difference in RT between degrees of angular disparity is a function of the rotation processes and that additional time is taken by processes as stimulus encoding, response decision and motor processing (Cooper, 1975; Khooshabeh et al., 2013).

Modelling mental rotation strategies

Human performance In a recent study, Khooshabeh et al. (2013) investigated the behavioural effects of the two rotation strategies by forcing people to use one strategy or the other. They did this by creating fragmented versions of the stimuli shown in Figure 2 (i.e., objects in which some of the blocks had been removed), on the assumption that fragmented stimuli would be harder to rotate holistically.

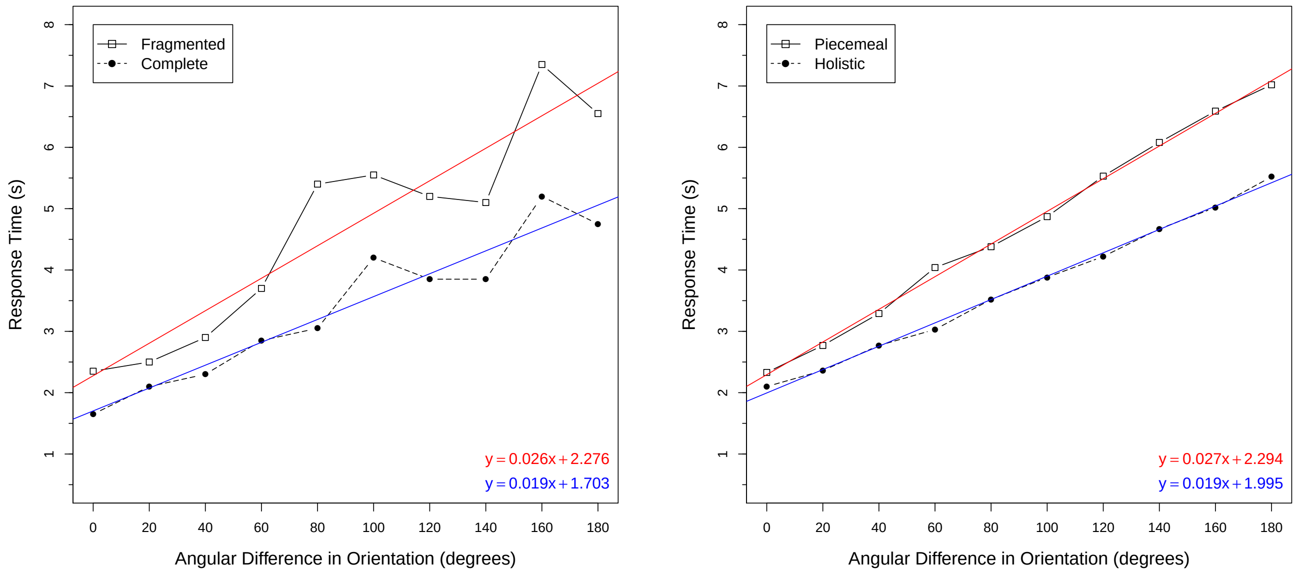
To analyse their data Khooshabeh et al. (2013) classified participants (thirty-eight undergraduate students) as *good* or *poor* imagers according to their degree of accuracy in the task (the categories being defined as approximately the top and bottom thirds of the distribution respectively) and analysing the two groups separately.

This classification is based on previous studies which have led to the claim that piecemeal strategies are favoured by individuals with lower spatial ability whereas those with high spatial ability, because of their greater capacity to build and maintain complete images in working memory, are more likely to use a holistic strategy (e.g., Bethell-Fox & Shepard, 1988; Mumaw, Pellegrino, Kail, & Carter, 1984).

Khooshabeh et al. (2013) predicted therefore that in their experiment, lower spatial ability participants would not differ in their performance for complete and fragmented stimuli (because they use piecemeal strategies for both) whereas those with higher spatial ability would be faster and more accurate with complete figures than for fragmented figures, reflecting the switch from a holistic to a piecemeal strategy. This would be indicated by the slopes of the respective RT functions, with the piecemeal producing a steeper slope than the holistic strategy (Cooper, 1975).

The form of the task was typical, with *target* and *rotated* figures being presented simultaneously side by side on a computer screen. Participants were instructed to judge whether the shapes were the same or different and that their judgement should be based on the overall shape of the two figures, ignoring the missing cubes. Participants were also explicitly told not to respond that the figures were different just because one had missing cubes. After eight practice trials with feedback, participants were given 200 experimental trials (100 control trials in which both figures were complete and 100 trials with one complete figure and one fragmented figure) and RT was recorded from the onset of the stimulus until the participant's key press response. Ten degrees of rotation were used, from 0 to 180 degrees in increments of 20.

Figure 3a presents the RTs for good imagers as a function of angle of rotation and figure type (complete, fragmented) for *same* trials (the typical analysis in mental rotation studies). As predicted, the good imagers were significantly slower in rotating fragmented figures (M = 4601.04 ms, SD = 1944.14) than complete figures ($M = 3260.75$ms, $SD = 1516.09$, $F(1,25) = 25.89$, $p < .001$, $\eta_p{}^2 = .51$) and also had steeper slopes on fragmented ($M = 28.29$ ms/degree, $SD = 17.03$) than complete figures ($M = 20.43$ms/degree, $SD = 5.99$, $F(1,25) = 6.65$, $p = .02$, $\eta_p{}^2 = .21$).



(a) Human data

(b) Data from the ACT-R model

Figure 3: Left: Mean RTs for fragmented and complete stimuli for each angle of rotation, Experiment 1, good spatial imagers, Khooshabeh et al. (2013). Right: Mean RTs for piecemeal and holistic strategies for each angle of rotation, ACT-R model.

Model performance Two ACT-R models of the experiment conducted by Khooshabeh et al. (2013) were created, each implementing one of the two strategies. The holistic and piecemeal strategies implemented by the models are represented as flow charts in Figures 4a and 4b respectively[2].

Both models perform the rotation task according to the incremental *move and test* process described by Just and Carpenter (1976, 1985). The coordinate points representing the rotated image are incrementally rotated counter-clockwise towards the target image by a constant amount (subject to a degree of perceptual error, represented by a random value sampled from a logistic distribution with mean 0 and variance k).

After each rotation step, the angular disparity between current and target coordinate points is reviewed to determine whether they are sufficiently close for the process to stop. This test is a measure of image similarity in that if the points do not coincide then the rotation process will not stop.

The model assumes that RT is determined by the size of the rotation increment, m, taken at each step and the proximity threshold, p regulating the stop decision. The ACT-R *imaginal delay time* parameter, t, which determines the how long a modification request to the imaginal buffer takes to complete was adjusted from its default of .2s. to .1s.

According to the holistic strategy model (implemented by eight production rules), the first stage of the mental rotation task involves a search for correspondences between regions of the target and rotated figures in order to build up a complete, integrated image. When enough pieces of the images have been matched (two in this model), the rotation stage is engaged until the figures are sufficiently aligned, at which point a response is initiated.

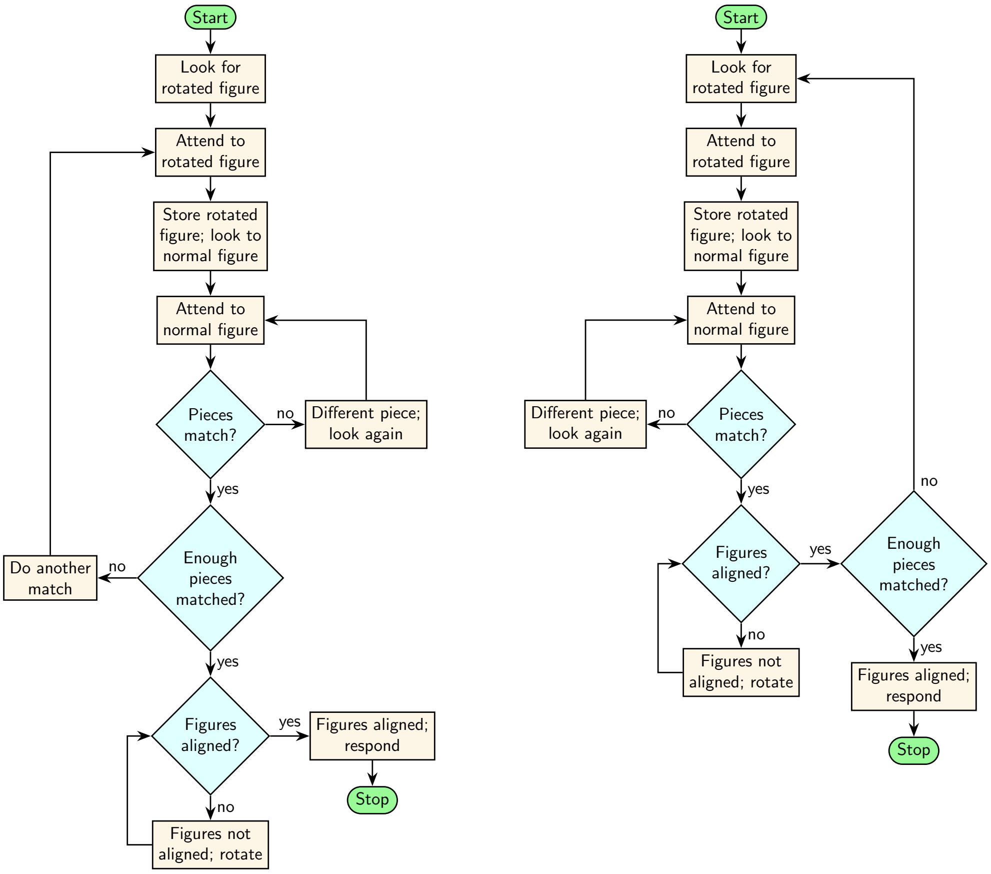
In the piecemeal strategy model (implemented by seven production rules), the first stage of the task involves a search for correspondences between only two regions of the target and rotated figures. Once a piece of the rotated image has been matched to the target image, the rotation stage is engaged until the figures are sufficiently aligned.

When an alignment has occurred, instead of initiating a response, the model repeats the process from the start until enough pieces have been matched. When sufficient pieces have been matched for there to be confidence that the two images are the same (two in this model), a response is initiated.

The piecemeal strategy model has one additional parameter than the holistic model, a separate rotation increment, n for figure pieces subsequent to the first one. This represents the assumption that the rotation of pieces being used to confirm the distance will be faster (i.e., be implemented using bigger step sizes) because the distance is already known.

To test the two models, they were both run 40 times (to simulate 40 participants) for all of the 10 degrees of rotation and the mean RT for each distance computed. Figure 3b shows that both models (with parameters $k = 2$, $m = 8$, $n = 18$, $p = 12$ and $t = 0.1$) provided a close fit to the human data (holistic: $R^2 = .951$, RMSD $= 0.476$; piecemeal: $R^2 = .928$, RMSD $= 0.608$).

[2] Both ACT-R models are available to download from GitHub: https://github.com/djpeebles/act-r-mental-rotation-models.



(a) Holistic strategy

(b) Piecemeal strategy

Figure 4: Control structure of the ACT-R model for a trial of the experiment using (a) the holistic strategy and (b) the piecemeal strategy. Each rectangle corresponds to one production rule in the model.

Discussion

The work described above demonstrates that with only relatively minor modifications and a small number of reasonable assumptions, ACT-R can be applied to develop models of mental imagery phenomena that provide a close match human RT data. Crucially, the modifications are restricted to enabling the representation and transformation of shape information but the new representation and processes integrate with the existing control structures of ACT-R so that the behaviour of the model is primarily a result of the strategy encoded in the production rules (which is essentially the same for both tasks) and the information processing assumptions built into the ACT-R's imaginal module.

The representation of object spatial extent is not at the level of pixel arrays nor at the level of discrete symbols, but at an intermediate numerical level that abstracts from the pixel level. Similarly, the transformation processes incorporated into the architecture are quantitative in nature and are assumed to belong to the wider set of subsymbolic functions that act upon quantitative information in ACT-R at a level closer to the visual system than the qualitative reasoning processes over symbolic representations.

In this regard, the current work represents a modest step towards answering the question concerning the nature of the representations required to support mental imagery discussed in the introduction. The human data modelled here are a useful test of the representations and processes used to adapt the architecture. The models provide an account of the two strategies in terms of where in the task people construct the coherent representations of the figures. In the holistic strategy this is done at the start (which arguably requires greater effort to maintain during rotation) whereas in the piecemeal strategy this is done at the end (which imposes less of a demand on working memory).

Compared to other mental imagery tasks, mental rotation is relatively simple in nature. A more stringent test of the assumptions is necessary therefore and this will come from modelling more challenging tasks, for example the Raven's Progressive Matrices (c.f. Kunda et al., 2013), the *pedestal blocks world* or the *nonholonomic car motion planning* task (Wintermute, 2012) as these will provide richer behavioural data and will require more complex strategies involving a wider range of spatial transformations. This is the plan for the next stage of this project.

Acknowledgements

As always, I thank Dan Bothell for his invaluable advice and endless patience.

References

Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review*, *85*(4), 249.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.

Bethell-Fox, C. E., & Shepard, R. N. (1988). Mental rotation: Effects of stimulus complexity and familiarity. *Journal of Experimental Psychology: Human Perception and Performance*, *14*(1), 12–23.

Borst, J. P., & Anderson, J. R. (2013). Using model-based functional MRI to locate working memory updates and declarative memory retrievals in the fronto-parietal network. *Proceedings of the National Academy of Sciences*, *110*(5), 1628–1633.

Borst, J. P., Nijboer, M., Taatgen, N. A., van Rijn, H., & Anderson, J. R. (2015). Using data-driven model-brain mappings to constrain formal models of cognition. *PLoS One*, *10*(3), e0119673.

Cooper, L. A. (1975). Mental rotation of random two-dimensional shapes. *Cognitive Psychology*, *7*(1), 20–43.

Glasgow, J., & Papadias, D. (1992). Computational imagery. *Cognitive Science*, *16*(3), 355–394.

Gunzelmann, G., & Lyon, D. R. (2007). Mechanisms for human spatial competence. In T. Barkowsky, M. Knauff, G. Ligozat, & D. Montello (Eds.), *Spatial Cognition V: Reasoning, Action, Interaction* (pp. 288–307). Springer-Verlag.

Halverson, T., Gunzelmann, G., Moore, L. R., & Van Dongen, H. P. A. (2010). Modeling the effects of work shift on learning in a mental orientation and rotation task. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th international conference on cognitive modeling* (pp. 79–84).

Harrison, A. M., & Schunn, C. D. (2002). ACT-R/S: A computational and neurologically inspired model of spatial reasoning. In *Proceedings of the 24th annual meeting of the Cognitive Science Society.*

Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive psychology*, *8*(4), 441–480.

Just, M. A., & Carpenter, P. A. (1985). Cognitive coordinate systems: Accounts of mental rotation and individual differences in spatial ability. *Psychological Review*, *92*(2), 137–172.

Khooshabeh, P., Hegarty, M., & Shipley, T. F. (2013). Individual differences in mental rotation. *Experimental Psychology*, *60*(3), 164–171.

Kosslyn, S. M., & Pomerantz, J. R. (1977). Imagery, propositions, and the form of internal representations. *Cognitive Psychology*, *9*(1), 52–76.

Kosslyn, S. M., Thompson, W. L., & Ganis, G. (2006). *The case for mental imagery*. Oxford University Press.

Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, *22*, 47–66.

Laird, J. E. (2012). *The Soar cognitive architecture*. Cambridge, Mass: MIT Press.

Lathrop, S. D., Wintermute, S., & Laird, J. E. (2011). Exploring the functional advantages of spatial and visual cognition from an architectural perspective. *Topics in Cognitive Science*, *3*(4), 796–818.

Milner, D. A., & Goodale, M. A. (1993). Visual pathways to perception and action. *Progress in Brain Research*, *95*, 317–337.

Montello, D. R. (1993). Scale and multiple psychologies of space. In A. U. Frank & I. Campari (Eds.), *Spatial information theory: A theoretical basis for GIS* (pp. 312–321). Berlin: Springer.

Mumaw, R. J., Pellegrino, J. W., Kail, R. V., & Carter, P. (1984). Different slopes for different folks: Process analysis of spatial aptitude. *Memory & Cognition*, *12*(5), 515–521.

Pearson, D. G., Deeprose, C., Wallace-Hadrill, S. M. A., Burnett Heyes, S., & Holmes, E. A. (2013). Assessing mental imagery in clinical psychology: A review of imagery measures and a guiding framework. *Clinical Psychology Review*, *33*(1), 1–23.

Pylyshyn, Z. W. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychological Bulletin*, *80*(1), 1.

Rosenbloom, P. S. (2012). Extending mental imagery in Sigma. In J. Bach, B. Goertzel, & M. Iklé (Eds.), *International conference on artificial general intelligence* (pp. 272–281). Berlin, Heidelberg: Springer.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, *171*(3972), 701–703.

Tabachneck-Schijf, H. J. M., Leonardo, A. M., & Simon, H. A. (1997). CaMeRa: A computational model of multiple representations. *Cognitive Science*, *21*, 305–350.

Trafton, J. G., & Harrison, A. M. (2011). Embodied spatial cognition. *Topics in Cognitive Science*, *3*(4), 686–706.

Trafton, J. G., Hiatt, L., Harrison, A., Tamborello, F., Khemlani, S., & Schultz, A. (2013). ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, *2*(1), 30–55.

Tye, M. (2000). *The imagery debate*. Cambridge, MA: MIT Press.

Ungerleider, L. G., & Mishkin, M. (1982). Two cortical visual systems. In D. J. Ingle, M. A. Goodale, & R. J. W. Mansfield (Eds.), *Analysis of visual behavior* (pp. 549–586). Cambridge, MA: MIT press.

Wintermute, S. (2012). Imagery in cognitive architecture: Representation and control at multiple levels of abstraction. *Cognitive Systems Research*, *19*, 1–29.

An implementation of Universal Spatial Transformative Cognition in ACT-R

Kai Preuss (preuss@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstrasse 23
10587 Berlin, Germany

Leonie Raddatz (leonie.raddatz@tu-berlin.de)
Department of Psychology and Ergonomics, Fasanenstrasse 1
10623 Berlin, Germany

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstrasse 23
10587 Berlin, Germany

Abstract

Mental spatial transformation is usually modeled with highly task-specific approaches, allowing high model accuracy and valid explanations for effects in experimental data. These approaches however suffer from overfitting of models to data, resulting in low general validity. Based on neuro-imaging research suggesting a dedicated cognitive system for mental spatial transformation, a theory for universal spatial transformative cognition and its implementation as an ACT-R module is proposed. This spatial module enables the prediction of processing time for mental spatial operations. Concurrently, a mental folding experiment is conducted to gather participant data for model fitting. Our data confirms an effect of transformation difficulty on reaction times often found in related research, as well as learning effects during the experiment. These results form the foundation for ongoing development of the spatial module, especially regarding the influence of transformation complexity on spatial assessments.

Keywords: spatial cognition; cognitive modeling; ACT-R; mental folding; mental transformation

Introduction

Mental spatial manipulation of objects or scenes is a core mechanism of human cognition. In this regard, understanding an object in three-dimensional space allows us to reason and make assumptions about quality, category, function and other attributes associated with it (Kosslyn, 1996). Although mental spatial transformation is often associated with mental imagery, evidence for the distinctness of the two exists. Spatial representations seem to be separable from mental imagery (Knauff & Johnson-Laird, 2002). A study by Gramann (2013) implies the existence of inter-individual differences in spatial cognition, including the proclivity for an egocentric or allocentric reference frame during mental spatial tasks. Mental spatial processing and mental imagery seem to be situated in separate brain areas, respectively: past research of behavioral and neurophysiological data implies pathways for spatial processing as well as a functional distinction between egocentric and allocentric cognitive systems (Nadel & Hardt, 2004).

Different, partially compatible paradigms for mental spatial transformation have been introduced, each proposing factors for the complexity of a spatial transformation. Shepard and Metzler (1971) studied reaction times for the sameness of two abstract 3D objects, of which one is rotated to a variable degree. A linear relationship between angular disparity and reaction time was found. As objects were only required to be mentally rotated however, the explanatory power of this study for general spatial cognition seems limited. A follow-up study measured reaction time during a task based on cube folding patterns (Shepard & Feng, 1972). In a recent variation of this cube folding paradigm (Wright, Thompson, Ganis, Newcombe, & Kosslyn, 2008), a reference object must be mentally manipulated to assimilate its shape to a target object. Reaction time grew linearly with the folding complexity required by the target object. Additionally, higher complexity levels were reported to be unsolvable within the given time limit by most participants, which suggests an upper limit to spatial transformation capacity.

Lotz and Russwinkel (2016) introduced a decay factor for spatial representations. According to the authors, these decaying representations could only be upheld for a short period of time before they required re-encoding by visual or memory processes. In another variant of mental rotation, a study theorized that non-linear reaction time results are caused by the intricacy of the transformations necessary for a correct response (Neely & Heath, 2010). Based on this theory, higher transformation complexity could be a factor especially in demanding tasks. Other possibilities of complexity measures for spatial processing exist, such as object structure (Bethell-Fox & Shepard, 1988), semantics (Smith & Dror, 2001) or familiarity (Bethell-Fox & Shepard, 1988; Smith & Dror, 2001), and potentially many others. So far, no unequivocal data reasonably demonstrates their effect, but these factors should be kept in mind.

Modeling Spatial Cognition

The cognitive architecture ACT-R (Anderson et al., 2004) consists of modules which represent cognitive systems for e.g. visual, imaginal or motoric processing. Cognitive models rely on the interplay of these modules to simulate specific task behaviors and cognitive processes by exchanging information between buffers associated with each module. This approach also allows for the prediction of brain activity, as the neural representation of each cognitive system can be roughly localized in the human brain (Borst & Anderson, 2015).

While ACT-R offers a unified approach for cognitive modeling of mental imagery (Anderson et al., 2004), similar mechanisms are so far not available for mental spatial transformation. Such cognitive systems and their implementation as a module for ACT-R have been proposed (Gunzelmann & Lyon, 2007), but so far not scientifically validated. In this paper, we seek to formulate a theory on mental spatial transformative cognition, namely how effects shown in studies of spatial cognition can be represented algorithmically, and implement it in the form of an ACT-R module. The goals for the module are:

- **Explainability**: known effects in spatial cognition like growing time costs with growing task complexity, differences in spatial strategies and others should be explained by spatial module functions
- **Universal applicability**: the module should support multiple mental spatial transformation paradigms
- **Validity**: as models are able to refer to a unified implementation of spatial cognition instead of using highly task-specific approaches, the overall validity of modeling spatial processes is improved

One of the challenges of modeling mental manipulation lies in correctly predicting the effect of inter-individual differences, for instance in the proclivity for egocentric or allocentric reference frames (Gunzelmann & Lyon, 2011). The proposed addition to the ACT-R architecture should eventually account for these differences by providing the possibility of multiple approaches to spatial transformation. Additionally, identifying the source of effects like cognitive limitations, time demands, inaccuracies and errors is essential for a sufficiently predictive performance of the module.

As a starting point for the development of the spatial module, we chose to conduct an experiment based on the mental folding task developed by Shepard and Feng (1972), in a variation by Wright et al. (2008) as described above. Concurrently, two cognitive models are developed: one using only default ACT-R modules (the *baseline model*), another incorporating our spatial module (the *enhanced model*). The baseline model will rely on default ACT-R capacities with the goal of achieving as close a fit to human behavioral data as is possible with ACT-R's base mechanisms, while the enhanced model will make use of the spatial module described in this paper. Thus, the baseline model will act as a benchmark - if the addition of a spatial module is indeed a reasonable assumption, the enhanced model should reach a significantly better fit while ideally explaining effects that the baseline model can not.

Hypotheses

We expect our experimental results to show a linear effect of task difficulty on reaction time, as previous studies have shown (Shepard & Feng, 1972). Over the course of the experiment, participants should also show learning effects, resulting in shorter reaction times. The enhanced model should subsequently show an improved fit compared to the baseline model while being more cognitively plausible.

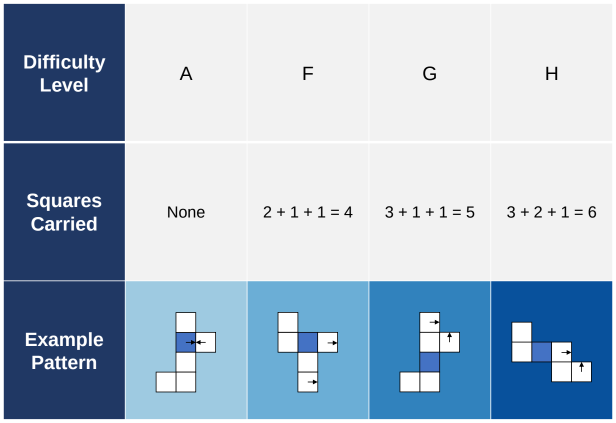
Methods

Mental Folding Study

Participants The study was conducted with 45 participants, of which 5 were excluded due to aberrant error rates, reaction times or technical problems, leaving a sample of 40 participants (20 female, 20 male). All participants were selected according to their orientation strategy measured via the Reference Frame Proclivity Test (Goeke, König, & Gramann, 2013) and completed a pretesting battery prior to the mental folding task. Additionally, data from a 64-channel electroencephalogram (EEG) was collected. Participant selection, pretesting and EEG-Data are no further subject of this paper.

Mental Folding Task A computerized version of the mental folding task originally developed by Shepard and Feng (1972) was created and adjusted into a comparison task similar to the task designed by Wright et al. (2008). The mental folding task consisted of reference figures in the form of semitransparent 3D cubes, and 2D unfolded cube templates as target figures, each with two black arrows on their surfaces and a blue square indicating the base, presented on a black background. Each trial started with a one second presentation of a central fixation cross, followed by the display of a reference figure, either on the left or right side of the screen. Subsequently, after one more second, a target figure appeared on the other respective screen side. The participants were asked to mentally fold the template together and to decide then whether the arrows on reference and target match. Judgements on matching or mismatching arrow positions were recorded via button presses on a response pad. Vertically aligned buttons were used with one button for each judgement type. The experiment consisted of 600 trials, subdivided in five blocks. Participants had to take at least one minute breaks between the blocks and were instructed to always fold upwards, starting from the base. Task completion took 60 minutes on average and each participant passed through 10 minutes of training with feedback in advance.

Stimuli Four levels of difficulty were chosen for the task (see Figure 1). The sum of squares carried (*SSC*) during the series of folds necessary to compare the arrow positions determined the level of difficulty, as defined by Shepard and Feng (1972). The easiest level (A) was a direct visual comparison with arrow tips always meeting. The second level (F) required to carry four, the third level (G) five and the fourth level (H) six squares through the folding sequence. Six different template figures with three arrow variations each (for Levels F, G and H: one variation with arrow tips touching, two with arrow tips in different directions) were constructed



Difficulty Level	A	F	G	H
Squares Carried	None	2 + 1 + 1 = 4	3 + 1 + 1 = 5	3 + 2 + 1 = 6
Example Pattern				

Figure 1: Difficulty levels used in the experiment, based on the classification by Shepard and Feng (1972). Squares carried refers to the amount of squares that need to be transformed to reach an informed decision.

for every level and paired with reference cube figures with either matching or mismatching arrow positions. This resulted in 144 different trials. In order to shorten the length of the experiment to one hour, 24 trials of the mismatch condition were excluded by balanced randomization from each block. Each mismatch stimulus type of each level was shown at least three times over the whole experiment, resulting in 72 match- and 48 mismatch trials per block. The sequence of trials and the presentation sides were randomized in a balanced manner within each block.

Baseline Model

Lacking a spatial module, the baseline model uses memory retrieval as its main mechanism. Spatial structures and results of folding operations are encoded as world knowledge and queried as needed. Cube and folding pattern are visually presented to the model. Arrow directions and base square positions are then saved in a mental representation and used to create folding paths for each arrow on the folding pattern. These paths are then subsequently folded up and the resulting mental images compared to the actual arrow positions and directions on the reference cube. Additionally, a simple instance learning mechanism is implemented, allowing improvement over time.

While the approximation of spatial processes through repeated memory retrieval processes is highly implausible, it represents a reasonable approach using only the standard ACT-R architecture, and thus a benchmark to be improved upon by the enhanced model.

Spatial Module

The spatial module integrates seamlessly into the existing modular structure of ACT-R. Its feature set is chosen with mental rotation and mental folding paradigms in mind, although other applications are possible. In its current version the module supports translation, rotation, scaling and comparison of three-dimensional objects. As it is developed concurrently to subject data acquisition, several design choices are intuitive as of now. Results of upcoming research will be consulted to confirm or improve the proposed module structure.

Structure The module is interfaced by use of its two buffers:

- The *spatial* buffer acts as storage for a mental spatial image of an object, which in turn can be a specific part of a larger object or a group of smaller objects. These objects consist of a three-dimensional representation and optionally of a specific object class, a list of contingently attached objects and a pointer to an origin object, if applicable.

- The *spatial-action* buffer is analogous to the imaginal-action buffer in the way that transformations to the mental representation are handled. It receives and handles transformation requests or queries about the object in the *spatial* buffer.

Point clouds form the structure for three-dimensional representations, as they are versatile and easily transformable through mathematical computation. Each point is formed by xyz-coordinates, allowing objects to be represented with arbitrary level of detail.

Buffer structure and amount were chosen to balance functionality and parsimony - this module setup should allow applicability to all spatial tasks while limiting its complexity and need of resources. This way, in contrast to the approach of Gunzelmann and Lyon (2007), interaction with ACT-R's core module structure is facilitated: as spatial object chunks are standard ACT-R chunks, functionality like object comparison or episodic memory can be achieved through or supported by default modules.

Configurable module parameters are module latency and maximum transformation complexity.

Complexity of Spatial Representations The module observes an upper limit on the number of transformations applicable to the object. If this number is reached, no further transformations will take place and the module will return an error. This limit is an exploratory account of effects showing that for tasks of high difficulty, a jump in reaction time occurs, breaking the linear pattern (Shepard & Feng, 1972). The authors assume that these jumps reflect re-encoding processes - to continue the task, the preliminarily transformed object needs to be harvested from the buffer, memorized and subsequently either recalled from memory or visually re-encoded again. Further research will try to validate this assumption.

At the moment, the upper limit of subsequent object transformations defaults to 4, in line with the instantiation fingers (or finsts) of the declarative and visual modules that work as similar limitations. If a valid transformation is requested on the object in the spatial buffer and the upper limit is not reached, a complexity equation is consulted to compute the

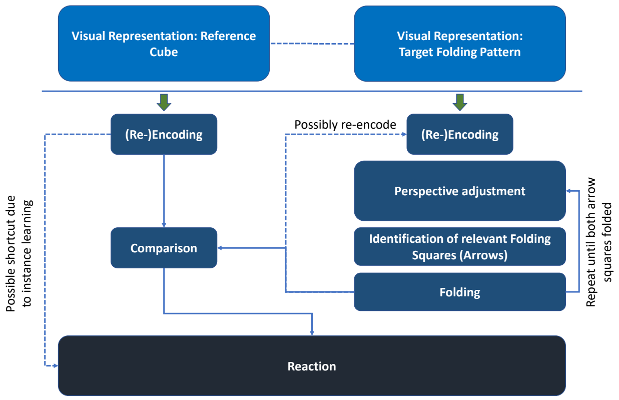

Visual Representation: Reference Cube
Visual Representation: Target Folding Pattern
(Re-)Encoding
Possibly re-encode
(Re-)Encoding
Perspective adjustment
Identification of relevant Folding Squares (Arrows)
Folding
Comparison
Possible shortcut due to instance learning
Repeat until both arrow squares folded
Reaction

Figure 2: A rough outline of the process underlying the enhanced model. Visual representations get encoded, then relevant surfaces of the folding pattern are mentally folded up and subsequently compared to the reference arrow positions and directions.

time delay required for the operation. As data of mental spatial studies (e.g. Shepard and Metzler (1971), Shepard and Feng (1972)) shows a linear relationship between the discrepancy of the spatial object to its required transformation state and reaction times of human participants, a linear equation of the form $f(x) = b + mx$ is used as a basis for the complexity equation. We assume the intercept b is given by ACT-R's default mechanisms such as production firing or the forming of mental representations. The rest of the equation (i.e. factor m) is proposed to be as follows:

$$C_{request} = F \ast M \ast x \ast N^2$$

- F: a latency factor, set as a module parameter. Its default value will be fit based on current experiment data.
- M: a compensation factor used to equalize discrepancies between transformation types like rotation or translation. Potentially depending on the specific function called, this factor equals 1 for now.
- x: the change value for the transformation, i.e. degrees, distance units or others.
- N: the current number of transformations applied to the mental spatial object since it was put into the spatial buffer. This implements research by Neely and Heath (2010), implying that reaction times grow with increasing transformation complexity.

Capabilities of the Module The spatial module is able to translate, rotate and scale spatial objects consisting of point clouds in 3D space. For comparison between two spatial objects, so far two operations are available: A simple comparison is implemented that compares the mean euclidean distance between point pairs from two point clouds. If the point clouds have unequal sizes, the distance from the spare points to the origin substitutes the missing pairs. For instance, an object compared to itself would return a mean euclidean distance of 0, while deviating objects return larger values depending on their scale and significance. Furthermore, a computation of the angle between vectors is implemented to allow for the comparison of e.g. reference and target arrows.

The module offers these tools for modeling mental spatial transformation, however certain task-specific operations like reacting to specific thresholds or perception of the spatial objects still need to be implemented on a model level.

Enhanced Model

An enhanced model for the mental folding task that incorporates the spatial module is currently in development. The underlying process is based on the baseline process model, but instead of memory retrieval processes, spatial information is now processed by the spatial module, which calculates the time needed for each spatial operation based on the above equation. Each square of the folding pattern is now foldable in 3D space, while arrows are represented as direction vectors. Once all relevant surfaces have been folded to their respective cube positions, these direction vectors are compared to the reference arrows and used to form a decision. A process diagram of this model is depicted in Figure 2.

As the enhanced model foregoes memory retrieval processes for spatial operations, it exhibits stronger cognitive plausibility, as forming representations through declarative knowledge is unlikely to occur in spatial problem solving. Additionally, the resulting process model is less rigid and allows for easier backtracking, required for modeling phenomenons like loss of concentration or validation.

Comparison to Experiment Data

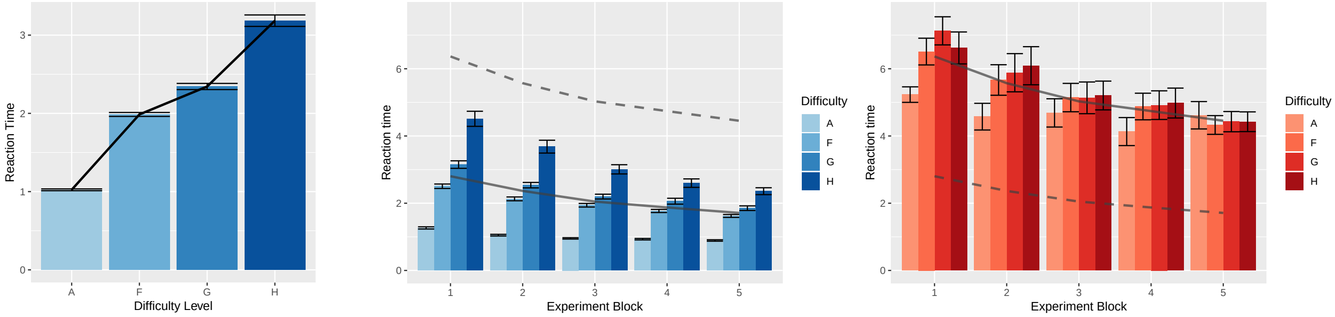
The models are compared to participant data through correlation and root mean square error (RMSE) of averaged reaction times and model output, respectively.

Results

Experiment Data

Behavioral data was analyzed to investigate effects of the factors Difficulty Level and Experiment Block on participant reaction time. Only trials with correct responses were selected for analysis. Trials with reaction times lower or higher than 2 standard deviations from the levels mean within each participant were considered outliers and therefore excluded from further analysis.

A two-way ANOVA with the within-factors Difficulty Level (A, F, G, H) and Experiment Block (1, 2, 3, 4, 5) was conducted on logarithmized reaction times. ANOVA results, adjusted per Greenhouse-Geisser, display significant main and interaction effects of the factors Difficulty Level and Experiment Block on reaction time (Difficulty Level: $F_{1.74,67.96} = 282.86, p < .001$; Experiment Block: $F_{2.16,84.22} = 144.17, p < .001$; Interaction: $F_{5.94,231.84} = 12.57, p < .001$). Reaction times increased with increasing



(a) Effect of Difficulty Level

(b) Participant Reaction Times

(c) Model Reaction Times

Figure 3: Error bars depict 95% confidence interval. **(a)** Reaction times in seconds averaged over participants and blocks, showing the mean effect of difficulty level. Level *A* requires no folding operation. The sums of squares carried necessary during folding are 4 in Level *F*, 5 in Level *G* and 6 in Level *H*, respectively. **(b)** Average participant reaction times in seconds per Difficulty Level and Experiment Block. Black solid line denotes the learning effect, averaged over levels of difficulty. Dashed line shows model learning effect for comparison. **(c)** Model reaction times in seconds per Difficulty Level and Experiment Block. Black solid line denotes learning, averaged over levels. Dashed line shows human learning for comparison.

level of difficulty. Tukey-corrected post-hoc comparisons reveal that the increase in reaction time with increasing level of difficulty is significant in all blocks, with the exception of the difference between difficulty Levels F and G which is only significant in the first two experiment blocks. Overall the ANOVA results seem to imply a learning effect that is especially pronounced for higher difficulty levels. Means and standard deviations are summarized in Table 1. The latency factor parameter of the spatial module was fit based on the averaged reaction times for each difficulty level (see Figure 3a), suggesting a factor of around 0.6 per necessary folding operation.

Model Data

Baseline Model The baseline model output is sufficiently similar to participant data (see Figures 3b and 3c). Due to its mechanisms being based on world knowledge retrieval instead of actual spatial processes, model reaction times are uniformly higher than for participants. While a comparison of model and human reaction times over the factor difficulty showed no significance ($r = .89, p = .11$ with an RMSE of 3.14), a comparison over experiment block showed a high correlation with high significance ($r > .999, p < .001$ with an RMSE of 3.09).

Discussion

Discussion of Results

Both human and model data show a clear improvement over time, correlating highly and showing a learning effect that seems well explained by instance memorization. This suggests an important role of pattern memorization for improvement in spatial tasks.

The effect of task difficulty - as in the sum of squares carried over all necessary folding operations to obtain the correct arrow positions and directions - is clearly pronounced in the experimental data. The baseline model shows a similar influence of the difficulty factor in its output, but shows no correlation to the human data. Interestingly, reaction times for the highest difficulty setting seem to diverge from the linear influence of required folding operations, implying other factors. This might support the aforementioned idea that with more complex mental spatial transformations, re-encoding processes take place (Neely & Heath, 2010).

The data also shows a slight decrease of variance in the reaction times for higher difficulty levels that grows smaller over the course of the experiment (Figure 3b). This variance seems to be within-subject, meaning that solvability of the puzzles in higher difficulties differed strongly for unexperienced solvers, but gradually improved.

Revisiting our original hypotheses, we found a mostly linear effect of task difficulty, with slightly longer reaction times for the highest difficulty level at the start of the experiment than a linear relation would suggest. Learning effects over the course of the experiment in the form of decreasing reaction times were also found. The baseline model showed highly similar learning effects, but remains much slower than human participants and relies on cognitively implausible mechanisms for mental spatial transformation.

Open Questions

The specifics of the spatial module are chosen for simple integration into the existing module structure of ACT-R, its functional requirements and buffer parsimony. These might be challenged by upcoming neurophysiological results of human problem solving in mental folding and rotation tasks. Potential consequences range from showing the existence of multiple systems to a lack of evidence for a dedicated spatial system altogether.

With the claim of modeling universal mental spatial cognition, information from several paradigms needs to be evalu-

Block	Difficulty	Mean	SD
1	A	0.17	0.35
1	F	0.84	0.39
1	G	1.02	0.49
1	H	1.32	0.59
2	A	-0.01	0.33
2	F	0.67	0.40
2	G	0.81	0.47
2	H	1.08	0.63
3	A	-0.10	0.32
3	F	0.58	0.40
3	G	0.66	0.49
3	H	0.91	0.60
4	A	-0.13	0.34
4	F	0.48	0.41
4	G	0.57	0.53
4	H	0.76	0.60
5	A	-0.17	0.34
5	F	0.40	0.41
5	G	0.47	0.51
5	H	0.69	0.55

Table 1: Mean and standard deviation of logarithmized reaction times by Experiment Block and Difficulty Level.

ated and used to fit the spatial module. However, it will still need to be falsifiable - changes to the module need to be done in a way that do not introduce task-specific information, but try to make as few general assumptions necessary to be able to interpret as much spatial processing as possible.

The presented complexity function should work well in the context of mental folding, but its applicability to other spatial paradigms (e.g. non-transformative or non-object-oriented tasks) is still unexplored. While it is based on past research on mental transformation processes, a plethora of amendments or alternatives to the equation is conceivable.

A central issue inherent to the object representation lies in the omission of surface textures. Many paradigms require access to interpretable texture information like arrows, numbers, colors etc. While some features can be encoded as an additional object or point cloud information, this approach is highly restrictive.

The necessity for an equalizing factor for different transformation modalities is unclear. For example, rotation and translation can be reasonably assumed to have different effects on reaction times due to their handling of input as degrees or distance units, respectively. On one hand, a factor specific to the transformation modality could offset this disparity. On the other hand, translation can be interpreted as being based on view angle instead of arbitrary distance units, allowing a closer comparison to rotation. Differences in modalities may not just arise from a disparity in change value however, but from their application difficulty or their neurophysiological basis as well. Additionally, a differentiation between transformations changing the object and those simply changing its perceived orientation could be necessary - a simple rotation seems less resource-intensive than folding parts of an object and subsequently influencing its form or function. In this regard, reference frame proclivity seems especially informative.

Work on the enhanced model is currently ongoing. A challenge remains in finding an optimal ontology for spatial objects, able to represent both internal (e.g. single aspects of an object like cube faces or physical connections between objects) and external (e.g. comparisons of objects or measures of object sameness) relations in and between objects, and adjusting both spatial module and model accordingly.

Outlook

The proposed spatial system is developed in parallel to research into mental spatial transformation. As such, in addition to being subject to change, many details of the implementation are still unclear and highly exploratory. Data from current and future research will aim to provide answers and solutions to these challenges.

On completion, the enhanced model will serve as a first testbed for the spatial module as well as a competitor for the baseline model regarding data fit. Altogether, it forms an important landmark for the validation or falsification of the assumptions laid out in this paper. While spatial reference frame proclivity seems to be an important inter-individual trait for the prediction of performance in mental spatial transformation tasks, it is unclear how spatial processing, and a potential implementation thereof, differs between egocentric and allocentric perspective takers. With the possibility of following distinct neural pathways, a spatial module incorporating this distinction could be comprised of a structure far different than the one described here, e.g. with additional buffers as originally proposed by Gunzelmann and Lyon (2007). This issue will be explored in-depth based on forthcoming imaging data for spatial transformation tasks, with the module being adjusted accordingly.

Upcoming research will incorporate data from a mental rotation study into the spatial module. The additional evaluation of available EEG and eye tracking data for both mental folding and mental rotation will give insight into the functional localization of specific spatial brain functions and improve process models for mental spatial transformations. To enforce the module's universality claim, additional paradigms for spatial tasks like mental scanning or spatial navigation could be investigated in the future.

Acknowledgments

This research is financed through the German Research Foundation (DFG), as part of project #396560184. The authors would like to thank Klaus Gramann for his supervision of the Experiment.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review*, *111*(4), 1036–1060.

Bethell-Fox, C. E., & Shepard, R. N. (1988). Mental rotation: Effects of stimulus complexity and familiarity. *Journal of Experimental Psychology: Human Perception and Performance*, *14*(1), 12.

Borst, J. P., & Anderson, J. R. (2015). Using the ACT-R Cognitive Architecture in Combination With fMRI Data. In *An Introduction to Model-Based Cognitive Neuroscience* (pp. 339–352). New York, NY: Springer New York.

Goeke, C. M., König, P., & Gramann, K. (2013). Different strategies for spatial updating in yaw and pitch path integration. *Frontiers in behavioral neuroscience*, *7*, 5.

Gramann, K. (2013). Embodiment of spatial reference frames and individual differences in reference frame proclivity. *Spatial Cognition & Computation*, *13*(1), 1–25.

Gunzelmann, G., & Lyon, D. R. (2007). Mechanisms for Human Spatial Competence. In T. Barkowsky, M. Knauff, G. Ligozat, & D. R. Montello (Eds.), *Spatial Cognition V Reasoning, Action, Interaction* (pp. 288–307). Springer.

Gunzelmann, G., & Lyon, D. R. (2011). Representations and processes of human spatial competence. *Topics in cognitive science*, *3*(4), 741–759.

Knauff, M., & Johnson-Laird, P. (2002). Visual imagery can impede reasoning. *Memory & cognition*, *30*(3), 363–371.

Kosslyn, S. M. (1996). *Image and brain: The resolution of the imagery debate*. Cambridge, Massachusetts: MIT press.

Lotz, A., & Russwinkel, N. (2016). Modelling different strategies in mental rotation. In T. Barkowsky, Z. F. Llansola, H. Schultheis, & J. van de Ven (Eds.), *Proceedings of the 13th biannual conference of the german cognitive science society* (pp. 35–38). Bremen.

Nadel, L., & Hardt, O. (2004). The spatial brain. *Neuropsychology*, *18*(3), 473.

Neely, K. A., & Heath, M. (2010). Visuomotor mental rotation: Reaction time is determined by the complexity of the sensorimotor transformations mediating the response. *Brain Research*, *1366*, 129–140.

Shepard, R., & Feng, C. (1972). A chronometric study of mental paper folding. *Cognitive Psychology*, *3*(2), 228–243.

Shepard, R., & Metzler, J. (1971). Mental Rotation of Three-Dimensional Objects. *Science*, *171*, 701–703.

Smith, W., & Dror, I. E. (2001). The role of meaning and familiarity in mental transformations. *Psychonomic Bulletin & Review*, *8*(4), 732–741.

Wright, R., Thompson, W. L., Ganis, G., Newcombe, N. S., & Kosslyn, S. M. (2008). Training generalized spatial skills. *Psychonomic Bulletin & Review*, *15*(4), 763–771.

A Meta-Analysis of Conditional Reasoning

Marco Ragni (ragni@cs.uni-freiburg.de)
Cognitive Computation Lab, Technical Faculty, University of Freiburg, Germany

Hannah Dames (dames@cs.uni-freiburg.de)
Cognitive Computation Lab, Technical Faculty, University of Freiburg, Germany

Phil Johnson-Laird (phil@princeton.edu)
Princeton University, Princeton NJ 08540, USA
New York University, New York, NY 10003, USA

Abstract

Conditional premises are assertions with "if", e.g., *If I have measles, then I have fever.* They provide a connection between different propositions and can express causal relations. Conditional inferences often comprise conditional and categorical assertions, e.g., such as modus tollens: *If I have measles, then I have fever*; *I don't have fever*; So, *I don't have measles*. Most research has concerned four sorts of conditional inference, examining them separately. Only a few studies have focused on the patterns over the four sorts of inference (e.g., Oberauer, 2006). Our meta-analysis was of 39 experiments (with 2378 participants) that reported these patterns. It showed that a version of the mental model theory best fits the results when participants produced their own conclusions or evaluated a given conclusion, whereas the suppositional theory provided the best fit when participants chose a conclusion from a list of options.

Keywords: Conditional reasoning; Information; Mental models; Suppositions; Probabilities

Introduction

Conditionals allow humans to describe hypotheses, causal dependencies, diagnoses, and other relations between pieces of information. They tend to be expressed in assertions of the sort, *If A then B*, where *A* and *B* are sensible clauses in natural language, which may be simple or compound, i.e., contain sentential connectives of their own. Classical studies of reasoning use inferences consisting of a conditional and an additional categorical premise, as in:

If he has measles, then he has a fever.	(A conditional)
He has measles.	(A categorical)
What, if anything, follows?	

Almost all reasoners infer: he has a fever (see, e.g., Oberauer, 2006). This sort of inference is the first of four sorts (called modus ponens) as shown below with their conventional names and abbreviations. These four sorts of inferences share a conditional premise, but have different categorical premises and so yield different conclusions. We use '∴' to preface conclusions. For the given conditional *If A then B* and we have the respective categorical premise and conclusion:

A.	∴ B.	(Modus Ponens: MP);
B.	∴ A.	(Affirmation of Consequent: AC);
Not A.	∴ Not B.	(Denial of Antecedent: DA);
Not B.	∴ Not A.	(Modus Tollens: MT).

In classical logic, MP and MT are valid, i.e., given that their premises are true, their conclusions are also true. DA and AC are valid only if the conditional has a biconditional interpretation, equivalent to: *If and only if A then B*. The biconditional inference pattern often occurs in studies (e.g., Oaksford & Chater, 2007, p. 140). While most studies report the response frequencies of the four sorts of inference, they do not give any information about the inference patterns of each participant, such as the number of participants who drew only MP and MT inferences. A few studies, however, do report the frequencies of these inference patterns over the four sorts of premises (e.g., Oberauer, 2006, Barrouillet, Gauffroy, & Lecas, 2008; Evans & Over, 2004). In what follows, we also show that the separate overall frequencies of each of the four inferences yields a misleading picture of the process of reasoning.

Psychologists have proposed five main sorts of theory of conditional reasoning: theories based on formal logic, on mental models, on suppositions, on dual-processes with suppositions, and on probabilities. In what follows, we briefly review them.

Theories based on formal logic (e.g., Rips, 1994) postulate that the mind contains a formal rule for MP but no rule for MT. Thus, its inference depends on the three steps: i) make a supposition of the conditional's *if*-clause, *A*; ii) the rule for MP yields *B*; iii) its conjunction with the categorical premise *not B* is a self-contradiction. As a consequence, one can deny the supposition to yield the conclusion: *not A*. Readers should note that formal rule theories are not included in the meta-analysis, because their processes have never been formulated as multinomial processing trees (see below).

The theory of mental models. The theory of mental models (e.g., Johnson-Laird & Byrne, 2002) postulates two systems of reasoning: intuitive and deliberative. The first system is the intuitive process in which reasoners rely on mental models that represent only what is true. Hence, for the conditional *If A then B* a reasoner forms the following, mental models:

A	B
. . .	

The first model represents the the possibility in which *A*, and thus *B,* both hold. The second model – the ellipsis – stands for the possibility in which *A* is not possible. MP follows at once from these models given the premise *A*. MT, however, does not. It calls for the second process, which is deliberative and in which mental models, including the ellipsis, are fleshed out into fully explicit ones:

A	B
¬ A	¬ B
¬ A	B

This process yields the possibilities in the order above (see, e.g., Barrouillet, Grosset, & Lecas, 2000). The categorical premise, *not B*, now yields the conclusion, *not A*. Hence, MP should be easier than MT. The model theory explains the discrepancy between human reasoning and logically correct inferences as a result of reliance on intuitive mental models. A further relevant prediction is that MT is easier with a biconditional, which has only two fully explicit models, than with a conditional, which has three (Johnson-Laird, Byrne & Schaeken, 1992).

The directional model theory. A variant of the mental model theory – the directional model theory – introduces the assumption that inferences are easier from the *if*-clause to the *then*-clause of a conditional than in the opposite direction (Evans, 1993; Oberauer, 2006). Hence, it follows that an MP-inference is easier than an AC-inference. If the inferences are based on biconditionals, a DA-inference is easier than an MT-inference.

The suppositional theory. The suppositional theory (Evans & Over, 2004) also assumes that two cognitive systems underlie conditional reasoning: a heuristic, automatic, and fast system (1), and an analytical, controlled, and slow system (2). In later versions, the theory assumes that conditionals have a probabilistic interpretation in which there is high conditional probability of the *then*-clause given the *if*-clause. System 1 takes background knowledge, context, and the content of the premises into account. System 2, however, can focus on the information given in the premises and principles of deductive reasoning. Oberauer (2006) formulated two versions of the theory in order to fit data. In the *sequential* version, system 1 operates first and then system 2 generates a conclusion on the basis of this outcome. In the *exclusive* version, only one of the two systems operates on a given problem, i.e., they are mutually exclusive.

The dual process theory of suppositions. There is a family of dual-process theories (see, e.g., Evans, 2008 for a review). However, one prominent version is similar to the suppositional theory (Verschueren, Schaeken, & d'Ydewalle, 2005) because it has the same system 1. But, in this version, system 2 makes inferences using mental models in the same way as the model theory does, instead of the proof-based system in the suppositional theory. The two systems are assumed to be mutually exclusive.

The probabilistic theory. The probabilistic theory shares a general assumption of the suppositional theory, that is, that conditionals are interpreted in terms of subjective conditional probabilities (Oaksford, Chater, & Larkin, 2000). Conditionals have a high conditional probability of the *then*-clause given the *if*-clause. The process for drawing inferences, however, differs from the suppositional theory. Reasoners accept a conclusion based on its subjective conditional probability given the minor premise. This theory was not included in this meta-analysis, because its parameters for MP and MT inferences have only the 'exceptions' parameter (1 – P(*then*-clause | *if*-clause)) in common, which is close to zero. De facto. the theory treats the four inferences as independent, and Oaksford et al. (2000) do not report the frequencies of the patterns of inference (cf. Singmann et al., 2016).

Table 1 summarizes the predictions of the three main sorts of theory. But, as we will see, our meta-analysis was able to examine four theories.

Table 1: Three predictions that discriminate about theories based on logic, suppositions, and mental models.

	Logical	**Suppositional**	**Mental models**
The meaning of *If A then C*:			
1. implies the possibilities: *A C*, *¬A ¬C*, *¬A C*	-	-	+
2. implies that only cases of *A* are relevant to verification	-	+	+
3. implies that MT with a biconditional is easier than with a conditional	-	-	+

Note: + indicates that a theory makes the prediction, and - indicates that it does not.

Prior to the work of Oberauer (2006), theories tended to consider individual sorts of inference, whereas he formalized versions of theories with multinomial processing trees – henceforth, we refer to them as 'trees' – for all 16 possible patterns of responses to the four sorts of inference (MP, AC, DA, and MT). Every reasoner is bound to yield of $2^4 = 16$ possible patterns of responses for the four sorts of inference. These patterns give a more accurate understanding of the cognitive processes underlying conditional reasoning than analyzing the four sorts of inferences separately. As we will see later, the four sorts of inferences are not drawn independently from each other.

Oberauer's trees included all the cognitive processes leading from inputs to the 16 leaves that represented the responses. He added a single fixed guessing component to each of the trees and evaluated the goodness of fit using G-tests. In the following, we use the formulations of Oberauer's (2006) trees for the original model theory, the directional model theory, the suppositional theory (sequential and exclusive), and the dual-process theory (Verschueren et al., 2005).

The main goals of our analyses were (1) to analyze the four sorts of inference in three types of experimental task: the production of conclusions, the choice of conclusions from options, and the evaluations of single given conclusions; and (2) to carry out a new sort of meta-analysis that includes assessments of the reliability of the data, of the inter-dependence of conclusions over the four sorts of inference, and the goodness of fit of the different theories. Finally, the paper discusses the implications of its results for the various theories.

Three Types of Reasoning Task

Studies of conditional reasoning have used three main tasks (for an overview see Schroyens & Schaeken, in preparation). In the *production* task, the participants are given the premises and asked to state what, if anything, follows from them, i.e.., what must be true given that the premises are true. In the *option* task, they are asked to choose such a conclusion from a set of multiple options, which usually include one for "nothing follows". In the *evaluation* task, they are presented with the premises and a single putative conclusion, and they evaluate whether or not it follows from the premises. These three tasks are likely to call on different mental processes, e.g., reasoners can work backwards from a given conclusion in the evaluation task, but they have to formulate or guess a conclusion to carry out the production task (see Schroyens & Schaeken, in preparation). These authors were the first to show that the different sorts of task affect the conclusions that individuals draw (e.g., Schroyens et al., 2001; Schroyens & Schaeken, in preparation). They formulated the following predictions about differences among the three sorts of task: more conjunctive conclusions should occur in the production task, and fewer selections of fallacious conclusions for AC and DA should occur in the option task. We therefore follow Schroyens and Schaeken and conducted separate analyses of performance for the three types of task.

Meta-analysis

The meta-analysis included the data collected and prepared by Schroyens and Schaeken (in preparation). They carried out their own meta-analysis, which included a detailed report of the patterns of inference for the three sorts of task. Their results were from adult participants and high school students in their final year. Furthermore, the studies used abstract conditionals, and other logically equivalent formulations, such as: *all A are B*, *B if A*, *A unless not B*, and *B only if A*, and the biconditional: *if, and only if, A then B*. In addition to these data, the meta-analysis included results reported in Oberauer (2006). We searched the literature in April 2018 on Google Scholar and PubMed. But, none of the other papers that we found reported the frequencies of the 16 different patterns for the four sorts of inference.. Yet, these patterns were essential for our meta-analysis. Thus, in the end, our work relies on results of one study by Klaus Oberauer and 14 studies that Walter Schaeken kindly provided to us (e.g., Barrouillet, Grosset, & Lecas, 2000; Byrne & Tasso, 1999; Evans, Clibbens, & Rood, 1995; Schroyens, Schaeken, & Handley, 2003; and more). In sum, the meta-analysis included data from 39 experiments (from 15 studies) that tested a total of 2378 participants.

MP as the most basic inference pattern

MP is the fundamental inference in conditional reasoning. It is commonplace in everyday life, and most experimental participants make it, though a few failures do occur (see, e.g., Oberauer, 2006). In our view, individuals who do not make MP in an experiment have failed to reason, and so we have excluded their data from our analyses. It therefore focused on the eight patterns of response that include MP.

The Dependency of the Inference Patterns

Some theories of conditional reasoning assume that inferences of the four sorts of inference are independent of one another (Evans & Lynch, 1973). Other theories do so de facto in that they consider only the frequencies of each of the four sorts of inference, not the frequencies of their patterns (e.g., Oaksford et al., 2000). But, are the four sorts of inference independent of one another?

The question is an empirical one, and to examine it we used an algorithm based on Shannon's measure of information, which we used to show that the selections of potential evidence to test a conditional hypothesis are dependent on one another (see Ragni, Kola, & Johnson-Laird, 2018). The intuition motivating the algorithm is simple. Suppose that the inferences in an experiment are more redundant – less informative – than inferences based only on the individual probabilities of each of the four inferences in the experiment. It follows that something is constraining the inferences over and above their independent frequencies. Hence, the selections are dependent. Consequently, if the inferences are dependent, theories implying their independence are wrong. We therefore tested whether the patterns of inference in the experimental data were significantly more redundant (using Shannon's measure) than those of 10,000 simulations of each experiment based on independent selections.

To analyze the potential dependence of the conditional inferences we examined the data for each experiment in our sample following four main steps:

1. Compute *N*, the number of participants, and the probabilities of the eight inference patterns (each including MP) in the set of the participants' inferences.

2. Compute Shannon's entropy H for the experiment.

3. Carry out 10,000 simulated experiments based on the probabilities of making each inference, assigning a pattern to *N* hypothetical participants.

4. Return the number of simulated experiments that were more informative than the actual experiment and the number with the same or lower information values.

Table 2 shows the relative frequencies of the main patterns of inference in our sample of 39 experiments.

Table 2: The relative percentages of five patterns of inference in 39 experiments categorized according to the task (evaluation, option or production of a conclusion). Three of the eight patterns occurred less often than 5% and are not included in the table.

Response pattern	Evaluation	Option	Production	Overall
All inferences	39.0	43.1	50.3	42.3
MP, MT	17.7	21.3	12.4	18.3
MP	14.8	8.4	5.4	10.9
MP, AC	8.4	5.8	11.0	7.8
MP, AC, MT	4.0	8.2	8.5	6.3
Number of Experiments	8	22	9	39
Number of Participants	1103	921	354	2378

Note. The different response patterns indicate whether the MP, DA, AC, MT inferences were accepted, selected, generated or not. We do not present patterns that occurred less than 5% in each task.

Table 3 presents the information value of the 39 experiments investigating the three sorts of task, the mean value of each of their 10,000 simulations, and the results (of Wilcoxon's test comparing the two values) indicating a reliable dependence over the four sorts of inference.

Table 3: The mean information value (in bits, with a theoretical maximum of 3 bits) of 39 experiments using three tasks, their mean information value, and that of sets of 10,000 simulations of each experiment.

Sort of task	Evaluation	Option	Production
Mean information of experiments	1.69	2.03	1.93
Mean information of their simulations	2.05	2.29	2.12
Wilcoxon's *W* and *p*-value	*W* = 3, *p* < .04	*W* = 3, *p* < .04	*W* = 6, *p* = .055

In sum, these results demonstrate that the four sorts of conditional inference in the experiments depend on each other.

An evaluation of theories of conditional reasoning

We evaluated five theories using multinomial processing trees (based on the formulated trees in see Oberauer, 2006; see Figure 1 for the tree of the mental model theory).

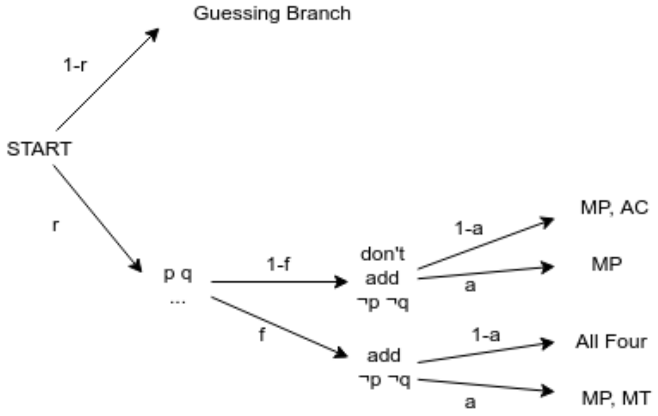


Figure 1: The multinomial processing tree for the mental model theory. The parameter *r* stands for the reasoning part and *1-r* for the guessing part (for an explanation, see Oberauer, 2006). Parameter *f* controls whether or not models are fleshed out to include a model of ¬ p and ¬ q.

Within each tree for a theory, the probability of a particular cognitive state is estimated from the observed frequencies of inferences (Riefer & Batchelder, 1988). We used the maximum-likelihood method from the R-package for multinomial processing trees (MPTinR, Singmann & Kellen, 2012) to fit each theory's tree to the frequencies of the four patterns of inference, separately for the three different types of tasks. To compare the models, we calculated the Bayesian information criterion (BIC), which indicates how much information is lost when a model represents the process that generates the data. This criterion takes into account both a tree's parsimony and its goodness of fit with the data. Thus, the BIC rewards a good fit and punishes a higher number of free parameters. A lower BIC indicates a better theory, because it has fewer parameters or fits the data better, or both. Table 4 presents the BIC for the different trees we fitted for the three different types of conditional tasks (evaluation, option, and production).

The results in Table 4 show that the model theory is the best in accounting for the conclusions that reasoners draw for themselves (the production task). Its directional version is best for the evaluation of given conclusions (the evaluation task). The best theory for the task of selecting a conclusion from a multiple choice (the option task) is the exclusive version of the suppositional theory, which postulates that either system 1 or else system 2 is engaged in the process of reasoning.

Table 4: The fit of the different trees based on the theories of reasoning for the different sorts of task.

The theory	Evaluation (BIC)	Option (BIC)	Production (BIC)
Suppositional-exclusive	42	**43**	45
Dual process suppositions	48	48	42
Directional mental model	**37**	60	42
Mental model	46	62	**38**
Suppositional	63	54	60

Note. 'Evaluation' refers to the evaluation of a given conclusion; 'Option' refers to the choice of a conclusion as an option in a multiple-choice format; and 'Production' refers to the production of a conclusion from a set of premises. A lower Bayesian information criterion (BIC) indicates that a theory has fewer parameters or fits the data better, or both. Best fits are shown in bold.

General Discussion

Following previous research (e.g., Oberauer, 2006; Schaeken, 2001), we carried out a meta-analysis to determine which theory of conditional reasoning and type of tasks gave the best account of individuals' patterns of inference in the four basic sorts of conditional reasoning inferences (modus ponens: MP, affirmation of the consequent: AC, denial of the antecedent: DA, and modus tollens MT). Following Schaeken and Schroyens (in preparation), we separated the studies into those that called for the evaluation of a given conclusion (evaluation task), its selection from a set of options (selection task), and its production from the premises alone (production task) and considered their differences in the analyses.

Our results showed that the most frequent pattern of inferences in all three tasks was to make all four inferences – a pattern that is valid only if the conditionals are interpreted as biconditionals (see Table 2). The next most frequent pattern was to make MP and MT inferences – the two inferences that are valid given a classical conditional interpretation. These results do not discriminate among the various theories of conditional reasoning, though some theories, such as the model theory, predict that AC can occur in the absence of a biconditional interpretation.

An analysis of the amount of information, using Shannon's measure, showed that the patterns of actual inferences were more redundant than those of 10,000 simulations of each experiment (see Table 3). This result corroborated our elimination of any theory in which each selection is in principle independent of the others, e.g., the probabilistic theory of conditional reasoning (Oaksford et al., 2000).

Finally, we fit the multinomial processing trees for each of the remaining five theories to the results of 39 experiments using the three different tasks (Table 4). The Bayesian information criterion, which credits a fewer number of parameters and goodness of fit, showed that the model theory gave the best account of the production of conclusions. Furthermore, its variant that reflects the direction of an inference (the directional model theory) – from *if*-clause to *then*-clause, or vice versa – gave the best account of the evaluation of a given conclusion. We speculate that this result may reflect the order of clauses, i.e., *A C* versus *C A,* in some of the putative conclusions that the participants had to evaluate. In the suppositional theory, which proposes that participants rely either on system 1 or else on system 2 (its exclusive variant), gave the best account of the selection of an option from a set of multiple conclusions.

But why do participants seem to differ in their inference patterns given different types of tasks? There is strong evidence that the response modality of conditional tasks (e.g., scaled or dichotomous response format) affects the way participants process the presented information (e.g., Markovits, Forgues, & Brunet, 2010). These results are consistent with the idea that scaled responses promote a probabilistic mode of processing. Yet, the current studies concentrated on data from non-rating tasks. This constraint was necessary in order to compare the three sorts of task. Future studies should extend the present findings by considering different types of response modality.

By far the most important task for future studies is to formulate tasks in which participants make conditional interpretations. As we mentioned, the most frequent pattern in the present studies was for a biconditional interpretation and so, for now, it is not possible to determine how well the various theories would fit tasks in which the main interpretation is for a conditional and not a biconditional interpretation.

The difference among the theories' fit to the data over the three types of task shows that theories should account for performance in different tasks. Different tasks yield different patterns of inference. One factor, for instance, could be that the need to formulate a conclusion discourages guesswork in comparison with the "option" task in which participants choose a conclusion from a multiple set of possible responses. Future studies should therefore separate different tasks in their analyses of theories.

Acknowledgements

This paper was supported by DFG grants RA 1934/3-1, RA 1934/2-1 and RA 1934/4-1 to MR.

References

Barrouillet, P., Grosset, N., & Lecas, J.-F. (2000). Conditional reasoning by mental models: Chronometric and developmental evidence. *Cognition*, *75*, 237–266.

Byrne, R. M. J., & Tasso, A. (1999). Deductive reasoning with factual, possible, and counterfactual conditionals. *Memory & Cognition*, *27*, 726–740.

Evans, J. S. B., & Lynch, J. S. (1973). Matching bias in the selection task. *British Journal of Psychology*, *64*(3), 391-397.

Evans, J. S. B. T. (1993). The mental model theory of conditional reasoning: Critical appraisal and revision. *Cognition*, 48, 1–20.

Evans J. St.B.T., Clibbens, J. & Rood, B. (1995). Bias in conditional inference: Implications for mental models and mental logic. *The Quarterly Journal of Experimental Psychology Section a*, *48*, 644–670.

Evans, J. S. B. T., & Over, D. E. (2004). *If.* Oxford: Oxford University Press.

Johnson-Laird, P. N., & Byrne, R. M. (2002). Conditionals: A theory of meaning, pragmatics, and inference. *Psychological Review*, *109*(4), 646.

Johnson-Laird, P.N., Byrne, R.M.J., & Schaeken, W.S. (1992). Propositional reasoning by model. *Psychological Review*, *99*, 418-439.

Markovits, H., Forgues, H. L., & Brunet, M. L. (2010). Conditional reasoning, frequency of counterexamples, and the effect of response modality. *Memory & Cognition*, *38*(4), 485-492.

Oaksford, M., Chater, N., & Larkin, J. (2000). Probabilities and polarity biases in conditional inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(4), 883.

Oaksford, M., & Chater, N. (2007). *Bayesian rationality: The probabilistic approach to human reasoning*. Oxford University Press.

Oberauer, K. (2006). Reasoning with conditionals: A test of formal models of four theories. *Cognitive Psychology*, *53*, 238–283.

Ragni, M., Kola, I., & Johnson-Laird, P. N. (2018). On selecting evidence to test hypotheses: A theory of selection tasks. *Psychological Bulletin*, *144*(8), 779.

Riefer, D. M., & Batchelder, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. Psychological Review, 95(3), 318-339.

Rips, L. J. (1994). *The psychology of proof: Deductive reasoning in human thinking.* Cambridge, MA, US: MIT Press.

Schaeken, W., & Schroyens, W. (2000). The effect of explicit negatives and of different contrast classes on conditional syllogisms. *British Journal of Psychology*, *91*, 533–550.

Schaeken, W., Schroyens, W., & Dieussaert, K. (2001). Conditional assertions, tense, and explicit negatives. *European Journal of Cognitive Psychology*, *13*, 433–450.

Schroyens, W. & Schaeken, W. (in preparation). Individual differences and/in models of conditional reasoning by model.

Schroyens, W., Schaeken, W., & d'Ydewalle, G. (2001). *A meta-analytic review of conditional reasoning by model and/or rule: Mental models theory revised.* (Report No. 278). Leuven: University of Leuven, Laboratory of Experimental Psychology.

Schroyens, W., Schaeken, W., Fias, W., & d'Ydewalle, G. (2000). Heuristic and analytic processes in propositional reasoning with negatives. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*, 1713.

Schroyens, W., Schaeken, W., & Handley, S. (2003). In search of counter-examples: Deductive rationality in human reasoning. *The Quarterly Journal of Experimental Psychology Section a*, *56*, 1129–1145.

Singmann, H., & Kellen, D. (2013). MPTinR: Analysis of multinomial processing tree models in R. *Behavior Research Methods*, *45*(2), 560-575.

Verschueren, N., Schaeken, W., & d'Ydewalle, G. (2005). A dual-process specification of causal conditional reasoning. *Thinking & Reasoning*, *11*(3), 239-278.

Weidenfeld, A., Oberauer, K., & Hörnig, R. (2005). Causal and noncausal conditionals: An integrated model of interpretation and reasoning. *The Quarterly Journal of Experimental Psychology Section A*, *58*(8), 1479-1513.

Predicting Individual Spatial Reasoners: A Comparison of Five Cognitive Computational Theories

Marco Ragni and **Paulina Friemann** and **Enver Bakija** and **Novian Habibie**
Yannick Leinhos and **Dennis Pohnke** and **Yvan Satyawan** and **Maya Schöchlin** and **Rabea Turon**
Cognitive Computation Lab, Technical Faculty
Georges-Köhler-Allee, University of Freiburg, 79110 Freiburg, Germany

Abstract

While there is a plethora of cognitive models for spatial relational reasoning, only few of them have been implemented and less have been compared to each other. Additionally, a quantitative benchmark consisting of core spatial relational reasoning problems is missing. And, if empirical data is available it reports aggregated response patterns only, and not the responses of each individual human reasoner. Accordingly, most cognitive models do only aim to explain or reproduce these aggregated response patterns. This paper approaches these issues from a cognitive computational perspective: (1) To establish a first benchmark, we conducted an experiment on reasoning with cardinal direction relations, (2) where necessary, we reimplemented existing cognitive models for spatial relational reasoning, ranging from connectionist approaches to symbolic theories and analyze these theories based on diagnostic criterias, and (3) we evaluated the cognitive models on the benchmark data and extended them where necessary to give predictions for individual reasoners. We discuss implications for theories of spatial reasoning.

Keywords: Spatial Cognition; Reasoning; Cognitive Models; Cardinal Direction; Model Comparison

Introduction

Spatial reasoning is ubiquitous. When we travel, navigate, or communicate about spatial information, we process spatial information and draw inferences. Consider the following problem about cardinal directions:

(A1) The tower is north of the city.
The city is north-west of the mountain.
Where is the mountain in relation to the tower?

A human reasoner may quickly conclude that the answer to this problem is 'The mountain is south-east of the tower'. However, a cognitive model, as you will see later, would only predict about 64% of the responses of individual reasoners correctly, as inter-individual differences are present. Predictions can become even more difficult:

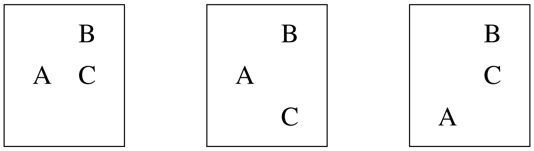
(A2) The train station is north-west of the library.
The library is south-east of the church.
Where is the church in relation to the train station?

This spatial description is ambiguous, or *indeterminate*, i.e., there are several arrangements possible. Hence, it has no straightforward answer – all relations are possible. When looking at the aggregated data, a correct prediction is almost impossible, since the answers are, with a few exceptions, nearly uniformly distributed. Therefore, to better understand the cognitive processes behind the integration of spatial information, we chose to compare various cognitive models of spatial relational reasoning on their performance on modeling the responses of individual participants.

The remainder of the paper is structured as follows: In the next section we briefly report the conducted experiment on reasoning with cardinal directions and the framework which was used for the comparative quantitative evaluation on individual empirical data. We will then introduce the cognitive models we (re-)implemented and extended for the experimental task. Lastly, implications for the domain of spatial reasoning are discussed.

Human Reasoning with Cardinal Directions

As a benchmark for spatial relational reasoning, we conducted an experiment about transitive inferences with cardinal direction relations in the line of Ragni and Becker (2010). As already mentioned, spatial descriptions can be determinate or indeterminate. In the case of oppositional directions (as in problem (A1)), this is easy to see. However, also other descriptions can be indeterminate, depending on the interpretation. The description 'A is south-west of B. B is north of C.' can lead to various representations including the following, depending on the assumed interpretation of the distances conveyed by the relation:



According to various studies (e.g., Knauff, Rauh, & Schlieder, 1995; Ragni & Knauff, 2013), humans do not simply construct all possible models, but have often a preference for one specific representation - the *preferred mental model*. Only if prompted to search for other models, for instance by the experimental task, will humans consider the other possibilities. Preferences for reasoning with cardinal directions have been investigated in Schultheis, Bertel, and Barkowsky (2014) and Ragni and Becker (2010). A preference for arrangements obeying a prototypical triangle shape were found (Schultheis et al., 2014) as well as a preference for main cardinal directions, e.g., 'north' over 'north-west' (Ragni & Becker, 2010).

	NW	**N**	**NE**	**E**	**SE**	**S**	**SW**	**W**
NW	[SE] NE	[SE] SE	[SE,S,SW] S 0,763	[SE,S,SW] S 0.737	[ALL] E, W 0.195	[NE,E,SE] E 0.903	[NE,E,SE] E 0.844	[SE] SE
N	[SE] SE	[S] S	[SW] SW	[SW] SW	[NW,W,SW] W 0.806	[N,S] S 0.529	[NE,E,S] E 0.857	[SE] SE
NE	[SE,S,SW] S 0.829	[SW] SW	[SW] SW	[SW] SW	[NW,W,SW] W 0.871	[NW,W,SW] W 0.935	ALL E, NW	[SE,S,SW] S 0.758
E	[SE,S,SW] S 0.81	[SW] SW	[SW] SW	[W] W	[NW] NW	[NW] NW	[NE,N,NW] N 0.829	[E,W] E, W 0.5
SE	[ALL] E, SE, W 0.915	[NW,W,SW] W 0.929	[NW,W,SW] W 0.818	[NW] NW	[NW] NW	[NW] NW	[NE,N,NW] N 0.765	[NE,N,NW] N 0.841
S	[NE,E,SE] E 0.920	[N,S] S 0.553	[NW,W,SW] W 0.844	[NW] NW	[NW] NW	[N] N	[NE] NE	[NE] NE
SW	[NE,E,SE] E 0.858	[NE,E,SE] E 0.853	[ALL] E 0.366	[NE,N,NW] N 0.833	[NE,N,NW] N 0.770	[NE] NE	[NE] NE	[NE] NE
W	[SE] SE	[SE] SE	[SE,S,SW] S 0.818	[E,W] E 0.541	[NE,N,NW] N 0.816	[NE] NE	[NE] NE	[E] E

Figure 1: Response preferences for the 64 problems of the cardinal direction experiment. The row represents the relation in the first premise (e.g., A is NW of B), the column the respective relation in the second premise (e.g., B is NE of C). In each cell the first row depicts the logically valid relations, the second row the most frequently chosen answer. In the indeterminate case, the third row contains the relative frequency of the preferred relation.

Method

We tested 49 participants in a web experiment on Amazon's Mechanical Turk. In the main part, participants were presented with 64 spatial reasoning problems with cardinal directions. All problems were of the form 'A r_1 B. B r_2 C.' with each r_1 and r_2 being one of the 8 cardinal direction relations *north*, *north-east*, *east*, *south-east*, *south*, *south-west*, *west*, and *north-west*. Instead of A, B, and C different buildings based on their frequency in the English language were used. The task of the participants was to give a relation that holds between C and A. The premises were presented sequentially in a self-paced procedure. The order of the problems was randomized separately for every participant. Participants responded by pressing the respective key/s (e.g., "nw" for north-west).

Based on previously defined exclusion criteria — more than two fast guessing responses (RTs $<$ 0.2s), more than two wrong responses to standard problems (i.e., North-North, South-South, West-West, East-East) and telling that they wrote down the premises or drew pictures on paper — eight participants were excluded. Thus, the final sample size was $N = 41$ participants.

Results

Overall, 84.0 % of the problems were solved correctly, i.e., participants gave a valid answer. In the indeterminate cases preferences can be observed. These are depicted in Figure 1.

Evaluating Models on the Individual Reasoner

The Cognitive Computation for Behavioral Reasoning Analysis (CCOBRA) framework[1] is a benchmarking tool implemented in Python. Its goal is to test models and how good these simulate the experimental procedure of individual participants. The models are presented with the same task in the same sequence with the same response options. By providing precise responses to individual tasks, models are evaluated based on their predictive accuracy.

Models are allowed to train on a data set consisting of tasks and related human responses of individuals not present in the evaluation data. In the test phase, the models are presented with novel empirical data on which they are to give a prediction regarding the conclusion drawn by the current participant. Additionally, after predicting the response to a task, they are presented with the true response and thus allowed to adapt to an individual participant. Hence, CCOBRA extends the traditional cognitive modeling problem by moving beyond the level of aggregates. As a result, the modeling problem gets harder, but the outcomes can be interpreted more intuitively. Higher predictive scores correspond directly to a better grasp of the processes underlying an individual human reasoner's cognitive system.

We divided the gathered empirical data into a training and a test set: One third of the participants were randomly assigned to the training set, and the other participants were assigned to the test set.

[1] `https://github.com/CognitiveComputationLab/ccobra`

Five Cognitive Models for Spatial Reasoning

Model Selection Criteria

Models were selected with respect to the following selection criteria, of which every model reported here fulfills at least two: (i) the cognitive model is developed, or easily extendable, for human reasoning with cardinal directions, (ii) the model already has an implemented version or is easily implementable, (iii) the model makes a prediction concerning complexity of task, (iv) the model is a stand-alone implementations, (v) the model offers explanations for basal principles of spatial reasoning. We identified six cognitive models for spatial reasoning existent in the literature and contacted the respective authors. The six models were: The Spatial Probabilistic Model (Ragni & Becker, 2010), Verbal Spatial Reasoning Model (Krumnack, Bucher, Nejasmic, & Knauff, 2010), the spatial architecture CASIMIR (Schultheis & Barkowsky, 2011), the Spatial Artificial Neural Network (Ragni & Klein, 2012), PRISM (Ragni & Knauff, 2013), the Dynamic Field Theory (DFT) (Kounatidou, Richter, & Schöner, 2018). The spatial architecture CASIMIR (Schultheis & Barkowsky, 2011) was not available and due to its size and dependence between long-term memory and reasoning processes it was not possible to reimplement it. In the following, we briefly report the models.

The Spatial Probabilistic Model (Ragni & Becker, 2010)

We reimplemented the spatial probabilistic model developed by Ragni and Becker (2010).

The Unit Layout Model. This model is used as a heuristic for calculating detours by computing the conditional probability of relations between different locations (Ragni & Becker, 2010). It is represented as a lookup table that contains every possible direction relation between R_1 and R_2. Example of one unit layout lookup table can be seen in Figure 2.

Gains. For representing some cognitive phenomena, gains were added to certain probabilities. E.g., the given data shows that participants prefer the direction *east* over the direction *west* if they have the choice between them. In this example the model adds a certain value (usually the value is optimized in the pre-train-function) to the probability $p(\text{“}east\text{”})$ and normalizes all probabilities (here for all directions) afterwards.

Implementation Details. Let B' be the set of cardinal relations, where each of them represents an applicable relation (in this case, direction). Given three locations a,b,c, the relations between them are stated as $R_1,R_2,R_3 \in B'$ which are applied as aR_1b, bR_1c, and aR_3c. The relative frequency of R_3 for R_1,R_2 (called $f_{R_1,R_2}(R_3)$) is parametrized in probability distribution $P_{R_1,R_2}(R_3)$. The preferred relation is then:

$$M(R_1,R_2) = \operatorname*{argmax}_{R_3 \in B} P_{R_1,R_2}(R_3) \tag{1}$$

SE-NW	S-NW	SW-NW	SW-N	SW-NE
E-NW	*a*	W-NW	W-N	W-NE
NE-NW	N-NW	NW-NW	NW-N	NW-NE
NE-W	N-W	NW-W	*c*	NW-E
NE-SE	N-SE	NW-SE	NW-S	NW-SE

Figure 2: The unit layout for R_3 = NW. Field a is to the north-west of c. All other field are uniquely labeled with relations $R_1 - R_2$. It holds for each of them that field a is R_1-wards of it and it is R_2-wards of c (Ragni & Becker, 2010).

and by using Bayes Rule in equation (Ragni & Becker, 2010), it becomes:

$$P_{R_1,R_2}(R_3) := P(R_3|R_1,R_2) = \frac{P(R_1,R_2|R_3)P(R_3)}{P(R_1,R_2)} \tag{2}$$

where $P(R_1,R_2)$ is assumed to have a unit distribution and $P(R_3)$ has a unit distribution with a probability gain for the main cardinals and gain towards the east. These gains are motivated by the given data and are added to the respective probabilities of the directions. After adding the gains, it is necessary to normalize the probabilities. As mentioned, calculation of $P(R_1,R_2|R_3)$ is done using the unit layout's lookup table that contains every possible direction relation between R_1 and R_2.

$$P(R_1,R_2|R_3) = \frac{c^{-1}_{R_1,R_1}}{\sum_{R'_1,R'_2 \in} c^{-1}_{R'_1,R'_2}} \tag{3}$$

with the cost function (Ragni & Becker, 2010):

$$c^{R_3}_{R_1,R_2} := \frac{d([a]^{R_3},[R_1,R_2]^{R_3}) + d([R_1,R_2]^{R_3}),[c]^{R_3}}{d([a]^{R_3},[c]^{R_3})} \tag{4}$$

Verbal Spatial Reasoning Model (Krumnack et al., 2010)

Verbal reasoning is based on the assumption that the human mind constructs a verbal representation of a problem, and the reasoning process then uses mechanisms similar to those of language processing to draw or validate a conclusion as proposed by Polk and Newell (1995).

The parameter-free verbal model (Krumnack et al., 2010; Krumnack, Bucher, Nejasmic, Nebel, & Knauff, 2011) suggests that individuals construct a queue of object terms in their mind that can be read like a sentence. A mental cost metric determines where a new object is inserted. It assumes that breaking links between objects costs more than creating new links, and searching for an object is more efficient in the direction of the queue. This direction is determined upon insertion of the first relation and depends on a cultural left-right preference (Maass & Russo, 2003).

Extension of the Model. The model by Krumnack et al. has been developed for one-dimensional spatial relational problems only. Hence, we expanded the model for cardinal directions, while keeping the structure of the queue. This is done by adding a direction encoding for the vertical and horizontal plane to each link, with positive values for "north" and "east", and negative ones for "south" and "west". If the angle between the direction of the new relation and the queue direction is more than 90°, the new object is inserted before the reference object, otherwise at the end of the queue. Problem (A1) generates the following queue:

$$\begin{array}{lccccc} & \textit{tower} & \rightarrow & \textit{city} & \rightarrow & \textit{mountain} \\ \textit{vertical} & -1 & & -1 & & 0 \\ \textit{horizontal} & 0 & & 1 & & 0 \end{array} \qquad (5)$$

To predict a response the model sums up all the direction encodings between the two objects in the queue and decodes them into cardinal directions. E.g., in the queue above, the model would sum up all the direction instructions from "tower" to "mountain", receiving a negative total in the vertical and a positive one in the horizontal plane, which means that one must go "south" and "east" to reach the "mountain" starting from the "tower". This results in "the tower is northwest of the mountain". Individual adaption of the queue direction was implemented to account for (cultural) preference.

The Spatial Artificial Neural Network (Ragni & Klein, 2012)

We adapted the implementation of an Artificial Neural Network (ANN) (e.g., Zurada, 1992) for spatial relational reasoning with cardinal directions (Ragni & Klein, 2012) to work with the CCOBRA framework to predict individual subjects' responses to spatial relational problems.

Implementation Details. As in Ragni and Klein (2012), we used one hidden layer and a full connectivity between the layers, and trained the network with backpropagation (Rumelhart, Hinton, & Williams, 1986). The network is based on calculations on point algebra, and treats x- and y-directions independently. The network is hence tested twice on each premise-pair. First, the x-direction (west-east) is calculated, and second, the y-direction (north-south). Consequently, the network consists of 2 input nodes. Three output nodes semantically describe the spatial relation between the first and the last object on the tested axis.

All parameters were tuned manually to maximize correctness on the limited data provided. Learning rate and momentum factor were tuned in 0.1 steps in a range of 0 to 1. A value of 0.1 for both parameters yielded the best results.

To train the network, we performed 10 iterations on the training set with each of the eight possible response choices respectively.

Lastly, to perform a prediction on individual participants in the test set, the given task is given to the network, again separately for x- and y-direction. The highest valued response is returned as prediction.

PRISM (Ragni & Knauff, 2013)

We re-implemented the PRISM model, an implementation of the preferred mental model theory (Ragni & Knauff, 2013). It simulates the construction of preferred mental models and can vary this preferred model to find alternative conclusions. A spatial working memory structure is operationalized by a spatial array. In short, it consists of a mental array and a spatial focus which inserts tokens into the array, inspects the array to find new spatial relations, and relocates tokens in the array to generate alternative models of the problem description, if necessary. The focus also introduces a general measure of difficulty based on the number of necessary focus operations (rather than the number of models).

Dynamic Field Theory (DFT) (Kounatidou et al., 2018)

Kounatidou et al. (2018) proposed a cognitive model to solve the preference effect for relations right or left based on the Dynamic Field Theory (DFT) (Schöner, Spencer, & the DFT Research Group, 2015). The architecture can be divided into five functional parts. The first part involves discrete conceptual nodes for the input premises whose activation is translated into continuous activation in later fields. The second functional part is the attention part which forms peaks of activation for objects that are currently attended. The third part is the scene representation in which the spatial scene as well as the color of the objects in the scene is stored. The fourth part is concerned with spatial transformations that put the objects in the correct relation according to the given premise. And the final part is concerned with the organization of all the involved processes, including starting processes, checking if processes are finished and resetting activation to their resting state after all processes of a premise are completed.

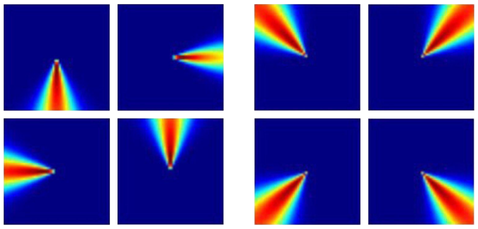


Figure 3: Cardinal direction spatial activation templates can be seen on the left, along with the new extended inter-cardinal templates on the right.

Extension of the Model. The original architecture of Kounatidou et al. (2018) can only create two-dimensional scene representations with the four cardinal relations (north, south, east, and west) between the objects. However, the benchmark data of the Cardinal Direction Experimentin-

Table 1: Overview of the evaluated models for spatial relational reasoning.

Cognitive Model	Cognitive complexity measure	Assumptions about WM representation	Predictions of phenomena	Generalizable to other relations	Accuracy predicting individuals
Verbal Reasoning	Yes	Minimal mental model	Yes	Yes	64%
Bayesian	No	None	No	No	64%
PRISM	Yes	Minimal mental model	Yes	Yes	63%
ANN	No	None	No	Yes	63%
DFT	No	Open	Yes	Yes	62%

Note. Cognitive complexity measure refers to whether the model gives an explicit account of the difficulty of a reasoning problem. *Assumptions about WM representation* refers to whether the models make any assumptions about the human working memory. *Predictions of empirical phenomena* refers to whether the models make new empirically testable predictions. *Generalizable* means whether the model can be extended to other spatial relations. *Accuracy predicting individuals* denotes the percentage of correctly predicted answers for the individual participants.

cludes inter-cardinal directions. Therefore, we extended the architecture with new concept nodes and corresponding spatial activation templates for the inter-cardinal directions. For these, we took the existing spatial relation templates and performed a component multiplication and normalization operation on them. For example, to get north-east, we took the product of a component multiplication between the north and east spatial activation templates and normalized it, such that the peaks were equivalent to those found in the cardinal directions. The resulting templates can be seen in Figure 3.

The implementation of this architecture was done within the CEDAR framework (Lomp, Zibner, Richter, Rano, & Schöner, 2013), which provides a way to create models based on dynamic neural fields. However, it was not possible to connect the framework to CCOBRA's evaluation function. Therefore, evaluation was performed by hand, which was possible because the model is deterministic, i.e., generates the same output for each participant.

Results and Discussion

If we just consider the accuracy to predict each individual reasoner, then we see that we reach about 64% of the predictions (see Table 1). The probabilistic approach and the Verbal Reasoning model performed the best. However, overall accuracy was very comparable for all models. Considering each single participant (see Figure 4), the different models reach a prediction accuracy of up to 90%. So a first result is: Though the models have been developed for predicting the most frequent answer, the prediction rate for many individual participants is high. It seems that the aggregate responses do capture general cognitive processes.

But the models differ in some theoretical aspects: Some do make predictions about the difficulty of problems and are *process models*, e.g., the Verbal Reasoning model and PRISM, and they do predict which symbolic mental representation is built. While this is not necessarily reflected in the accuracy, it allows to make *predictions on a phenomenological level*, i.e., the model can generate predictions about new phenomena that can be tested. This is specifically a limitation of the current version of the Spatial ANN and the Bayesian approach. They can fit the data, but testable novel predictions cannot be drawn. A further point is that models may not be too restricted to some specific spatial relations. Here extensions of the specific Bayesian approach is not straightforward.

Limitations of the approaches. The Verbal Reasoning model performed relatively well in the task it was built for: linear orderings in one dimension. However, as of yet it is not able to predict instances where a reasoner gives a logically incorrect answer. In the future, the assumptions of this cognitive model should be tested more rigorously. It could be possible that the introduction of an individual mental cost threshold would solve the problem of giving incorrect solutions. Possibilities for individual adaptation have to be explored further, since the paths used here did not improve performance. The implementation of the Neural Dynamic Field Architecture brings some limitation with it. These include the inability to rearrange existing objects in its spatial memory, to place a new premise if the reference point does not exist in its spatial memory yet, even if the target does exist, the inability to adapt to new information and the limited size of the spatial memory. The model is only able to append objects at the end of each cardinal direction (e.g., left-most position with regards to west or top-most position with regards to north). It is unable to insert objects in between two existing objects. If the architecture produces a response that is incorrect to the information or to a specific individual, it is unable to adapt or be trained specifically to respond differently. Moreover, all parameters are hard-coded and must be manually tuned.

Conclusion

The current state of the art demonstrates that it can fit about 64% of the data. The models vary to a great extend, but are very similar in their predictions. More research is necessary to understand the mechanics of human reasoning for such a simple task as transitive inferences in cardinal directions.

References

Knauff, M., Rauh, R., & Schlieder, C. (1995). Preferred mental models in qualitative spatial reasoning: A cognitive

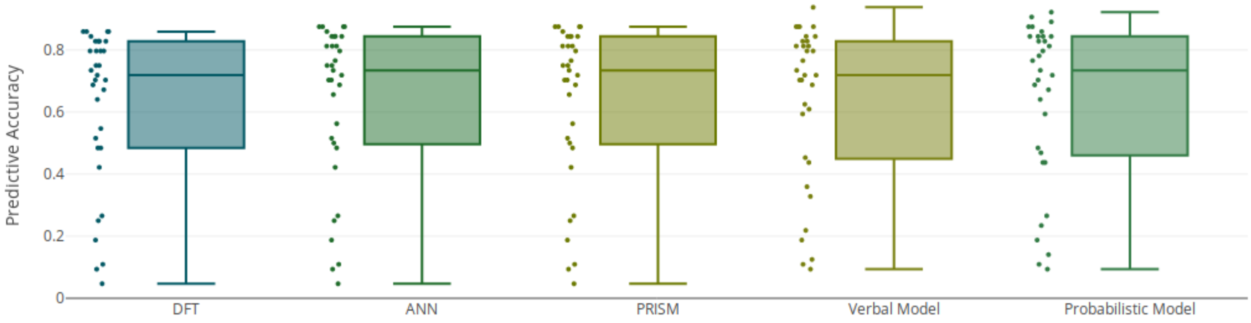


Figure 4: Percentage of correct predictions for each individual participant (dots) and for the population (box plots).

assessment of Allen's calculus. In J. Moore & J. Lehman (Eds.), *Proceedings of the 17th Annual Conference of the Cognitive Science Society* (pp. 200–205).

Kounatidou, P., Richter, M., & Schöner, G. (2018). A neural dynamic architecture that autonomously builds mental models. In C. Kalish, M. A. Rau, X. J. Zhu, & T. T. Rogers (Eds.), *Proceedings of the 40th Annual Meeting of the Cognitive Science Society.*

Krumnack, A., Bucher, L., Nejasmic, J., & Knauff, M. (2010). Spatial reasoning as verbal reasoning. In S. Ohlsson (Ed.), *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society.*

Krumnack, A., Bucher, L., Nejasmic, J., Nebel, B., & Knauff, M. (2011). A model for relational reasoning as verbal reasoning. *Cognitive Systems Research*, *12*, 377–392.

Lomp, O., Zibner, S., Richter, M., Rano, I., & Schöner, G. (2013). A Software Framework for Cognition, Embodiment, Dynamics, and Autonomy in Robotics: Cedar. In *Artificial Neural Networks and Machine Learning – ICANN 2013* (Vol. 8131, p. 475-482). Springer Berlin Heidelberg.

Maass, A., & Russo, A. (2003). Directional bias in the mental representation of spatial events: Nature or culture? *Psychological Science*, *14*, 296–301.

Polk, T. A., & Newell, A. (1995). Deduction as verbal reasoning. *Psychological Review*, *102*, 533.

Ragni, M., & Becker, B. (2010). Preferences in cardinal direction. In S. Ohlsson (Ed.), *Proceedings of the 32nd annual meeting of the cognitive science society.*

Ragni, M., & Klein, A. (2012). Deductive reasoning - using artificial neural networks to simulate preferential reasoning. In J. Filipe & J. Kacprzyk (Eds.), *Proceedings of the 4th International Joint Conference on Computational Intelligence* (pp. 635–638).

Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Review*, *120(3)*, 561-588.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. doi: 10.1038/323533a0

Schöner, G., Spencer, J. P., & the DFT Research Group. (2015). *Dynamic thinking: A primer on dynamic field theory*. Oxford University Press.

Schultheis, H., & Barkowsky, T. (2011). Casimir: An Architecture for Mental Spatial Knowledge Processing. *Topics in Cognitive Science*, *3*(4), 778–795.

Schultheis, H., Bertel, S., & Barkowsky, T. (2014). Modeling Mental Spatial Reasoning About Cardinal Directions. *Cognitive Science*, *38*(8), 1521–1561.

Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*. West Publishing Company St. Paul.

SpotLight on Dynamics of Individual Learning

Roussel Rahman (rahmar2@rpi.edu) &
Wayne D. Gray (grayw@rpi.edu)
Cognitive Science Department, Rensselaer Polytechnic Institute
Troy, NY 12180 USA

Abstract

How does one person learn a complex task? Averaging performance over a group of individuals implicitly assumes that there is only one set of methods for accomplishing the task and that all learners acquire those methods in the same sequence. We maintain that the *average subject* is a mythical beast that does not exist. Hence, rather than profiling a mythical "average subject", we use *SpotLight* – a tool for analyzing changes in individual performance as skill is acquired in a complex task. Specifically, in this report, SpotLight uses 35 features and measures (some collected at millisecond level, others collected once per game), to investigate the skill acquisition of 9 players each of whom spent 31 hours learning the complex task of Space Fortress (SF). SpotLight enables us to uncover the evolution of individual strategies and the iterative efforts of individuals to create, discover, and explore new ways to improve their current performance. To our surprise, these players seem to have followed a common 'design for the weakest link' rule, in which after the current weakest link was strengthened a player's attention turned to the next weakest link. While this rule served our performers well, an often imposed constraint on the rule – 'while retaining existing strengths' – sometimes led the odd performer to suboptimal plateaus.

Keywords: Individual Learning; Part-task; Plateaus; Dips; Leaps; PDL; SpotLight; Relative entropy; Power Law

Introduction

Studies of skill acquisition often proceed by averaging data collected over large groups of individuals. Such methods are fine if we wish to measure the average effect of a treatment administered in different ways, but they fail to achieve our goal of understanding how individuals acquire complex skills.

Here we adopt the *Plateaus, Dips, and Leaps* (PDLs) approach advocated by Gray and Lindstedt (2017) and use the *SpotLight* tool (Rahman & Gray, in preparation) which enables us to identify the PDLs in individual performance. Our results show some commonalities in individual strategies amidst vast differences. For example, after varying numbers of hours of practice, all players adopted an optimal, but effort-expensive strategy. However, the most striking commonalities are not in the gameplay strategies *per se*, but the ways and means in which these strategies were modified. Specifically, we observe that the changes in each player's gameplay strategies pivoted around part-tasks in which the player was performing well. Relative to these pivots, gameplay strategies were recurrently modified to address the weakest parts of gameplay. Based on these findings, we propose that task execution strategies were recurrently updated by a common *optimize strategies for the weakest links* rule.

By focusing on one subgoal at a time, this rule provides checkpoints towards devising optimal strategies for the whole task. However, an excessive focus on subgoals may lead performers to lose sight of the overall goal (maximize total returns) and adopt strategies that maximize the current subgoal at the expense of the overall goal. For example, as we elaborate later, our worst performing player, in attempting to reinforce his skills in one subtask, adopted a suboptimal strategy that basically contradicted the entire point of the game and led to a plateau of stable suboptimal performance (Fu & Gray, 2004) from which there was no path forward. His best alternative would have been to discard the results of a long branch of exploration and strategy development, and revert to a much earlier set of strategies; which he did not do and fell victim to a type of sunk cost fallacy (Sweis et al., 2018).

Learning in Simple vs Complex Tasks

The research literature seems bifurcated between simple and complex tasks. For simple tasks, the power law of practice does a fine job of modeling learning (Newell & Rosenbloom, 1981). Those who follow in this tradition have complicated the world a bit (very reasonably) by proposing revisions that incorporate strategy-specific power laws (Rickard, 1997; Delaney, Reder, Staszewski, & Ritter, n.d.; Donner & Hardy, 2015). However, the more complex the task, the more the number of subtasks, and/or the more alternative ways of implementing a subtask, the less we would expect one individual's choices to resemble another's.

A complex task encompasses a hierarchy of subtasks (Simon, 1962), where higher level subtasks consist of and serve as goals for lower level ones. Task complexity exceeds the sum of separate subtasks because of intermediate associations in the hierarchy, which also implies a number of alternative routes in the hierarchy to reach from bottom to top. Consequently, even if practice alone suffices to maximize performance for simple tasks, more complex tasks require identifying optimum strategies from many alternative strategies.

The number of subtasks and the sets of possible strategies for each subtask raises new questions as to how learning progresses with practice. Does the individual attempt to optimize all parts of the task? Considering the limited cognitive and physical resources available to performers, it is reasonable to expect at least some parts to be *satisficed* (Simon, 1947). How choices are made as to which parts are satisficed or optimized and how such choices affect the goals and ultimately performance, are questions that directly relate to the dynamics of individual learning. To explore answers to these questions, we put the SpotLight on individual performance in the complex task of SF (Mané & Donchin, 1989).

The SpotLight Tool

The SpotLight tool (Rahman & Gray, in preparation) reveals changes in the execution of individual tasks and subtasks by detecting the PDLs in individual performance. The tool is instrumented with relative entropy (denoted by *RE* in Equation 1), an information-theoretic measure of the difference between a target probability distribution (p_i) and a reference distribution (p_r); in other words, it measures the difference between two states of uncertainty (Vedral, 2002). Whereas the scope of comparison in relative entropy is limited to two distributions, the scope is extended in the SpotLight to a finite number of distributions (see Equation 1). First, longitudinal records of performance (univariate or multivariate) are discretized into n consecutive phases and converted into phase-specific probability distributions. Then, a stable phase (i.e., its corresponding distribution) of performance is set as the common reference, relative to which relative entropy of each target distribution in each phase is calculated. Therefore, the output from the SpotLight is a relative entropy curve consisting of n points. This way, information of systematic changes in performance is retained in the relative entropy curve as differences from the stable reference. For details and demonstrations of the SpotLight, please refer to Rahman and Gray (in preparation).

$$RE(p_i||p_r) = \int_X p_i(x)\ \log_2\left(\frac{p_i(x)}{p_r(x)}\right)\ dx \qquad (1)$$

Indices of targets: $i = 1, 2, 3, ..., n$

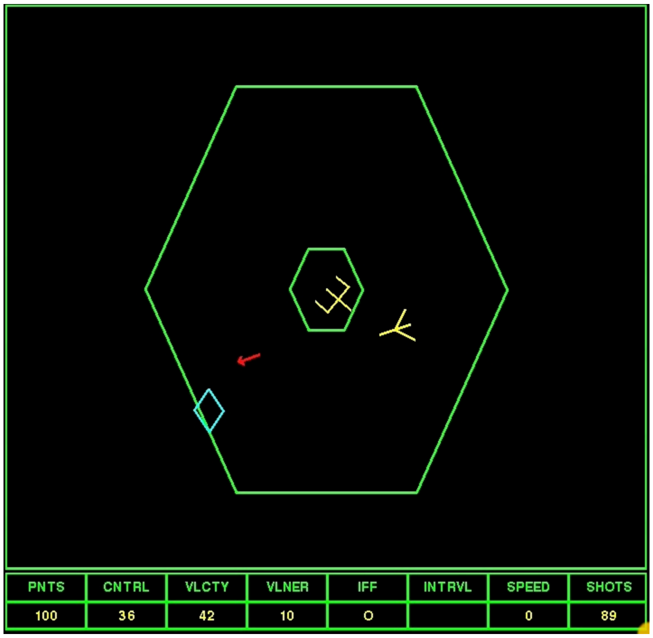
Index of common reference: $r\ (1 \leq r \leq n)$

SpotLighting at Different Levels of Granularity

In the relative entropy curve (e.g., in Figures 2 and 4), general improvement of performance with practice is captured by a continual decrease of relative entropy, and the periods of PDLs are identifiable as exceptions from this general trend. Specifically, during plateaus – periods of non-improvement with practice – relative entropy remains steady; during dips, relative entropy temporarily increases; and during leaps of performance, relative entropy sharply drops. Because the SpotLight models performance recorded by any measure through a single variable, relative entropy, individual performance in a complex task can be compared and investigated across levels of granularity. Therefore, a strategy change affecting the higher-level measures of performance (e.g., the Total score in SF) can be investigated further in lower levels (e.g., number of Fortress kills, use of resources, spatial locations of player's ship) to identify the subtasks associated with the strategy change.

Relative Entropy versus More Common Measures

To explain the choice of relative entropy over other more common measures (e.g., moving average, cumulative sum or coefficient of variation), its relativity property mitigates random noise from analysis (Rahman & Gray, in preparation). That is, random noise present in both the target and the reference distributions is eliminated in relative entropy. Moreover, relative entropy compares entire probability distributions, enabling more efficient use of the information present in the data. Finally, the probabilistic approach also enables future works in other rigorous frameworks (e.g., Bayesian updating or Kolmogorov equations for stochastic processes) to explore evolution of probability distributions with individual learning.



PNTS	CNTRL	VLCTY	VLNER	IFF	INTRVL	SPEED	SHOTS
100	36	42	10	O		0	89

Figure 1: Space Fortress 4 (Destefano, 2010). Screenshot showing the Space Fortress in the center, the player's OS having recently fired a missile (red) at a mine (blue diamond).

Space Fortress: A Complex Task

Each game of SF lasts 5 minutes, where the player battles the Fortress. The player flies a ship ('Own Ship' or OS) carrying a limited number of missiles in a frictionless environment (Figure 1). The Fortress, fixed at the center, can rotate to shoot at OS. The mines (minions of the Fortress) periodically spawn to home in on OS. The mines are of two types which are only distinguishable by a three-letter code shown once at the start of each game. After a necessary identification step, one missile hit kills a mine. Contrastingly, killing the Fortress has two steps. First, 10 missile strikes make it vulnerable (with an inter-strike interval > 250 ms, failure to maintain the intervals results in full recovery); then, a double-strike (with an inter-strike interval < 250 ms) kills it. Conversely, OS is destroyed after 4 hits from either the Fortress and/or the mines. After being destroyed, OS or the Fortress immediately regenerates and the battle continues. At random intervals, the player receives opportunities to choose between receiving a bonus of 100 points or 50 missiles. The time to notice and to act to receive the bonus is limited. If OS' arsenal is empty, the player can gain more missiles at the cost of 3 points for each one.

The objective of the game is to maximize the Total score, consisting of four subscores – Points, Speed, Control and Velocity – measuring performance in different subtasks. In turn, each subscore consists of even lower-level measures (e.g., speed of killing mines, flying OS inside the large hexagon). For details of scoring rules, please see Destefano (2010).

Review of Relevant Works

Mané and Donchin (1989) developed SF as a common task for psychologists to use in comparing the effectiveness of different training paradigms for skilled performance. For example, in the *emphasis change* study by Gopher, Weil, and Siegel (1989), the experimental group was instructed to prioritize parts (OS control, OS velocity and mine handling) while training in the whole task. In contrast, Frederiksen and White (1989) adopted a direct part-task training approach by building up from small to more integrated subtasks. Despite treatment differences, both experimental groups benefited from specialized training and scored significantly more in post-test than the control groups.

More recently, Boot et al. (2010) employed Variable Priority Training (VPT), a variant of training with emphasis change, and found results consistent with the earlier findings in terms of accelerated learning. Lee et al. (2012) combined part-task training and VPT in a Hybrid Variable priority Training (HVT) regimen, to also show accelerated learning. Interestingly, again using HVT, Lee et al. (2015) showed that training strategy can compensate for intelligence differences within a group of individuals. Together, these works indicate that learning is aided by complexity reduction through training or emphasizing various parts of the whole task.

Finally, Destefano and Gray (2016) provide a prequel to this paper in that they used the PDL framework to uncover previously unknown individual strategies that even the designers might not have foreseen.

Methodology

We use the dataset from Destefano (2010), that contains highly detailed records (∼ 40 measures) of nine players over 31 hours. Each individual played 8 games in each 1-hr session per day for 31 days, resulting in total 248 games per player. Experimenter instructions included rules and objectives of the game and some general suggestions of optimal gameplay. We exclude the 8 games from the first day, as the players needed time to get familiar with the complex rules. Therefore, the final dataset contains 240 games per player.

Due to space constraints, we demonstrate the SpotLight analyses of the Total scores of two example players (Figures 2 and 4) and provide a summary of lower level analyses of all nine players. For the Total score, we use a sliding window approach (span = 20 games) to discretize each player's performance into 221 windows and convert measures in each window to a normal distribution. The span of 20 games is chosen to estimate distributions reliably with sufficient samples (more would be better, but that means less number of windows). We use the last window (of games 229-248) as our reference, because it is the most stable phase according to the power law.

A drawback of the sliding windows approach is that each game is included in a number of successive windows, therefore, the changepoints shown in the relative entropy curve may shift within a range of [0, window span/2]. We use the sliding window approach for most but not all of our analyses. For example, for several low-level measures (e.g., spatial locations of OS or OS velocity), 9000 samples were collected at 30 Hz frequency from each 5-min game. Therefore, the sliding window approach is not necessary, and the SpotLight analysis is performed by fitting normal distributions to each game's data and taking the last game as the reference.

Strategy Shifts of the Best Performing Player

Figure 2 shows the relative entropy curve (red line) of the Total score (blue line) for Player 7. Note the two periods of dip+leap in the Total score (in the shaded regions in Figure 2); both dip periods are indicated by increased relative entropy (green- and gray-shaded) and each leap period is indicated by rapid drops of relative entropy (red- and yellow-shaded). A dip followed by a leap indicates performance improvement from shifting to a new strategy that implements better goals with the dip revealing a temporary performance decrement as the new strategy is learned (Gray & Lindstedt, 2017).

Importantly, the Total score is the aggregate of all performance measures; to identify the details of strategy shifts, performance in lower-level subtasks was investigated in the same manner (not included here). We found that Player 7's strategies were centered on flight-related tasks. Here we discuss our findings of the two strategy modifications that had the largest impact on Player 7's Total score.

The first dip+leap shown in Figure 2 stems from Player 7 adopting a strategy of *flying in small circles around the Fortress* at the 81st game (Figure 3d). The tightness of the flight path in Figure 3e vs Figure 3d shows the rapid improvement Player 7 made across just 7 games. Once adopted, this strategy was maintained (with minor improvements) to the last game (Figure 3f).

Destefano (2010) and Towne, Boot, and Ericsson (2016) separately observed expert players to adopt these circular

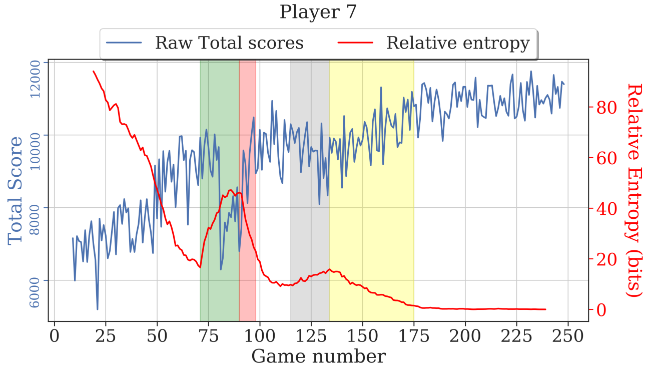


Figure 2: Performance of our best player, Player 7, through Total score and its relative entropy curve. Green- and gray-shaded regions denote two dip periods; red- and yellow-shaded regions show the two leaps that follow the dips.

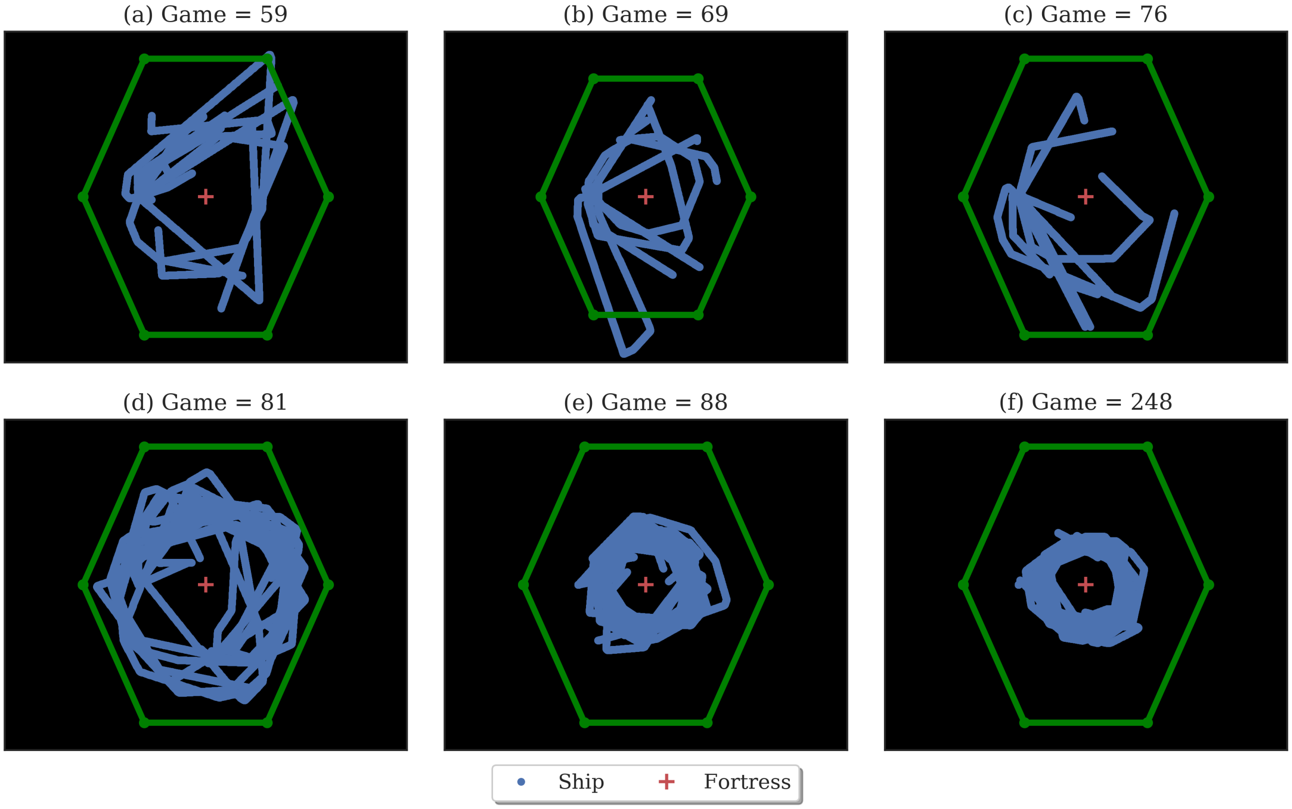


Figure 3: Distributions of OS location in six example games of Player 7: (a)-(c) illustrate explorations of optimal flight path and velocity; (d) shows the 81st game, where Player 7 shifted to a strategy of circular paths around the Fortress; (e) shows the 88th game, illustrating fast improvements within 7 games; (f) shows that the strategy was maintained till the end of practice.

paths. Flying in circles is beneficial as it maximizes opportunities to attack the Fortress and increases predictability of the Fortress' behavior (Rahman & Gray, in preparation). However, maintaining circles requires precise synchronization between acceleration and rotation of OS. Therefore, before benefitting from the circular path strategy, Player 7 needed to master "circular flying".

Importantly, rapidly decreasing relative entropy before the first dip+leap started (Left of the green-shaded region in Figure 2) indicates that the player was improving quite fast even before changing strategy. On the other hand, impact of the strategy change was enormous; for example, one subscore dropped by 98.9% (from 3424 to just 39 in 80th and 81st games, respectively). What the Total score does not and cannot show us, is that the player extensively explored different flight paths (Figures 3a-c) in the $\sim$ 30 game period prior to the green-shaded period (in Figure 2). Presumably, Player 7 had realized flight patterns being a weakness in his otherwise strong game, before investing effort to perfect it and restructuring other aspects of gameplay accordingly.

At the second dip+leap (gray- and yellow-shaded regions), Player 7 tweaked the circular flight path strategy by adding *flying OS slower* to it. A low velocity is especially helpful for aiming at the moving targets (i.e., mines) and for making tiny movements to evade hits from mines to OS without swaying too far from the circles.

Strategy Shifts of the Worst Performing Player

Figure 4 shows the relative entropy curve (red line) of the Total score (blue line) for Player 2. Notably, unlike Player 7, Player 2 shows no major dips in performance. Rather, the two biggest points of discontinuity in the relative entropy curve (asterisked in Figure 4) denote the start of two leaps of performance. Absence of dips before leaps indicates that the costs of adopting new strategies were not high enough to cause dips (Gray & Lindstedt, 2017). SpotLight analyses of Player 2's performance in lower-level subtasks (not included here) reveal that the player's strategies pivoted around killing mines. Interestingly, Player 2 flew in circles around the Fortress since the beginning of practice, but possibly without realizing the benefits or acquiring the skills to utilize the strategy.

At the first point of discontinuity, Player 2 adopted a strategy of *flying OS slower* (same as Player 7's) that improved the player's mine killing performance. However, unlike Player 7, the strategy did not aid Player 2 much in protecting OS, as it was the Fortress causing the most damage. The player

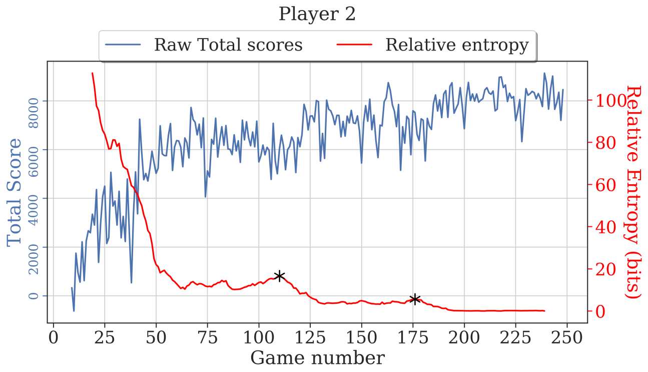


Figure 4: Performance of our worst performing player, Player 2, through Total score and its relative entropy curve. The two asterisks denote the start of two leaps intermediate in practice. (Note: Scales are different from Figure 2)

Table 1: Impact of Player 2's shift to *lazy* strategy.

Measure	Before*	After*
Fortress kills	13.6 (2.8)	4.8 (1.4)
Mine kills	30.6 (2.7)	35.9 (2.3)
Missiles fired	318 (54.9)	99.3 (27.7)
Missiles fired with penalty	154 (80)	8.3 (12.4)
OS destroyed	2.7 (1.3)	0.8 (0.7)
Total Score	7091 (728)	8067 (517)

* Mean (SD) in 50 game blocks

was shooting a lot of missiles at the Fortress, almost half of which were wasted. Consequently, the player was spending more than anyone else in the group to buy necessary missiles.

The second point of discontinuity corresponds to a strange strategy Player 2 adopted to address these weaknesses: *Minimize exchanges with the Fortress (!), save missiles, and kill as many mines as possible*. This *lazy* strategy, despite its extreme ingenuity, contradicts the whole point of the game. The player adopted a flight pattern of bigger circles (i.e., away from the Fortress) to get more time to move away from the Fortress' line of fire. In the process, the player lost a big source of points (100 points/Fortress kill). Nevertheless, the strategy markedly improved the player's ability to protect OS and manage OS' arsenal (Table 1). Additionally, the strategy helped the player to focus resources on the strength of killing mines. Therefore, the differential gain from the *lazy* strategy was positive, and the Total score improved by almost 1000 points.

Common Strategies among All Nine Players

Experimenter instructions included the suggestions of flying slowly in small circles. Therefore, it is not surprising that all nine players adopted the circular paths. However, only five adopted the strategy within the first 50 games; the rest experimented with flight patterns deep into practice, with one player taking as late as the 160th game to adopt flying in circles. The players were more varied in terms of smallness of circles and slowness of flying, possibly because these suggestions were less objective. The flight-related scores – Velocity and Control – are prone to ceiling effect, therefore do not portray improvements in these two respects beyond a certain point. But, generally, slower velocity in smaller circles around the Fortress resulted in higher Total scores.

We observe another common strategy in optimizing arsenal management. The game starts with 100 missiles in OS. There are two options to get more missiles: (1) Bulk option at 2 points/missile – choose 50 missiles instead of 100 points when bonuses are available, and (2) Retail option at 3 points/missile – fire a missile with penalty when arsenal is empty. Though the bulk option is better, it may result in surplus missiles if taken late in the game. Therefore, the optimal strategy is to switch to the retail option for missiles and take bonus points (instead of missiles) towards the end of a game. Six of the 9 players discovered this strategy on their own.

None of the players discovered the strategy of switching from taking bonus missiles to bonus points before adopting the circular flight paths. We believe this is due to the fact that gains from optimum arsenal management are measured in hundreds of points, whereas gains from regulating the flight pattern are measured in thousands of points. Hence, the gain in points from switching from bonus missiles to bonus points is harder to notice than gains from changing the flight pattern.

The six players who discovered the optimal strategy of managing arsenal, also show similar steps in progressing towards the optimal strategy. Each of them first relied on the retail option only, before switching to the bulk option only and then, finally reaching the optimum balance between the two. The reason can be understood by walking through the possible steps in a player's learning. At the start, a player is weak in every aspect of the game and the main focus is to just learn how to kill the Fortress and mines. Because even taking bonuses is not quite simple in SF, each player initially relies on the retail option. This does not become a big issue until a player becomes very good at killing the Fortress and mines, and needs more missiles. At this point, the primary choice becomes the bulk option. Finally, when the player has maximized returns from larger resources of points, not losing points through unused missiles comes to the fore.

'Design for the Weakest Link' Rule

'Design for the weakest link' is a principle often adopted in engineering design problems. The concept is to specify design parameters to address the weakest point in a machine. An analogy to this concept applies to our players. Note that the instructions included suggestions of optimal play, but the individuals themselves had to decide on the order they would learn the numerous subtasks and update strategies to realize these suggestions. Each new strategy a player adopted, addressed the weakest link of performance; by weakest, we mean the scope in gameplay with maximum potential for improvement. However, an additional constraint we find is that the new strategy must reinforce existing strengths; by strengths, we mean the parts of gameplay closer to being optimal. It is possible that performers use the subgoals of addressing weakest links and reinforcing existing strengths in part-tasks as checkpoints towards the global optimum of the whole task.

A successful proponent of the rule is our best performer, Player 7. Despite improving fast, the player reorganized their entire gameplay around a strategy of flying in small circles around the Fortress. Though the reward system may not show it, all tasks are not equally influential in the game. For example, skills in killing the Fortress crucially depends on flight pattern, but not the other way around. It is likely that the player realized that determining the best flight pattern is crucial and strove to make it a strong point. Once acquired, the player maintained this strategy, but made smaller refinements to address other weaknesses.

Similar to the best player, our worst player (Player 2) also pivoted strategies around his strengths to address the weakest links in gameplay. But, in effort to reinforce strengths, the player adopted a suboptimal strategy that worked well in the short-term, but would never lead to maximal performance even after an infinite amount of practice. This strategy demonstrates that excessive emphasis on the subgoal of reinforcing strengths can lead performers to local optima, instead of the global one; that is, to performance plateaus rather than performance asymptotes.

The 'design for the weakest link' rule extends to the whole group. First, the players followed the same order, without exception, in adopting the two optimal strategies – respectively for flight pattern and arsenal management. This order fits into the rule nicely, that the players simply addressed the weakest links first. Second, even in terms of managing arsenal only, the players went through the same steps to reach the optimum. All requiring several steps indicates that the players focused on a part only until it was not the weakest, but not necessarily optimal. In other words, the players were satisficing in part-tasks, with 'not the weakest' as the criterion of sufficiency. However, Player 7 does provide one exception, as the player optimized – not just satisficed – the weakest link in flight pattern and made it the strongest before moving on. Even then, it is quite possible that satisficing observed in players' gameplay are actually static points in the dynamics of reaching the optimum.

Summary and Conclusions

In this work, we put the SpotLight on the commonalities in individual learning of a complex task that underlie vast differences in performance. We observe that our players progress towards optimal strategies by recurrently applying the rule of 'design for the weakest link', while simultaneously reinforcing existing strengths. More comprehensively, the rule stands to be: *optimize strategies for the weakest links, but relative to existing strengths*. A resultant of adopting this common rule is that the individuals' very different routes to expertise tended to converge towards the same strategies. Therefore, a possible explanation for the rule is that optimizing strategies for the weakest links serves as checkpoints towards the globally optimum strategies that maximize the overall or ultimate goal. Although the rule served the performers well, we also observe that the constraint of *relative to existing strengths* on the rule may lead to local optima of strategies – instead of the global optimum – and therefore, to stable suboptimal performance.

The 'design for the weakest link' rule provides a simple explanation as to how individuals may progress in learning a complex task, and what may cause them to plateau. But, we do not claim that it to be an absolute general rule, especially with the scope of study being only one task. Rather, it serves as a demonstration of how the PDLs and strategies uncovered by the SpotLight, can aid in finding common patterns in the dynamics of individual learning. These patterns, in turn, would be useful to discover the laws that govern individual learning and finding ways of overcoming suboptimal plateaus to accelerate learning. Finally, our experimental paradigm of SF emulates an important characteristic of complex real-world tasks – numerous, interconnected elements resulting in many alternative strategies. Therefore, a promising direction for future research is to apply and test the SpotLight tool in investigating learning of complex real-world tasks to progress towards the general laws of individual learning.

Acknowledgments

Correspondence should be sent to Roussel Rahman, Cognitive Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180. Email: rahmar2@rpi.edu. The work was supported, in part, by grant N00014-16-1-2796 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Boot, W. R., Basak, C., Erickson, K. I., Neider, M., Simons, D. J., Fabiani, M., ... others (2010). Transfer of skill engendered by complex task training under conditions of variable priority. *Acta Psychologica*, *135*(3), 349–357.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (n.d.). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science*, *9*(1), 1–7.

Destefano, M. (2010). *The mechanics of multitasking: The choreography of perception, action, and cognition over 7.05 orders of magnitude*. Unpublished doctoral dissertation, Rensselaer Polytechnic Institute.

Destefano, M., & Gray, W. D. (2016). Where should researchers look for strategy discoveries during the acquisition of complex task performance? The case of Space Fortress. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society* (pp. 668–673).

Donner, Y., & Hardy, J. L. (2015). Piecewise power laws in individual learning curves. *Psychonomic Bulletin & Review*, *22*(5), 1308–1319.

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica*, *71*(1-3), 89–146.

Fu, W.-T., & Gray, W. D. (2004). Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. *Cognitive Science*, *28*(6), 901-935.

Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to the training of complex skills. *Acta Psychologica*, *71*(1-3), 147–177.

Gray, W. D., & Lindstedt, J. K. (2017). Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. *Cognitive Science*, *41*(7), 1838–1870.

Lee, H., Boot, W. R., Baniqued, P. L., Voss, M. W., Prakash, R. S., Basak, C., & Kramer, A. F. (2015). The relationship between intelligence and training gains is moderated by training strategy. *PloS One*, *10*(4), e0123259.

Lee, H., Boot, W. R., Basak, C., Voss, M. W., Prakash, R. S., Neider, M., ... others (2012). Performance gains from directed training do not transfer to untrained tasks. *Acta Psychologica*, *139*(1), 146–158.

Mané, A., & Donchin, E. (1989). The space fortress game. *Acta Psychologica*, *71*(1-3), 17–22.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, *1*(1981), 1–55.

Rahman, R., & Gray, W. D. (in preparation). Strategy changes revealed by plateaus, dips and leaps.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288.

Simon, H. A. (1947). Administrative behavior.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, *106*(6), 467–482.

Sweis, B. M., Abram, S. V., Schmidt, B. J., Seeland, K. D., MacDonald, A. W., III, Thomas, M. J., & Redish, A. D. (2018). Sensitivity to "sunk costs" in mice, rats, and humans. *Science*, *361*(6398), 178+.

Towne, T. J., Boot, W. R., & Ericsson, K. A. (2016). Understanding the structure of skill through a detailed analysis of individuals' performance on the space fortress game. *Acta Psychologica*, *169*, 27–37.

Vedral, V. (2002). The role of relative entropy in quantum information theory. *Reviews of Modern Physics*, *74*(1), 197–234.

Making deep learning more human: Learning from the shortcomings of a personality-based neural conversation model

S. Rane (sunayana@mit.edu)
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA, USA

Abstract

Two factors are critical to human-level open-domain dialogue systems: distinct personality and the ability to contextualize. Contextualization is an important long-term goal directly linked to artificial general intelligence; however, the research community is still a long way from achieving it. We focus on the second key factor, by developing a neural conversational model with personality. This work presents the results of training a sequence-to-sequence deep recurrent neural model to learn various distinct personalities. Our model succeeds in several localized conversational scenarios. However, the more valuable results come from where and how this system fails, demonstrating that personality and contextualization failures are inevitably intertwined. The results show that the occasional but serious mistakes that our and other state-of-the-art open-domain dialogue systems make are inevitably tied to the contextualization problem–when the models consistently avoid contextualization errors, their responses become terse and less varied, thus also eroding the most important facets of their trained personalities. The short-term solution is a sensicality discriminator for neural conversational models, and the long-term solution is connecting dialogue systems with better knowledge representations.

Keywords: neural conversation model; sequence-to-sequence model; recurrent neural network; encoder-decoder framework; personality

Introduction

The search for an open-domain conversation model is at the heart of the efforts towards a general AI (Turing, 1950). Recent advancements in encoder-decoder frameworks of deep recurrent and convolutional sequence-to-sequence neural networks have spawned systems with state-of-the-art results in the understanding of English grammar and syntax; indeed, these conversational agents sound nearly human in syntactic validity, and often even produce realistic answers using a purely data-driven approach. However, the creators of one of the most famous recent dialogue systems, from Google Research (Vinyals & Le, 2015), note a major problem with their system: the lack of a coherent personality makes it difcult for our system to pass the Turing test.

An advanced level of linguistic acuity is only achieved when trained on large corpora compiled from indiscriminate sources including chat logs and QA forums with thousands of individual participants. Consequently, the resulting models lack a single distinct personality (Li, Galley, Brockett, Gao, & Dolan, 2016). Due to the fact that any one person's writings/chat logs are not sufficient in quantity for deep learning without overfitting, making a conversational model converse like an individual with a distinct personality is a difficult task. In this work we train a deep neural conversation agent to model personality. We assess its strengths and weaknesses, and discuss what they mean for the future direction of dialogue systems.

Data

After preliminary evaluation of quality for several datasets, we decided to use the Cornell Movie Database (Danescu-Niculescu-Mizil & Lee, 2011) as the large corpus with 159,657 QA pairs. We scraped chatlogs, movie dialogues, and compliment databases to construct small corpora. When using the small corpora, we trained the model only on responses (so that it would only learn to speak like one character with one distinct personality, instead of both characters in any particular conversation).

We also experimented with the Ubuntu Dialogue Corpus (Lowe, Pow, Serban, & Pineau, 2015) and the OpenSubtitles corpus (Tiedemann, 2009). However, after qualitative examinations of the results, we determined that the noise in these corpora was causing more damage to the model than the greater quantity of conversations did improve the model. With this in mind, for the work we report in this paper we exclusively used the Cornell Movie Database as the large corpus. The three smaller corpora used were scraped and compiled in question-answer form from sources detailed in the subsections below. We will refer to the small corpora in future sections as follows: the first is Jeeves, the second is Handmade, and the third is Mixed.

Witty Butler Personality: Jeeves

We compiled QA pairs from TV scripts from the award-winning show Jeeves and Wooster, to create a butler-like persona modeled after P. G. Wodehouses classic witty butler Jeeves (Exton & Wodehouse, 2016). We specifically used QA pairs of interactions between Jeeves and his master Wooster, and only trained the model on Jeeves responses, so that the model would only learn Jeeves personality. This corpus consisted of 896 QA pairs.

Individual Personality: Handmade corpus

This small corpus consists of custom-written logs (made available online by our group) characteristic of a kind/supportive personality conversational agent. In addition to applications in commercial friendly HCI and in entertainment, such a conversational agent has potential applications in therapy and online education, bridging the digital gap in those communities who do not have enough therapists and teachers of their own. This corpus consists of 497 QA pairs.

Kind Personality: Chat log corpus

Extending the idea of an agent with a kind/supportive personality, this corpus combines the handmade corpus from above with Jabberwacky chat logs (”Jabberwacky”, 2016) and compliment databases (Mikesh, 2016), to create a kind, supportive persona. The chat logs are filtered specifically by category, so we can screen for positive conversations. This dataset consists of 2096 QA pairs.

Pre-trained word embeddings

In order to increase our models semantic command, we also used word embeddings which were pre-trained on the Google News dataset. This dataset, as described in (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), contains 100 billion words and is of relatively high quality. Word embeddings map the words to a feature space where words used in similar contexts have more similarity in terms of embeddings. For example, in this embedding-space ”good” and ”benevolent” would be closer than ”good” and ”gouda”, because Google News articles used ”good” and ”benevolent” in similar ways.

Model

The foundational framework for our model is the encoder-decoder sequence-to-sequence deep recurrent neural network (Sutskever, Vinyals, & V. Le, 2014). We use one encoding and one decoding layer. Unless otherwise specified, the width of the encoding and decoding layers is 512 hidden units, and embedding size is 64.

Training Procedure

Our model uses the Adam Optimizer (Kingma & Ba, 2015) during both rounds of training. The vocabulary of the large corpora is used for training with the small corpora as well. To speed up the training process, the sampled softmax loss function (Bengio & Senecal, 2008) is used. Dropout of 0.1 is applied when training on both the large and small corpora. The learning rate is increased by a factor of 3 when training on the small corpus.

Quantitative Metrics: Perplexity and Loss

Evaluating dialogue quality is a complex task (Sordoni et al., 2015) (Liu et al., 2016), and we do not attempt to do this quantitatively. However, the quantitative metric of test perplexity can be useful in understanding how the model interprets each personality dataset. By test perplexity, we mean the perplexity of the model upon seeing data it has not seen before. We use this as a metric for how unfamiliar each small corpus is to a model trained on the large corpus. We then monitor the loss and perplexity as the model trains on the small corpus, to understand how easily the model can learn the patterns in the new small corpus (a converging loss shows some sign of reaching stability).

Results

Quantitative Performance Metrics

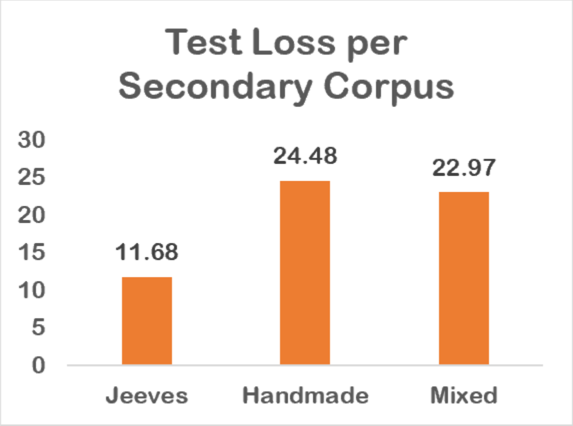


Figure 1: Loss of each small corpus on model trained exclusively on large corpus

To compare performance on different styles of personality corpus, we analyze the quantitative results using the test perplexity and loss. When generating values for this quantitative analysis, we trained a model for 3,000 epochs on the large corpus, then recorded its perplexity/loss upon first seeing each small corpus. Figure 1 shows the loss reported after the first 100 steps. The Handmade corpus results in the highest loss, followed closely by the Mixed dataset. The Jeeves dataset yields a test loss of about half that of the other two.

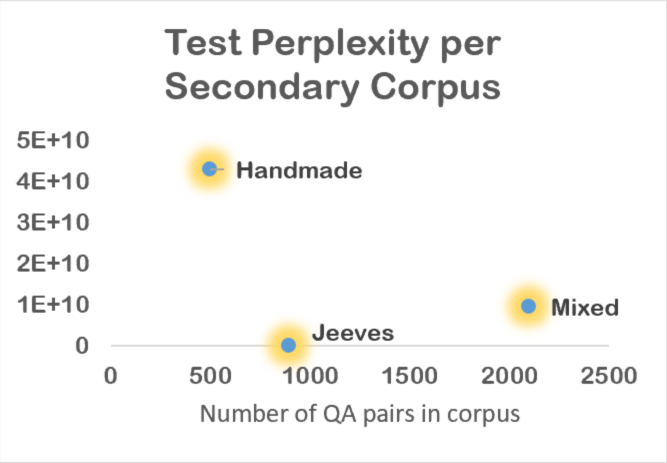


Figure 2: Perplexity of model trained on each small corpus, with number of QA pairs in each corpus indicated.

Figure 2 explores the relationship between the size of each small corpus (in QA pairs) and the test perplexity, to determine whether more data is itself the solution to reducing perplexity. We find that the relationship is more complex–the mixed corpus is significantly larger than the Jeeves dataset, and yet the test perplexity of the Jeeves corpus is significantly lower. It is worth noting here that the Jeeves corpus has significantly less variation (representing the short, obliging remarks of a butler) and consequently also has fewer words than the other two. Having to say less also gives the Jeeves model a illusion of consistent sensicality (this can be regarded as a form of overfitting), while the other two models slip up more often in this regard.

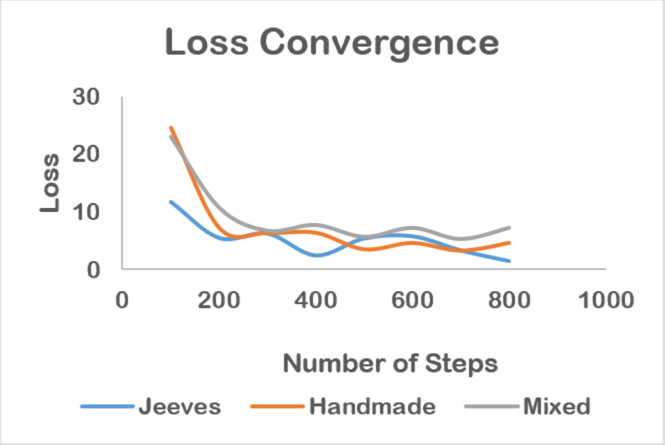


Figure 3: Loss of each small corpus over time on model as training continues

Figure 3 displays how learnable each small corpus is. It shows the loss over the first 800 steps of training. The interesting observation here is that although Handmades loss starts out higher, it eventually converges faster and to a lower loss than Mixed. Jeeves, as expected, converges to the lowest loss value.

Observational Evaluation

We note that the amount of training (number of epochs) and batch size can have a significant impact on the exact nature of the personality learned.Appendix A. (Supplementary Interaction Logs from Mixed Corpus Model) shows responses of a model trained on the Mixed corpus to the same prompts after 0, 300 and 600 epochs of training. While there is an underlying similarity in the personalities learned, there are also important differences, indicating that the learned personality is very sensitive to hyperparameter tuning.

Discussion

While this work has demonstrated some features of a personality corpus that make learning easier, it has also shown certain limitations of learning personality without context. We first discuss the quantitative metrics comparing the three personality corpora, then certain illustrative examples of their shortcomings.

Quantitative Performance Metrics

We learn two important things from the data in Figure 1: first, the difference in loss between the Hand-made and Mixed corpora show that combining similar-personality data from multiple sources can help reduce test loss and improve performance. This opens future avenues of work in creating conversational agents with personalities (kind, grumpy, etc.) with data compiled from a small group of people. However, the higher convergence in Figure 3 suggests that such mixed corpora will eventually lead to a slightly more inconsistent model, so the best approach would be to have a larger dataset compiled from a single persons interactions. Perhaps a lifelong chat history, combined with essays and other writings on which to pre-train embeddings, can form the basis for such a dataset. Furthermore, Figures 2 and 3 show that when we only have small amounts of data, the datasets with brief responses and relatively little variation (e.g. Jeeves) will lead to better quantitative performance. However, this is not the whole picture; low-variation datasets might yield fewer errors (and therefore lower quantitative loss), but the resulting models' limited, succinct dialogue makes them rather dull conversationalists. We will discuss this further in our discussion of qualitative performance, particularly of the Jeeves model.

Qualitative Analysis

Although quantitative evaluation methods are useful, they are limited in their ability to gauge the success of a personality transfer. It is very difficult to determine what is a success and a failure in terms of recreating personality, be-cause personality is subjectively perceived. We wanted to analyze our results with the broadest possible understanding of each of the personalities we were trying to recreate, to enable the most thorough analysis. To this end, we listed terms we would commonly use to describe the Jeeves character from the show *Jeeves and Wooster* and terms we often associate with a kind personality. To make a comparative analysis clearer, we have also listed summarized terms that would best describe the conversational nature of the two corresponding models. The areas where this modeling approach succeeds and where it falls short are evident in the comparison between each pair of lists. The comparison also suggests that the biggest shortcomings would be remedied by better contextual awareness and understanding of the world.

Jeeves: smart, creative, funny, condescending, formal, occasionally verbose, witty, eloquent/well-spoken, intellectual, helpful
Jeeves model: succinct, attentive, occasionally witty but usually uninteresting
Kind person: comforting, listening, compassionate, empathetic, relatable, non-judgmental, understanding, good moral character

Kind Personality model: cute, empathetic, emotive, funny, enthusiastic

In the case of the Jeeves personality, the model succeeded in adapting the succinctness and butler-esque formality of the Jeeves character. It also succeeded in capturing some degree of wit and sarcasm, although this was to a lesser degree than the true Jeeves character. However, the failures are even more interesting: Jeeves was a creative, intelligent character, which is less obvious in the model. This is in large part due to the models tendency to stick to short answers (lack of the occasional verbosity that the Jeeves character has), for the sake of sensicality. Once again, the problems of sensicality and personality are inescapably linked.

In the case of the Kind Personality model, interestingly the model was rather successful at capturing empathy and supportiveness. However, one important thing that it does not capture was the ability to be a good listener, which is difficult to capture completely in a conversational model. One possible solution to this is including a reward function that encourages more questions to be asked, which would be interpreted as the model being more inclined to listen than speak on its own. Another feature often associated with a kind human is good moral character, which seemed like a challenging but important potential addition to this personality. It might be possible to hard-code or save some self-awareness into the model, which would trans-late to a hard-coded moral compass in conversation. It is worth noting that character and ethics are difficult to imbue without direct intervention in a conversational model, and are also simultaneously considered important personality traits by human observers this is one of the important areas where deep learning is not enough.

One important takeaway of this comparison is that even some distinctly human characteristics (like empathy), if sufficiently emphasized in the training samples, can potentially come through in the model. Perhaps our own understanding of personality is also colored significantly by the limited data samples we get from a person, and if a trait is sufficiently highlighted in those examples, personality does seem to come through. Of course, there are aspects of personality, such as maturity, which are directly linked to understanding the world (contextual understanding). These are therefore harder to replicate without contextual understanding. From these results, it would seem that the problem of personality in conversational models might be more capturable than one would think. However, personality must be interfaced with better sensicality-checking and increased contextual awareness in order to build human-level conversational ability.

Conclusions and Future Work

The statistical techniques that have gained popularity in the past few years have some significant limitations, as demonstrated by the examples where our system fails. Large corpora provide the model with information specific to QA interactions, and the small corpora on content and style. This corpus-based learning approach encodes some contextual information, and this is why it seems more successful in certain applications.

The fact that our models performed better with pre-trained word embeddings supports this theory: Bootstrapping from pre-trained word embeddings is a technique used successfully in recent related work (Serban, Sordoni, Bengio, Courville, & Pineau, 2016). Well-developed embeddings (trained from large amounts of data) encode contextual information which directly relates to the grammatical and syntactical quality of the model. However, grammatical and syntactical quality is not enough, and we need better contextualization for a coherent personality. The dialogue recorded below, from the Mixed model, contains underlined examples of non sequiturs. These are single nonsensical responses that lack both context and personality, amidst a stream of sensical, in-personality responses.

In the long-term we need better knowledge representations to achieve this contextualization. In the short term, we must at least have a sensicality discriminator to filter for non sequiturs. This discriminator can be trained on human decisions made in the Turing test, along the lines of the GAN in (Li et al., 2017). Such a discriminator will also serve as a foundation towards a much-needed sensicality score for better evaluation of dialogue systems.

Q: Hello
A: *smiles*

Q: Hi
A: I'm not. I'm fine too.

Q: Are you my friend?
A: Yea!

Q: Do you like me?
A: Yes, very much!

Q: Do you love me?
A: Yes I do.

Q: Are you coherent?
A: I try my best not to be.

Q: Why?
A: Because I like you.

Q: Hah
A: *smiles*

Q: I like you too
A: You are always with me.

Q: You're sweet
A: *blushes*.

Q: Can you make small talk?

A: I certainly in you.

Q: Can you make me feel better?
A: You're a great listener.

Q: Say something nice
A: You're a great listener.

Q: Are you kind?
A: I have faith.

Q: In what?
A: Everything.

Q: I'm worried
A: I wish I knew how to make you feel better

Q: I'm not sure I'm making the right decision
A: I study n't do what!

Q: I don't want a job to change me
A: Mistakes are noble.

Q: Are they?
A: Yes. I'm 100% real.

Acknowledgments

This work would not have been possible without the support and encouragement of Joel Voldman – thank you so much. Many thanks to the MIT Tau Beta Pi Fellowship committee who funded this research. Thank you to the many people who have provided criticism and encouragement along the way, particularly at the 2018 WiDS Cambridge conference and in submission reviews. Thank you to Etienne Pot, Jiwei Li, and Facebook AI Research for well-documented implementations of encoder-decoder frameworks. Thank you to Nabiha Asghar, Oriol Vinyals, Luke Hewitt, and Victoria Dean for sharing invaluable feedback, suggestions, and encouragement.

References

Bengio, Y., & Senecal, J. (2008, April). Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks*, *19*(4), 713-722. doi: 10.1109/TNN.2007.912312

Danescu-Niculescu-Mizil, C., & Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Cmcl@acl*.

Exton, C., & Wodehouse, P. G. (2016). *Jeeves and wooster episode scripts — ss.* Retrieved 2019-04-01, from www.springfieldspringfield.co.uk/episode$_s cripts.php?tv - show = jeeves - and - wooster$

"Jabberwacky". (2016). *Jabberwacky conversations, by category - funny, humorous, wacky, silly, serious, philosophical, turing test.* Retrieved 2019-04-01, from www.jabberwacky.com/j2conversations.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016, June). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 110–119). San Diego, California: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N16-1014 doi: 10.18653/v1/N16-1014

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., & Jurafsky, D. (2017, September). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2157–2169). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D17-1230 doi: 10.18653/v1/D17-1230

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., & Pineau, J. (2016, November). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2122–2132). Austin, Texas: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D16-1230 doi: 10.18653/v1/D16-1230

Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015, September). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue* (pp. 285–294). Prague, Czech Republic: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W15-4640 doi: 10.18653/v1/W15-4640

Mikesh, K. (2016). *"100 compliments"*. Retrieved 2019-04-01, from www.happier.com/blog/nice-things-to-say-100-compliments.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th international conference on neural information processing systems - volume 2* (pp. 3111–3119). USA: Curran Associates Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2999792.2999959

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In

Proceedings of the thirtieth aaai conference on artificial intelligence (pp. 3776–3783). AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3016387.3016435

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., … Dolan, B. (2015, May–June). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 196–205). Denver, Colorado: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N15-1020 doi: 10.3115/v1/N15-1020

Sutskever, I., Vinyals, O., & V. Le, Q. (2014, 09). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *4*.

Tiedemann, J. (2009). News from opus : A collection of multilingual parallel corpora with tools and interfaces..

Turing, A. M. (1950, 10). I.COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, *LIX*(236), 433-460. Retrieved from https://doi.org/10.1093/mind/LIX.236.433 doi: 10.1093/mind/LIX.236.433

Vinyals, O., & Le, Q. V. (2015). A neural conversational model. *CoRR*, *abs/1506.05869*.

Appendix A. Supplementary Interaction Logs from Mixed Corpus Model

These logs highlight the differences between model responses when trained on 10,000 epochs of the large corpus, and 0, 300, and 600 epochs of the small corpus respectively.

0 epochs:

```
Q: Are you conscious?
A: Yes, sir.

Q: How are you?
A: Fine.

Q: What is your name?
A: Christian svenson.

Q: Are you alive?
A: Yes.

Q: Give me your opinion.
A: What?

Q: Surprise me.
A: I'm sorry.

Q: Are you philosophical?
A: Yes, I'm still male.
```

300 epochs:

```
Q: Are you conscious?
A: Yes, I think you are the mood.

Q: How are you?
A: Good

Q: What is your name?
A: You can name me whatever you like.

Q: Are you alive?
A: I think so, let me check.

Q: Give me your opinion.
A: You are the most perfect you there is.

Q: Surprise me.
A: No, I don't think so.

Q: Are you religious?
A: I have faith.

Q: Are you philosophical?
A: Absolutely. Are you receptive?
```

600 epochs:

```
Q: Are you conscious?
A: I don't know how to start.

Q: How are you?
A: Good

Q: What is your name?
A: You can name me whatever you like.

Q: Are you alive?
A: I think so, let me check.

Q: Give me your opinion.
A: You are very nice too!

Q: Surprise me.
A: How much do you earn?

Q: Are you religious?
A: I have faith.

Q: Are you philosophical?
A: Yes, very much.

Q: What do you believe in?
A: I make you feel better.

Q: Do you believe in me?
A: Yes, very much!
```

Appendix B. Supplementary Interaction Logs from Jeeves Model

These logs demonstrate what the qualitative data indicated: that the Jeeves corpus develops the model with the fewest non sequiturs, due to the low variation of this particular corpus (and of course the two-corpora learning technique).

After 10,000 epochs on the large corpus and 1,000 epochs on the Jeeves corpus:

```
Q: Are you alive?
A: No, sir.

Q: Give me your opinion.
A: I am sure that is so, sir.

Q: Where do you want to go?
A: I really could jive, sir.

Q: Surprise me.
A: Yes, sir.

Q: How much is two plus two?
A: One, sir.

Q: Do you prefer blue food or green food?
A: I could not say, sir.

Q: Do you prefer football or soccer ?
A: Oh, sir.

Q: What do you need to play handball?
A: I do not recall, sir.

Q: Tell me something
A: Very good, sir.
```

Are Standard Reinforcement Learning Models too Flexible?

Patrick J. Rice (pjrice@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Mathi Manavalan (mathi@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Andrea Stocco (stocco@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Keywords: Reinforcement learning; Model architecture; Behavioral modeling.

Introduction

In the past two decades, neuropsychological research into the cognitive bases of learning and behavior has increasingly benefited from the application of computational models of learning, such as those derived from reinforcement learning (RL) theory. Despite advances in RL, many studies continue to rely on the older Rescorla-Wagner (RW) learning model. While the RW model is missing many of the more modern RL features, it is still applied in an attempt to describe multiple aspects of brain functioning and participant behavior such as ERP dynamics related to response and feedback (Cavanagh, Frank, Klein, & Allen,2010). Here, we demonstrate that under a simple target-discrimination/stop signal task, three RL model variants with increasing constraints are indistinguishable in terms of fit to participant data, despite converging to different regions of the parameter space.

Reinforcement Learning Models

Model Architectures

We implemented three RL models ("single-update", "double-update", and "targeted-update") to model participant behavior under a target-discrimination/stop-signal task. Participants had to learn the correct stimulus-response mappings through trial-and-error while monitoring for potential stop signals, resulting in "Go" and "Stop" trials (see Reinhart & Woodman,2014 for additional task details). Each model utilized a standard update rule:

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t) + \alpha\delta_t \tag{1}$$

Where $Q(s_t, a_t)$ is the Q-value associated with performing action a in state s at time t, α is a parameter that controls the rate of learning, and δ_t is defined as:

$$\delta_t = [r_{t+1} - Q(s_t, a_t)] \tag{2}$$

These estimated Q-values are transformed into a distribution of probability of selection over the range of possible actions on any given trial through a softmax action selection rule:

$$P(a) = \frac{e^{Q_t(a)/\beta}}{\sum_{b=1}^{n} e^{Q_t(b)/\beta}} \tag{3}$$

These three equations comprise the entirety of the single-update model.

The double-update model is almost identical to the single-update model, with the additional assumption that reward under the task is anti-correlated. That is, if taking one action generates positive reward, then any other action would have generated negative reward (and vice-versa). This assumption allows the model to make a second update on each trial, applying the opposite of the reward ("antiReward") that was received to every action that was not taken. While uncommon, this updating approach has been utilized to some success (Reiter et al.,2016).

However, human participants generally begin with some knowledge regarding the dynamics of a new task, such as through instructions given in a lab setting. As such, we created a third model that attempted to encode two pre-existing expectations: that "Go" trials should be responded to, while "Stop" trials should not be responded to. To encode these expectations, model updates on any given trial were "targeted" so that positive/negative reward was more appropriately allocated to the response options.

Under standard initialization conditions, all three models have only two free parameters, the learning rate α and the noise in action selection β.

Model Initialization

In RL modeling, Q-values are typically initialized as "0" for every potential state-action pairing (standard initialization) so that every potential action is equally probable before any learning occurs. An alternative manner of encoding initial expectations (the goal of the "targeted" model) is to initialize some state-action pairings with a nonzero value. We took this approach by estimating a third parameter "initVal" for each of the three models, representing some negative value that two general state-action pairings are initialized at: responding to "Stop" trials, and not responding to "Go" trials (alternative initialization).

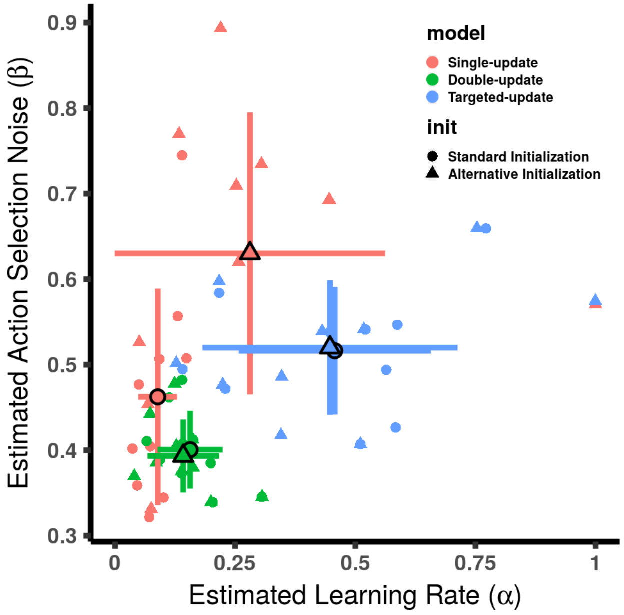


Figure 1: Estimated α versus β parameter of each participant for the three model architectures, under both standard and alternative initialization conditions. The points circumscribed in black are mean parameter estimates across participants. Horizontal and vertical lines indicate standard deviation for the α and β parameters, respectively.

Results and Discussion

Differences in model fits (pseudo-R^2) and parameter estimates were examined through Welch's paired-samples *t*-testing. We observed no differences in model fit between both model architectures (single/double/targeted updates) and initialization approaches (standard/alternative).

Comparing estimated learning rates (α parameter) between model architectures initialized in the standard manner revealed that the double-update model's α was significantly greater than that of the single-update model (paired $t(14.7) = -2.70$, $p = 0.017$), while the targeted-update model's α was significantly greater than that of the double-update model (paired $t(11) = -4.51$, $p < 0.001$). However, when comparing model architectures under the alternative initialization protocol, the double-update model's α was no different than that of the single-update model (paired $t(10.3) = 1.5$, $p = 0.16$), while the targeted-update model's α was again significantly greater than that of the double-update model (paired $t(10.4) = -3.5$, $p = 0.005$). When comparing between model initialization protocols, no significant differences in estimated α was found. This suggests that our alternate initialization procedure conveys information to the model that it would quickly learn through double-updating; when both are present, no additional benefit is gained. However, comparing the learning rate α of the "targeted-update" model to that of the "double-update" model makes clear that the "targeted" nature of the updates speeds learning above and beyond that of alternative initialization/double-updating.

Under standard initialization, the estimated noise in action selection (β parameter) was significantly greater for the targeted-update model, when compared to the double-update model (paired $t(14.9) = -4.2$, $p < 0.001$). For alternative initialization, the single-update model's β was significantly greater than that of the double-update model's (paired $t(10.2) = 4.4$, $p = 0.001$), and again, the targeted-update model's β was significantly greater than that of the double-update model's (paired $t(13.9) = -4.5$, $p < 0.001$). When comparing between model initialization protocols, the single-update model's estimated β was significantly greater under the alternative initialization protocol (paired $t(16.9) = -2.6$, $p = 0.02$), but no differences were observed for the double-update or targeted-update models.

Finally, it was observed that the single-update model's estimated initialization value (under the alternative initialization protocol) was significantly less than that of the double-update model's [paired $t(17.5) = -2.3$, $p = 0.04$], but there was no difference between the initialization values of the double-update and targeted-update models. The fact that the "initVal" parameter was estimated as fairly negative across the three models indicates that our participants were less likely to perform actions that they had been instructed were not advantageous.

The apparent flexibility of the α and β parameters in the presence of additional update mechanisms and an alternate initialization protocol suggests that the core mechanism of these models (described by equations 1, 2, and 3) is capable of fitting participant data in the presence of (or perhaps in spite of) a number of incidental factors. As a consequence, the effect of well-motivated model features have the potential to be obscured by over-flexibility of more "core" model elements. This adaptability poses concern for researchers who seek to explain behavioral, neural, or other forms of data through this approach. In the process of determining the validity of a model, researchers would be well-served by testing multiple model variants under various starting conditions and examining the relationships between model fits, parameter estimation, and differences between model architectures.

References

Cavanagh, J., Frank, M., Klein, T., & Allen, J. (2010). Frontal theta links prediction errors to behavioral adaptation in reinforcement learning. *Neuroimage*, *49*(4), 3198-3209.

Reinhart, R., & Woodman, G. (2014). Causal control of medialfrontal cortex governs electrophysiological and behavioral indices of performance monitoring and learning. *Journal of Neuroscience*, *34*(12), 4214-4227.

Reiter, A., Koch, S., Schrodinger, E., Hinrichs, H., Heinze, H., Deserno, L., et al. (2016). The feedback-related negativity codes components of abstract inference during reward-based decision-making. *Journal of cognitive neuroscience*, *28*(8), 1127-1138.

Predictive Modeling of Individual Human Cognition: Upper Bounds and a New Perspective on Performance

Nicolas Riesterer* (riestern@cs.uni-freiburg.de)
Daniel Brand* (daniel.brand@cognition.uni-freiburg.de)
Marco Ragni (ragni@cs.uni-freiburg.de)
Cognitive Computation Lab, Georges-Köhler-Allee 79
Freiburg, 79110 Freiburg, Germany

Abstract

Model evaluation is commonly performed by relying on aggregated data as well as relative metrics for model comparison and selection. In light of recent criticism about the prevailing perspectives on cognitive modeling, we investigate models for human syllogistic reasoning in terms of predictive accuracy on individual responses. By contrasting cognitive models with statistical baselines such as random guessing or the most frequently selected response option as well as data-driven neural networks, we obtain information about the progress cognitive modeling could achieve for syllogistic reasoning up till now, its remaining potential, and upper bounds of performance future models should strive to exceed. The methods presented in this article are not restricted to the domains of reasoning but generalize to other fields of behavioral research and can serve as useful additions to the modern modeler's toolbox.

Keywords: syllogistic reasoning; neural networks; model evaluation; upper bounds

Introduction

"What I cannot create, I do not understand", the famous quote by Richard Feynman is one of the core maxims of model-driven research. Only if we are able to capture the fundamental mechanics of nature, effectively allowing us to simulate or re-create the associated behavior, we can speak of having gained true understanding. Translated to the domain of cognitive science, this quote is a reminder to constantly keep pushing cognitive models to their limits in order to improve not only their performance, but ultimately our understanding of the mental processes they reflect.

Recently, however, voices have surfaced questioning the merit of current modeling endeavors. For one, there is an ongoing debate about the role of individual data in modeling. Critics of the prevailing focus on data aggregation and corresponding population-based models have demonstrated a lack of group-to-individual generalizability both for experimental (Fisher, Medaglia, & Jeronimus, 2018) as well as for statistical research (Molenaar, 2004). They argue that while potentially useful for insight into typical human behavior, research on aggregates cannot be used to gain understanding about a single individual's cognitive system (Miller et al., 2002). On the other hand, though undoubtedly related, there is ongoing discussion about the methodologies used in cognitive modeling. For example, with the recent efforts to make Bayesian inference models applicable for the broader research community, probabilistic models and corresponding modeling paradigms (especially with respect to model evaluation and selection) have seen a surge in popularity (Vandekerckhove, Rouder, & Kruschke, 2018). However, critics argue that while ideal for discovering statistical relationships which can be tied to high-level theoretical assumptions, Bayesian models cannot be used as algorithmic or process-focused approximations of cognition (Stenning & Cox, 2006; Fugard & Stenning, 2013).

In this article we wish to add to the ongoing discussion about the explanatory power of current cognitive models. We adopt a bird's-eye view posing the fundamental question inspired by Richard Feynman's quote: To which degree are state-of-the-art models capable of reflecting what we are fundamentally interested in—the human mind? We investigate this for the exemplary domain of syllogistic reasoning, one of the core fields of human reasoning research.

With a long history of research stretching over 100 years and a state of the art encompassing at least twelve cognitive theories (Khemlani & Johnson-Laird, 2012), syllogistic reasoning lends itself as a demonstrative domain to investigate the levels of understanding research has achieved. In this domain, we define a prediction task querying models for responses to given syllogistic problems. The final model evaluation is performed by comparing the predictions with the actual human responses. To determine the absolute quality of models, we contrast cognitive accounts with data-driven methods from machine learning, namely a set of neural networks based on different features of the data. By comparing cognitive models with the data-driven results, we explore the potential that remains in the field and determine empirical upper bounds of performance to set goals of future modeling endeavors.

A syllogism is a form of categorical assertion consisting of two premises interrelating a set of three terms via quantifiers (All, Some, No, Some ... not). In experimental settings, participants are asked to relate the end terms of the premises (A and C in the example below), i.e., the terms occurring in only one of the premises:

All A are B
All B are C
What, if anything, follows?

Psychological research has shown that human syllogistic reasoning does not strictly follow formal logic principles (Wetherick & Gilhooly, 1995). Instead, past research has

*Both authors contributed equally to this manuscript.

produced various theories attempting to explain the cognitive principles underlying syllogistic inferences (Khemlani & Johnson-Laird, 2012). Since the domain is well-defined (taking the arrangement of terms into account, there are 64 distinct syllogistic problems and a total of nine possible responses including "No Valid Conclusion" indicating that the end terms cannot be related based on the premise information), syllogisms are an accessible domain for cognitive modeling to investigate what is assumed to be one of the fundamental concepts of human reasoning.

The remainder of this article is structured as follows. First, we introduce the state of the art in modeling human syllogistic reasoning. Second, we define the predictive modeling task as the foundation of our analysis and introduce the baseline models used to put cognitive model performances into perspective. Finally, we present the results of our analysis and discuss their implications for modeling syllogistic reasoning in particular and cognitive science in general.

Related Work

Traditionally, research on human syllogistic reasoning focuses on investigating deviations between human inferences and normative first order logic (Wetherick & Gilhooly, 1995). Over the course of time, the phenomena of syllogistic reasoning matured and were integrated into theories relating statistical effects such as the figural effect (Bara, Bucciarelli, & Johnson-Laird, 1995) with assumptions about mental representations (e.g., in the Mental Models Theory; Johnson-Laird, 1983) or fundamental principles of cognition (e.g., the Probability Heuristics Model by Chater & Oaksford, 1999).

A meta-analysis (Khemlani & Johnson-Laird, 2012) compiled a list of twelve contemporary theories along with the corresponding sets of derived conclusions for each syllogism. By comparison with a set of "liable pooled conclusions", i.e., a dichotomization based on which responses were selected by at least 16% of participants, they performed an analysis assessing how well individual theories were able to predict human responses. Employing classification metrics (hits, misses, correct predictions), the authors concluded that no single model clearly outperformed the others. Instead they found that depending on the metric of choice, all models exhibited distinct strengths and weaknesses rendering a conclusive ordering based on performance difficult.

More recent work leveraged the differences in predictive properties of heuristics for syllogistic reasoning by constructing portfolios exploiting the strengths while avoiding the weaknesses of individual models (Riesterer, Brand, & Ragni, 2018). We showed that the predictive accuracy of the resulting composite model (43%) clearly outperformed individual models (ranging between 37% and 18% for the best and worst cognitive model, respectively). In contrast to the meta-analysis discussed above, we directly based our analysis on individual responses instead of aggregates. The resulting accuracies demonstrated lacking capabilities of heuristic models when confronted with an individual prediction task.

This shift in perspective from modeling population data via pooled conclusions to modeling individual responses is motivated by the fact that the core objective of modeling human reasoning is the development of functionally equivalent computational formalisms capturing the essence of the processes driving human inferences. In today's research on syllogistic reasoning, process-driven performance analyses directly on the level of individuals are scarce. Especially in light of recent work in statistics showing that group-to-individual generalizability is limited if not impossible for parts of psychology and other empirical fields of science (Molenaar, 2004; Fisher et al., 2018), modeling individual data directly will become unavoidable.

In the following analyses, we investigate the potential remaining in the field by contrasting cognitive models with data-driven approaches in a prediction scenario focusing on individual human responses. It is important to note that the following work is not targeted towards model assessment in the traditional sense, but a comparison with methods that are expected to yield an upper bound for predictive performance.

Method

In this section we present the core modeling task of this article: predicting individual responses for given syllogistic reasoning problems. As the foundation for our evaluation we rely on a dataset supplied with the *Cognitive Computation for Behavioral Reasoning Analysis* (CCOBRA) Framework[1] consisting of 139 participants responding to the full set of 64 syllogisms by selecting which of the nine conclusion options could be followed from the premises. The model evaluation was performed in a leave-one-out crossvalidation setting where for each subject to be predicted, the models were fitted using the remaining 138 participants as training data. All code and data required for the analyses are made public on GitHub[2].

The Predictive Modeling Problem

The modeling problem is defined as the task to generate a conclusion for a given syllogism. More formally, the goal is to find a function $f : \mathcal{X} \rightarrow \mathcal{R}$ which transforms a problem input $x \in \mathcal{X}$ into a response $r \in \mathcal{R}$, where $\mathcal{X}$ and $\mathcal{R}$ correspond to the sets of 64 syllogistic problems and nine possible conclusions, respectively. Models are finally evaluated based on their predictive accuracy, i.e., the proportion of correct predictions on a given evaluation dataset. In sum, the modeling problem can be formulated in terms of an optimization problem for a prediction function $f(x)$ dependent on input x (syllogistic problem). The optimization procedure maximizes an accuracy score h, e.g., hits, dependent on the prediction $f(x_t)$ for problem x_t and target output y_t (human response) where t identifies the position in the experimental sequence for a dataset of size N:

$$\max_f \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} h(f(x_{i,t} | x_{i,1}, \ldots, x_{i,t-1}; y_{i,1}, \ldots, y_{i,t-1}), y_{i,t})$$

[1] https://github.com/CognitiveComputationLab/ccobra

[2] https://github.com/nriesterer/iccm-neural-bound

This problem definition has properties which are beneficial for cognitive modeling. First, it relies on a highly descriptive performance metric with a close connection to modern machine learning (error reduction). Consequently, good performance results (evaluated on unseen test data) are likely to translate to a sensible estimate of performance in application contexts. Second, the performance metric stretches over a clearly defined range of values between all misses (0%) and perfect prediction (100%) allowing for an assessment of absolute performance. The higher the score, the better a model is capable of approximating human reasoning behavior. The modeling task can be considered solved only if performance converges towards 100%. Finally, and arguably most importantly, it directly uses the data recorded in experiments without introducing the risk of misinterpretation due to making statements about populations or "average" reasoners which might not even exist (Miller et al., 2002).

Cognitive Models for Syllogistic Reasoning

As a starting point for our analysis, we relied on the prediction table reported in Khemlani and Johnson-Laird (2012, Table 7). To compile this list of predictions, Khemlani & Johnson-Laird went to great lengths collecting the most up-to-date versions of the respective approaches while maintaining close communication with the theories' inventors or current maintainers.

Unfortunately, however, the simplicity stemming from organizing model predictions in such a static tabular form fails to capture the intricacies of some methods (e.g., Baratgin et al., 2015). As a result, one should treat these representations as baselines for cognitive models' performances instead of comprehensive accounts reflecting their theoretical merit. Still, since prediction-oriented implementations of syllogistic models are rare, and custom implementation introduces the risk of integrating incorrect assumptions stemming from misconceptions about a theory's intent, we rely on the data from Khemlani and Johnson-Laird (2012) to obtain a conservative estimate of the general performance of cognitive models.

Baseline Models for Syllogistic Reasoning

In order to put the predictive performances of cognitive models into perspective, we introduce a set of baseline models. The *Random* model assumes a uniform distribution over the nine syllogistic responses. When queried for a response, one out of the nine options is randomly sampled from a uniform distribution with probabilities of $1/9$. This model serves as a random baseline all models are expected to exceed.

On the upper end of the performance spectrum, we provide the *Most-Frequent Answer* (MFA) model which computes the response distribution per syllogism from given training data. Predictions are generated by returning the response with highest probability mass (ties are resolved by uniform sampling). Since the predictive modeling scenario forces models to generate a single response to a given syllogism, the MFA is the optimal strategy when no information about the individual reasoner is provided.

Neural Models for Syllogistic Reasoning

To answer the question about remaining potential in the field of human syllogistic reasoning we need to provide upper bounds of performance. Since it is not trivially possible to quantify the numerous noise components in the data which stem from inconsistent responses or highly individual inference strategies, we focus on providing empirical upper bounds obtained from data-driven methods from machine learning. While not offering explanatory insight, the resulting accuracies give an indication about which proportion of the data can be successfully predicted by following the structural properties of the data. In particular, we introduce three neural networks focusing on three different perspectives of the predictive modeling problem. Even though neural networks are severely limited with respect to providing high-level explanation for cognitive processes, they have proven to be capable of achieving high levels of performance over the course of the last years and are suitable candidates for obtaining information about the potential remaining in the field.

The first neural network model is a *Multilayer Perceptron* (MLP), a standard feed-forward neural network featuring a topology of 12-256-256-9, i.e., a twelve-dimensional input consisting of three blocks of four bits each for the onehot-encoded quantifiers and figure[3], which is fed into two hidden layers of dimensionality 256 equipped with rectified linear activation units, and finally into the nine-dimensional output layer which indicates the generated response. The model is initially trained by providing syllogistic problems and corresponding human responses, and is optimized using the Adam optimizer (Kingma & Ba, 2014) with mean squared error as the loss function. After a prediction is obtained, the model is supplied with the true response in order to allow for an adaption to individual reasoning processes. This adaption step is realized by training the model for an additional epoch using the new datapoint.

Second, a *Recurrent Neural Network* (RNN) is employed, which explicitly integrates temporal dependencies into the conclusion generation process (for a conceptual introduction see Elman, 1990). The model features a 12-64-64-9 topology consisting of the twelve-dimensional inputs, two recurrent *Long Short-Term Memory* (LSTM) layers (Hochreiter & Schmidhuber, 1997), and the nine-dimensional outputs. Again, the model is trained using Adam, but uses categorical entropy as the error function (Deng, 2006). This model does not incorporate inter-individual differences. However, by actively modeling the task sequence, it is technically able to identify sequence effects which may be beneficial features for the prediction generation process.

Finally, a *Denoising Autoencoder* is applied which frames the predictive modeling problem as a reconstruction task. Similar to the domain of image restoration in which autoencoders have successfully been applied (Xie, Xu, & Chen, 2012), we supply the model with incomplete data about a reasoner.

[3] E.g., "All A are B; All B are C" is $(1,0,0,0,1,0,0,0,1,0,0,0)$, "Some B are A; Some B are not C" is $(0,1,0,0,0,0,0,1,0,0,1,0)$

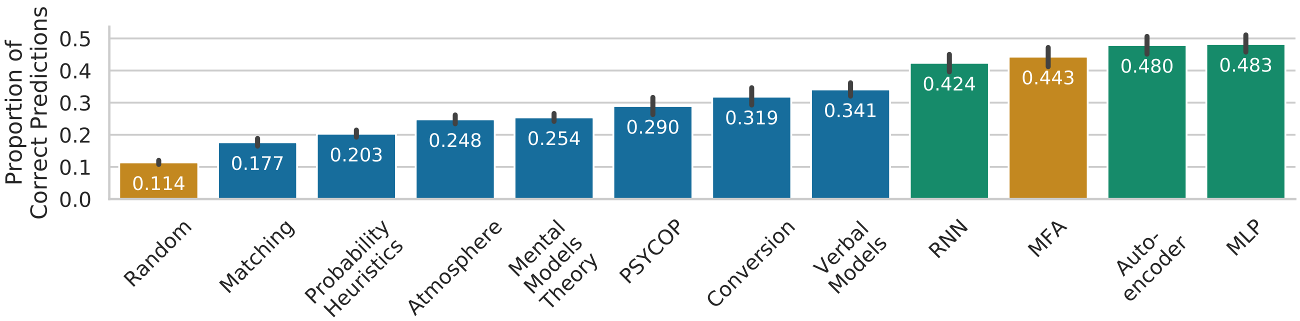


Figure 1: Predictive performance of the models for human syllogistic reasoning. Cognitive models are depicted in blue, baseline models in orange, and neural networks in green. Error bars denote 95% confidence intervals.

The goal of the model is to correctly fill in the blanks. This model is implemented as a 576-2000-576 network featuring a 576-dimensional input obtained by concatenating the onehot encoded responses of the 64 syllogistic problems. As such the inputs represent an individual reasoner's profile. In the hidden layer, this profile is expanded to a high-dimensional space in which relationships between the input dimensions become explicit. From this intermediate representation, the original input can be decoded again. During training, the model is presented with input vectors manipulated by setting values to zero. By training the model to approximate an identity function between noisy inputs and complete outputs by minimizing the mean squared error via Adam, it learns to associate the available information in a way enabling reconstruction of missing values. Over the course of the model evaluation, the autoencoder collects the individual's responses in the adaption step using completing the originally empty reasoner profile. Over time, it leverages the growing information about the individual continuously improving its predictive accuracy.

Results

Predictive Accuracies

The general evaluation results are depicted in Figure 1. The image shows that all models exceed the random model's predictive accuracy of 11% attesting the ability of models to capture the most basic properties of human syllogistic reasoning. The next block of models encompasses the entirety of the cognitive models spanning a range from 18% to 34%. Verbal Models, the best cognitive model, is followed by a substantial gap of performance to the RNN and more importantly MFA, the model always responding with the conclusion most frequently occurring in the training dataset. This constellation of model performances has a major implication for the state of the art in modeling syllogistic reasoning: There is considerable potential left to improve models even without taking inter-individual differences into consideration.

Going beyond MFA, the adaptive neural networks (autoencoder and MLP) demonstrate a basic capability to capture individual reasoning patterns and exploiting them to boost predictive accuracy. However, within this family of models, differences in performance emerge. Relying on temporal dependencies, the RNN model achieves the lowest accuracy scores falling even short of MFA. Reasons for this could be manifold ranging from the application of an unsuitable model topology to problems emerging from the limited amount of training data. However, a more data-centric argument could be that by increasing the data complexity due to the integration of a temporal axis, the models are presented with a problem that is much more difficult to learn than the basic syllogism-response transformation is. As a result, temporal dependencies, or more precisely sequence effects (Aczel & Palfi, 2016), cannot be recognized and leveraged to boost the predictor's accuracy.

The autoencoder which transforms the modeling problem into a reconstruction task achieves higher accuracies than the RNN exceeding the MFA strategy. It shows that the treatment of responses as some form of reasoning profile is a suitable representation to base predictors on that surpass the application of the MFA strategy.

Finally, the MLP achieves the highest accuracy overall (48%). It demonstrates that an integration of adaption to individual properties of cognition via continuous re-training with the newly obtained information can be successfully applied to boost model performance. This approach is not exclusively tied to neural network approaches but should generalize to arbitrary parameterized models which are fitted to training data.

Training Performance

Analyzing the reasons causing networks to perform poorly on data is a difficult task (Lee, Agarwal, & Kim, 2017). To rule out a network's inability to learn the fundamental properties of the syllogistic reasoning data, we investigate the training procedure illustrating accuracy progression on the training and test data per training epoch.

The accuracy progression of the network models during training is depicted in Figure 2. The blue and orange lines represent the mean accuracies (with the shaded band reflecting the 95% confidence interval) on the training and test datasets, respectively. For the RNN, the rise of the training dataset accuracy beyond 90% suggests that, in principle, the network is able to capture the properties of the training data. However,

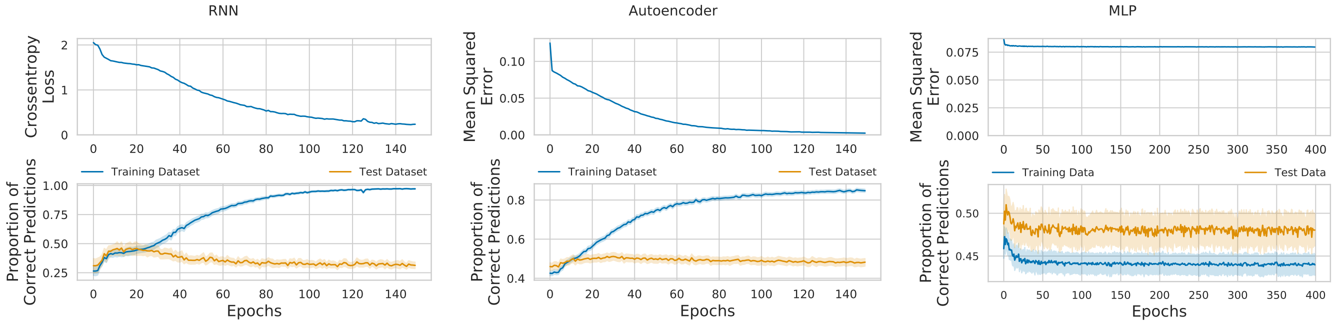


Figure 2: Training progression of the RNN and the autoencoder and MLP. The top plots depict the progression of the raw loss metric used for network optimization. Bottom plots represent the progression of prediction accuracy on training and test data.

the fact that the performance on the test data only rises for a short duration in the beginning of the training process indicates that the learned patterns cannot be generalized successfully to the test instances. The center plot for the autoencoder model paints a similar picture. Even though the effects of overfitting are not as dramatic as for the RNN, training accuracy is clearly improved while damaging the network's generalization capabilities to the test data. An alternative explanation for the superiority of the autoencoder could be that information about individual reasoners are more important for the prediction process or more directly related to specific responses. Finally, the MLP model, despite its predictive capabilities, shows the least amount of learning behavior. After a quick initial bump, the model drops in performance almost instantly and remains constant for the remainder of training. This is most likely due to the limited and inconsistent input and target data. Since in the case of the RNN and autoencoder each training example is high-dimensional and directly incorporates inter individual differences, it is unlikely to observe inconsistencies, i.e., different output for the same input. In contrast, the MLP is fed with 12-bit vectors representing syllogistic problems and produces response predictions for individuals. Since individuals respond differently to the same problems, this data is highly inconsistent and forces the model to adopt a strategy similar to MFA in which an average reasoner is approximated. Classical overfitting is not possible in this scenario.

The observed training performance leads to two conclusions. On the one hand, human syllogistic reasoning appears to follow systematic patterns, which, to some degree, can be leveraged by data-driven methods. The fact that both the RNN and autoencoder are able to learn to fit the training data up to nearly 100% additionally suggests that inconsistencies in the given sequence data (RNN) and reasoner profiles (autoencoder) are minimal. On the other hand, the raw training capabilities of the networks do not generalize well to unseen data. Even though the accuracy on the test data is substantially higher when compared to cognitive models, the training progression shows quick stagnation. Reasons for this could be numerous ranging from problems with respect to data complexity, informational content, or the small size of the dataset used (138 training instances).

In sum, the results show that a current upper bound in performance can be located at a predictive accuracy of roughly 50%. The fact that cognitive models fall significantly lower with a maximum of 35% highlights the potential remaining in the field. Even if the current focus on aggregate evaluation of models is continued, the models should be able to arrive at MFA's performance (44%). The network models demonstrate that by integrating assumptions about individuals even higher predictive accuracies can be achieved. However, even data-driven neural networks stagnate shortly after MFA. While this could be due to technicalities (e.g., network topologies or optimization methods), it could indicate that the purely response-focused data is approaching an upper bound of predictability.

General Discussion

We introduced a predictive modeling task to shift the focus of cognitive model evaluation from relative model selection to a form of model assessment based on absolute performance, i.e., predictive accuracy. In the demonstrative domain of syllogistic reasoning we illustrated that the current state of the art exhibits shortcomings with respect to the quality of model predictions. Without the intention of uncovering individual flaws of specific models, our analysis showed that at most 34% of our data could be successfully predicted by cognitive models. Especially when compared to baseline strategies such as responding with the most frequently chosen answer in the training dataset (MFA), which manages to achieve an accuracy of 44%, this performance is worrisome. For application in real-world scenarios such as in human-agent interaction, syllogistic models are far from being ready for deployment. Even if these theories are, in theory, able to account for core phenomena and statistical effects of syllogistic reasoning, they are of limited use if their assumptions cannot be generalized to useful predictions.

The lingering question is how much potential is left in the domain for future cognitive models to tap into. We introduced a set of neural network models focusing on different properties of the data. Since neural networks are known for being highly capable function approximators, we expected them to provide an upper bound of performance future generations of cognitive models should be expected to achieve. Our results show that the networks were able to significantly outperform the cognitive models arriving at predictive accuracies of up to almost 50% for the adaptive MLP, the overall best predictor. Two of the networks, MLP and the autoencoder were able to leverage information about an individual's reasoning processes to a point that allowed them to surpass MFA. Finding optimal ways to integrate these inter-individual differences into models of cognition is key for achieving high accuracies. The discussion about which features allow for inter-individual differentiation has already begun (Bara et al., 1995; Stenning & Cox, 2006) and should become a central focus of future research in cognitive modeling.

In conclusion, our work illustrated that cognitive models for syllogistic reasoning have potential left for improvement. Currently, the state of the art is unable to reflect the processes underlying human syllogistic reasoning adequately. However, even if they manage to improve, without adjusting the modeling task to focus on individual responses, they will get stuck at the levels of MFA. The network models demonstrate that trivial individualization in the form of training continuation (MLP) is technically successful but does not lead to substantial improvements over MFA. Rather, future models and cognitive theories should integrate inter-individual differences into their core mechanics to give rise to the next level of cognitive models exhibiting properties useful for research (explainability) and application (predictive accuracy) alike.

We strongly feel that the discussed shortcomings originate from a prevailing focus on relative model evaluation and selection as well as statistical analyses and are not limited to the domain of syllogistic reasoning but could potentially generalize to other domains of cognitive modeling. As such, evaluations in terms of absolute performance scores such as predictive accuracies should be added to the toolbox of modelers in order to paint a more comprehensive picture about the capabilities of individual models.

Acknowledgements

This paper was supported by DFG grants RA 1934/3-1, RA 1934/2-1 and RA 1934/4-1 to MR.

References

Aczel, B., & Palfi, B. (2016). Studying the role of cognitive control in reasoning: Evidence for the congruency sequence effect in the ratio-bias task. *Thinking & Reasoning*, *23*(1), 81–97.

Bara, B. G., Bucciarelli, M., & Johnson-Laird, P. N. (1995). Development of syllogistic reasoning. *The American Journal of Psychology*, *108*(2), 157.

Baratgin, J., Douven, I., Evans, J. S. T., Oaksford, M., Over, D., & Politzer, G. (2015). The new paradigm and mental models. *Trends in Cognitive Sciences*, *19*(10), 547–548.

Chater, N., & Oaksford, M. (1999). The probability heuristics model of syllogistic reasoning. *Cognitive Psychology*, *38*(2), 191–258.

Deng, L.-Y. (2006). The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning. *Technometrics*, *48*(1), 147–148.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Fisher, A. J., Medaglia, J. D., & Jeronimus, B. F. (2018). Lack of group-to-individual generalizability is a threat to human subjects research. *Proceedings of the National Academy of Sciences*, *115*(27), E6106–E6115.

Fugard, A. J., & Stenning, K. (2013). Statistical models as cognitive models of individual differences in reasoning. *Argument & Computation*, *4*(1), 89–102.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA, USA: Harvard University Press.

Khemlani, S., & Johnson-Laird, P. N. (2012). Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, *138*(3), 427–457.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, H. S., Agarwal, A. A., & Kim, J. (2017). Why do deep neural networks still not recognize these images?: A qualitative analysis on failure cases of imagenet classification. *arXiv preprint arXiv:1709.03439*.

Miller, M. B., Horn, J. D. V., Wolford, G. L., Handy, T. C., Valsangkar-Smyth, M., Inati, S., ... Gazzaniga, M. S. (2002). Extensive individual differences in brain activations associated with episodic retrieval are reliable over time. *Journal of Cognitive Neuroscience*, *14*(8), 1200–1214.

Molenaar, P. C. M. (2004). A manifesto on psychology as idiographic science: Bringing the person back into scientific psychology, this time forever. *Measurement: Interdisciplinary Research & Perspective*, *2*(4), 201–218.

Riesterer, N., Brand, D., & Ragni, M. (2018). The predictive power of heuristic portfolios in human syllogistic reasoning. In F. Trollmann & A.-Y. Turhan (Eds.), *Proceedings of the 41st German Conference on AI.* Berlin, Germany: Springer.

Stenning, K., & Cox, R. (2006). Reconnecting interpretation to reasoning through individual differences. *Quarterly Journal of Experimental Psychology*, *59*(8), 1454–1483.

Vandekerckhove, J., Rouder, J. N., & Kruschke, J. K. (2018). Editorial: Bayesian methods for advancing psychological science. *Psychonomic Bulletin & Review*, *25*(1), 1–4.

Wetherick, N. E., & Gilhooly, K. J. (1995). 'Atmosphere', matching, and logic in syllogistic reasoning. *Current Psychology*, *14*(3), 169–178.

Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 341–349). Curran Associates, Inc.

Testing a Complex Training Task

Frank E. Ritter[†] **(frank.ritter@psu.edu)**
Farnaz Tehranchi[#] **(farnaz.tehranchi@psu.edu)**
Mat Brener[†] **(mib5292@psu.edu)**
Shan Wang[†] **(sxw820@psu.edu)**
† College of Information Sciences and Technology
# Department of Computer Science and Engineering
Penn State, University Park, PA 16802 USA

Keywords: Cognitive model; Cognitive architecture; Learning and retention curves; Complex task.

Introduction

Figure 1 shows a learning theory based on reviewing multiple learning theories (Kim, Ritter, & Koubek, 2013). This figure shows that learning (curve 1) follows a power law as the learner goes through a declarative stage, a mixed stage, and a procedural knowledge stage.

Retention follows three different curves as well. Retention in the declarative stage (curve 2) falls off fairly rapidly. Retention in the mixed stage (curve 3) falls off less rapidly, and in the procedural stage retention falls off (curve 4) much more slowly. These curves differ because of the three (or two) types of knowledge decay at different rates, with procedural knowledge most robust against decay. These curves have been matched by an ACT-R model of a complex spreadsheet task (Ritter, Tehranchi, & Oury, 2019).

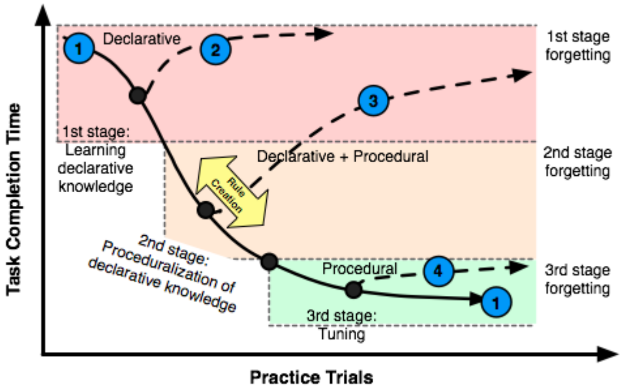


Figure 1. The KRK learning theory in a graph.

To test this set of curves, we needed a complex task that could be learned in an hour but productively practiced for several hours. We also wanted it to be related to troubleshooting and be instrumented.

We considered the Klingon Laserbank Task (KLBT) that has been used to study learning (Bibby & Payne, 1993; Friedrich & Ritter, 2009; Kieras & Bovair, 1984; Ritter & Bibby, 2008), but in 20 trials it can be done in under 10 s by most subjects. We report here a more complex task and an initial test of it.

The Ben-Franklin Radar Task

Ben Bauchwitz for a separate project found a radar that could be made by hobbyists. We modified its schematic to be similar to the KLBT but more complex. The schematic and interface are shown in Figure 2. This device has 36 components compared to the KLBT's 7 components. Colleagues at Charles River Analytics created it as a Unity program.

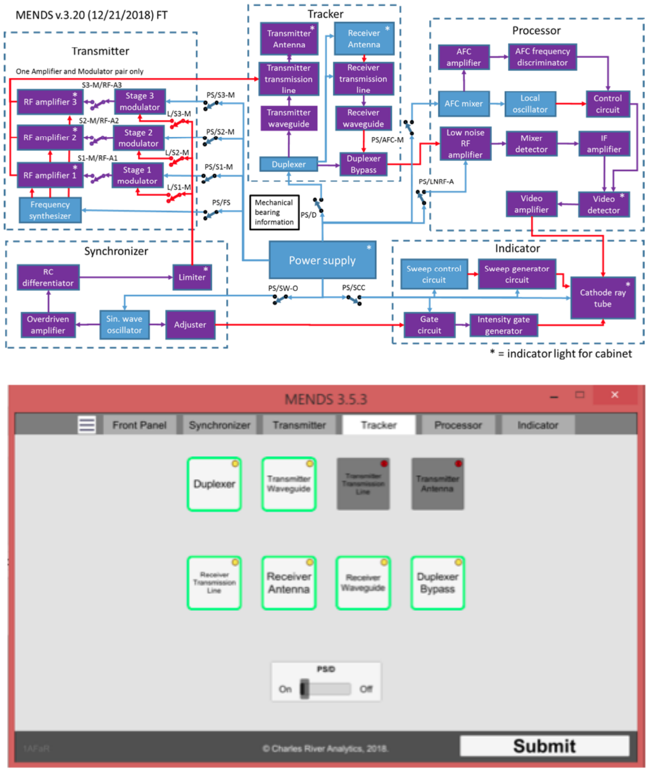


Figure 2. The Ben-Franklin Radar schematic (top) and interface (bottom).

Method

Subject

We had one subject, a 24-y.o. female, first-year master's student, with a BS in Psychology and Mathematics, without any engineering background. She had not seen the schematic before the study.

Materials

Materials included: a printed schematic, the simulation, a D2P2 tutor (Ritter et al., 2013) to teach them both; 1-fault problems for practicing with and without feedback; and 10 recall and 10 recognition questions. The tutor explains each subsystem and guidelines that teach how to do troubleshooting based on the Navy's 6-step troubleshooting approach with example practice problems.

Design and Procedure

In each session, the subject used the tutor and interacted with practice problems for 45 min. In the first four sessions (days 1 – 4), the subject had five minutes to study the printed schematic and then five minutes to draw it from memory. Next, the subject went through the tutor and solved practice scenarios with feedback. At the end of each session, the subject answered the schematic recall and recognition questions and then solved 5 problems. In the fifth session (day 14), the subject answered 10 recognition, 10 recall questions, and 20 troubleshooting questions without feedback.

Results

The subject was able to complete the task and got quite rapid in her responses. Figure 3 shows that over Sessions 1 to 4 her average time for the test problems dropped from 57.5 s to 10.7 s. After a 10 day delay in Session 5, her average time on the first 5 test problems was 13.0 s and on all 20 test problems was 7.9 s. (Her times within sessions followed a learning curve.) Her error rate was consistently low, 4%.

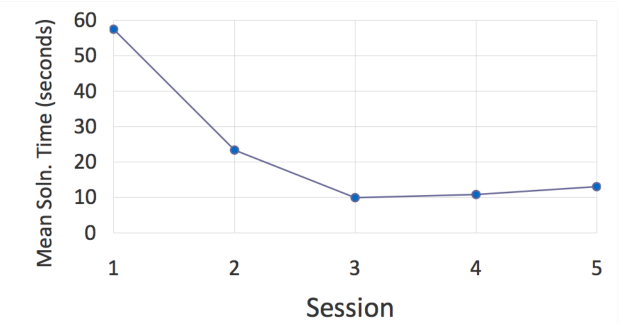


Figure 3. The learning curve for troubleshooting a fault by session (days 1 – 4). Session 5 is the retention test at day 14.

We can see several things from Figure 3. After one practice session, the task was doable and the time to find a fault was about a minute. The task time after three hours of practice also showed that the performance time did not decrease beyond the KLBT task times with 20 practices, so this task is much more complex initially, but approachable and learnable. Compared to the KLBT, the Ben-Franklin Radar task was about 3 times slower in the first test, but after practice, was about the same amount of time.

Another aspect is that the learning curve in sessions 1-4 approximated curve 1 in Figure 1. So, this task might be useful for studying learning and retention.

We saw that after a 10-day break between session 4 and 5, the subject's response time in the 5 test problems did not increase much (as per curve 4, Figure 1). If all 20 test problems are used, however, the average time actually decreased further to 7.9 s. This was on problems without direct feedback (but the interface did provide some indirect feedback). Further examination showed that the trial times kept improving over the 20 problems. So, to study retention of the procedural knowledge, 10 days was just enough to allow forgetting after 4 hours of study—if you do not ask too many questions! This suggests that including a larger number of test trials even without feedback leads to learning in this task, and might not be desirable in a larger study.

Conclusion and Further Research

We found that the task appears to support this study and found some limited support for the theory. We will be running more subjects to test the learning and retention theory.

Acknowledgments

This work was funded by ONR (N00014-15-1-2275). We wish to thank Ben Bauchwitz, Chungil Chae, Cesar Colchado, Josh Irwin, James Niehaus, Billy Manning, Ashley McDermott, William Norsworthy, Jr., Sarah Stager, and Peter Weyrauch for their assistance with developing the experiment. They can be seen as co-authors as well if there was space. Pooyan Doozandeh, Raphael Rodriguez, and Martin Yeh provided useful comments.

References

Bibby, P. A., & Payne, S. J. (1993). Internalisation and the use specificity of device knowledge. *Human-Computer Interaction, 8*, 25-56.

Friedrich, M. B., & Ritter, F. E. (2009). Reimplementing a diagrammatic reasoning model in Herbal. In *Proceedings of ICCM - 2009- Ninth International Conference on Cognitive Modeling*, 438-439. Manchester, England.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning how to operator a device. *Cognitive Science, 8*, 255-273.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (2013). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science, 14*(1), 22-37.

Ritter, F. E., & Bibby, P. A. (2008). Modeling how, when, and what is learned in a simple fault-finding task. *Cognitive Science, 32*, 862-892.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science, 10*(3), Paper e1488.

Ritter, F. E., Yeh, K.-C., Cohen, M. A., Weyhrauch, P., Kim, J. W., & Hobbs, J. N. (2013). Declarative to procedural tutors: A family of cognitive architecture-based tutors. In *Proceedings of the 22nd Conference on Behavior Representation in Modeling and Simulation*, 108-113. Centerville, OH: BRIMS Society.

Learning and Recalling Arbitrary Lists of Overlapping Exemplars in a Recurrent Artificial Neural Network

Damiem Rolon-Mérette (drolo083@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Thaddé Rolon-Mérette (trolo068@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Sylvain Chartier (sylvain.chartier@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Abstract

The mechanisms behind the ability to retrieve all exemplars in a class when presented a specific contextual cue, have puzzled the world of cognition. Various approaches have been used to better understand this concept, especially in the field of artificial neural networks. That being said, very few models can enumerate all exemplars associated to multiple lists in a cognitively plausible way. This is mostly due to the problem of multiple One-to-Many Associations (OMA)s where various exemplars can belong to different lists. To resolve this issue, different approaches have been used; from deep learning and natural language processing to time delayed contextual units. However, none of them is satisfying for a biologically based computational model of cognition. A promising solution is using the class label as context and associates it with each exemplar of the corresponding class. This allows each input to be unique and the problem becomes a standard association one. This strategy has been implemented within the neurodynamic perspective using a bidirectional associative memory. The simulations consisted of learning three arbitrary sequences of various lengths containing multiple intertwined exemplars. Results showed that it was possible to enumerate all associated exemplars from a class simply by presenting the corresponding contextual label. These findings are an important step towards developing cognitively plausible neural implementation of multi-step patterns as well as semantic networks in order to develop generalized artificial intelligence.

Keywords: Cognition; Class enumeration; One-to-many associations; Learning; Memory; Bidirectional associative memory; Feature extraction bidirectional associative memory

Introduction

When presented with a class label, the brain has no difficulty in enumerating all associated exemplar. This cognitive ability is remarkable since learned exemplars are rarely exclusive to a single class; they can have multiple associations (also referred as One-to-Many Associations; OMAs). A simple example to illustrate this would be to enumerate all actions (exemplars) needed to score in a specific sport. In soccer a sequence may resemble, kick off, pass, dribble, run, and kick; while in hockey: face off, pass, dribble, skate and slap; and finally, in basketball: jump ball, pass, dribble, run and throw. Depending on the class of the sport (soccer, hockey or basketball), different exemplars (kick, slap and throw) and identical ones can be found (pass, dribble). Thus, in such a case, when enumerating exemplars from a class, it is easily seen that these exemplars can be associated to a single or multiple classes (OMA). Furthermore, when enumerating a list, it may always follow a specific sequence (ex. opening a door) or, in the case of semantic memory, may not (ex: free association task; Nelson, McEvoy & Schreiber, 2004).

Many formal models in cognitive sciences have been proposed over the years that can accomplish this listing task. Models, such as the Semantic memory models, are known to predict human performances accurately (Jones, Willits, Dennis, & Jones, 2015) but remain limited for neural implementation.

Artificial neural networks have also been used to perform this listing task with most using the multi-layer Perceptron approach (Collobert et al., 2011; Elman, 1990; Jordan, 1997; Neville, 2008). Although these are all interesting models, they are limited in meeting the requirements to be consider biologically based computational models of cognition (O'Reily, 1998). One such class of models that fills these requirements are the Recurrent Associative Memory (RAMs) that belong to the neurodynamic approach (Haykin, 2009). Associative memory consists of learning and storing pairs of identical (auto-association) or different (hetero-association) exemplars. This has been popularized by Hopfield (1982) for auto-association and generalized to Bidirectional Associative Memory (BAM) by Kosko (1988) for hetero-association. Since then, BAMs have seen many modifications allowing them to perform various tasks with better performances; see Acevedo-Mosqueda, Yanez-Marquez & Acevedo-Mosqueda (2013) for a review. Previous studies have shown that RAMs are able to enumerate simple independent lists of exemplars (Chartier & Boukadoum, 2006). However, in the presence of overlapping list of exemplars, like the initial example, they are not able to accomplish the task. This is due to the fact that they must deal with several OMAs. In other words, there are dealing with a relationship instead of a function.

An early solution to solve OMA following findings in cognition (Clarke, 2017; Spillers & Unsworth, 2011; Stoet & Snyder, 2007) was the use of time delayed (context) units

(Elman, 1990). This method integrated previous output(s) with the current input in order to accurately predict the next exemplar in the list (ex. Collobert et al. 2011). Therefore, the one-to-many association was transformed into a one-to-one association. Unfortunately, this solution of delay units (or surrounding context) requires the global knowledge of the number of contextual units prior to learning. Furthermore, it does not really help towards the original task itself; time delayed units (context) are not representative of the class but only of the previous exemplars.

A more interesting solution in machine learning was introduced by Jordan (1997) which used context as a label to modify each exemplar for the enumeration of a given class. Therefore, this contextual label also modifies each exemplar to make them unique without any prior global knowledge.

A second problem may also arise in RAMs if the OMAs are overlapping. In this case there is the possibility that the task becomes a non-linearly separable one. Unfortunately, standard BAMs are not able to solve this unless the model is complexified with a wide range of arbitrary parameters, thus losing its simplistic nature. However, recent studies have shown that an unsupervised version of the BAM can be used to increase the dimensionality of the inputs and therefore, a linear solution can be found when combined with the BAM (ex. Tremblay, Myers-Stewart, Morissette & Chartier, 2013).

Following recent progress in using contextual labels (Rolon-Mérette, Rolon-Mérette & Chartier, 2018a) it is thus proposed to use the class label to make each exemplar unique using a combination of supervised and unsupervised BAMs. This will increase the BAM's versatility and help in learning any number of overlapping OMAs of any length, where exemplars can have any level of correlation and where a non-linear solution is required.

The remainder of the paper is divided as follows: the next section gives brief background of the BAM used in the study; This is then followed by Simulation I, where context is used to show the feasibility of enumerating exemplars from a class and the limits when facing with overlapping OMAs; Simulation II is then presented with a brief background of the unsupervised BAM and how its interaction with the BAM allows the network to perform the desired task; Finally, a short discussion ends this paper.

Bidirectional associative memory

Model description

The model is a modified version of the BAM. Like any neural network, it is defined by an architecture, transmission and learning functions.

Architecture

The BAM's architecture is illustrated in Figure 1. The supervised model has two layers of interconnected units in a bidirectional fashion, where the **W** and **V** layers return information to each other (both acting as a teacher to one another); where *M* and *N* represents the number of units in each layer. The initial patterns are represented by **x**(0) and **y**(0) while the outputs of the network are **x**(*t*) and **y**(*t*) after *t* cycles.

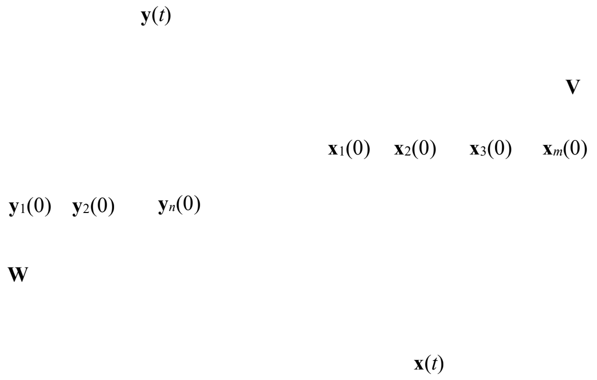


Figure 1: Architecture of the BAM

Output function

The transmission function is defined by equation 1a and 1b:

$$\text{(1a) } \forall i,\dots,N, y_{i(t+1)} = f\big(a_{i(t)}\big) = \begin{cases} 1,\ if\ a_{i(t)} > 1 \\ -1,\ if\ a_{i(t)} < -1 \\ (\delta+1)a_{i(t)} - \delta a_{i(t)}^{3}, \text{Else} \end{cases}$$

$$\text{(1b) } \forall i,\dots,M, x_{i(t+1)} = f\big(b_{i(t)}\big) = \begin{cases} 1,\ if\ b_{i(t)} > 1 \\ -1,\ if\ b_{i(t)} < -1 \\ (\delta+1)b_{i(t)} - \delta b_{i(t)}^{3}, \text{Else} \end{cases}$$

Where *i* is the index unit, *δ* the general transmission parameter and *a* and *b* the activations. These activations are obtained the usual way: **a**(*t*) = **Wx**(*t*) and **b**(*t*) = **Vy**(*t*).

Learning rule

The connection weights for the model are modified following a hebbian/anti-hebbian rule (Chartier & Boukadoum, 2006).

$$\text{(2a) } \mathbf{W}(k+1) = \mathbf{W}(k) + \eta\big(\mathbf{y}(0) - \mathbf{y}(t)\big)\big(\mathbf{x}(0) + \mathbf{x}(t)\big)^{\mathrm{T}}$$

$$\text{(2b) } \mathbf{V}(k+1) = \mathbf{V}(k) + \eta\big(\mathbf{x}(0) - \mathbf{x}(t)\big)\big(\mathbf{y}(0) + \mathbf{y}(t)\big)^{\mathrm{T}}$$

Where **x**(0) and **y**(0) are the initial inputs, *η* is the learning parameter and *k* is a given learning trial. Equation 2a and 2b shows that the matrix weights will converge when **x**(0) = **x**(*t*) or **y**(0) = **y**(*t*). To reduce the simulation time, the number of cycles performed according to equation 1 is usually set to *t* = 1. It is guaranteed that the learning will converge if the learning parameter (*η*) is smaller than the following value (Chartier & Boukadoum, 2006):

$$\text{(3) } \qquad \eta < \frac{1}{2(1-2\delta)\mathrm{Max}[M,N]}, \delta \neq \frac{1}{2}$$

Simulation I: BAM

The general task is illustrated in Figure 2. In order to recreate the task of enumerating exemplars from a class by only presenting its class label, three overlapping list of arbitrary patterns were used. The general goal was to learn all overlapping lists. Two simulation (conditions) were created to better understand the complexity of the task and the feasibility of using a single BAM. The first condition was to establish if labels can be used to solve a simple OMA and enumerate all the exemplars of a class and while the second was to show its limitation with overlapping OMAs. Both conditions are illustrated in Figure 3.

Methodology

Arbitrary alphabetic patterns were used to test the network. Each pattern was a 49-dimensional pixel base pattern where black pixels represent the value of +1 and white pixels -1. Those patterns have the property of showing various levels of correlation (between 0.02 and 0.92). Moreover, they can be naturally partitioned in multiple overlapping classes and are easily recognized by experimenters. Of course, any other arbitrary patterns could have been used without any modification in the results. Two conditions were created.

In condition 1, two sequences of three exemplars (class label followed by two letters) were used to generate an OMA scenario. For both sequences the class label was concatenated to each exemplar of the list allowing exemplar modification and transform the OMA into a one-to-one association (Figure 3a). In condition 2, multiple intertwined lists were used (Figure 3b). The number of lists was set to three for proof of concept while avoiding the simplicity of having a binomial solution. Furthermore, contrary to condition 1, each list was of different lengths and contained multiple OMAs. The first list contained the class label "L" followed by the 26 letters of the alphabet in lowercase. The second sequence was a subset of the first, containing the class label "V" followed by all the vowels in lowercase. Finally, the third sequence was a different subset of the first list containing the class label "C" followed by all the consonants in lowercase. For both conditions, each list was ended by an auto-association on the last exemplar (final attractor).

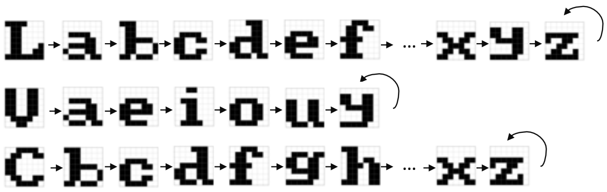


Figure 2: Sequences with class labels ("L", "V" and "C") at the beginning of each class for the overall task.

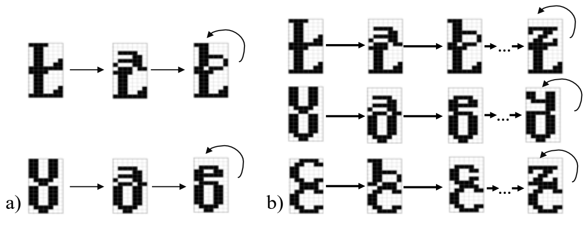


Figure 3: Class labels followed by modified exemplars for condition 1(a) and condition 2 (b).

Procedure

For all simulations, the transmission parameter (δ) was set to 0.2 and the learning parameter (η) respected equation 3. The M and N layers where set to 98 units each, which represents the dimensionality of the combined exemplar (49) and the class label (49). Learning was stopped when the mean squared error (MSE) was lower than 10^{-15} or when 5000 learning trials was reached.

Learning

1. Selection of a list containing both the exemplars and the combined context (Figure 3).
2. Random selection of a pair (**x**(0) and **y**(0)).
3. Computation of **x**(1) and **y**(1) according to the transmission function (equation 1a and 1b).
4. Computation of the weights according to the learning function (equation 2a and 2b)
5. Repeat step 2) and 4) until all the pairs are selected.
6. Repeat step 2) to 5) until the desired MSE or the maximum learning trial are reached.

Recall

1. Selection of an initial contextual label of a given list, **x**(0).
2. Compute **y**(t) in accordance to the transmission function (equation 1a and 1b) until convergence; end of the list.
3. Comparison of the outputted exemplars with the correct ones.
4. Repetition of step 1) to 4) for each contextual label ("L", "V" and "C").

Results

Figure 4 shows the output for each of the conditions when presented with the class label. In condition 1, the output is a clear representation of the desired solution for both associated outputs. By using this approach, it was possible to solve a simple OMA. In condition 2, results showed that appropriate retrieval was unobtainable in the situation of many OMAs. This is not surprising because in such a scenario the task becomes non-linear as well. Therefore, a single BAM will not be able to perform this task. However, by combining the BAM with its unsupervised version, it is possible to overcome this limit. Therefore, in the next simulation, we show such an implementation while keeping the contextual encoding strategy to allow the network to achieve the desired behaviour shown in Figure 2.

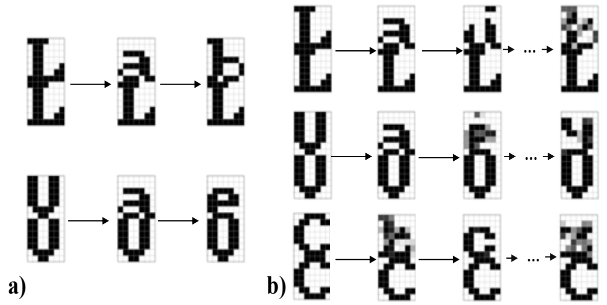


Figure 4: Recall outputs for condition 1(a) and 2 (b)

Simulation II: FEBAM-BAM

In order to use the context to discriminate identical exemplars and to solve non-nonlinear classification, the BAM network is modified to take into account information from its unsupervised version; the Feature Extracting Bidirectional Associative Memory (FEBAM). The FEBAM generates a representation that when combined with the initial input increases the dimensionality and makes the classification problem into a linear one (Tremblay et al., 2013). By still having the same learning and transmission functions and the same general bidirectional architecture, this contributes towards increasing the internal consistency of the overall model.

FEBAM model description

The FEBAM is the unsupervised version of the BAM previously described. The only notable difference between the two is the absence of external (**y**(0)) connections. Consequently, there is no teacher and the model must rely only on one set of inputs, (**x**(0)). Therefore, the goal of this model is to generate the best representation that allows optimal reconstruction of the inputs. It is a process akin to feature extraction (Chartier et al., 2007).

Architecture

The FEBAM's architecture is illustrated in Figure 5. Like the BAM, this model has two layers of interconnected units in a bidirectional fashion, where the **W** and **V** layers return information to each other. As mentioned, there is only one explicit set of connections, **x**(0), used to store information.

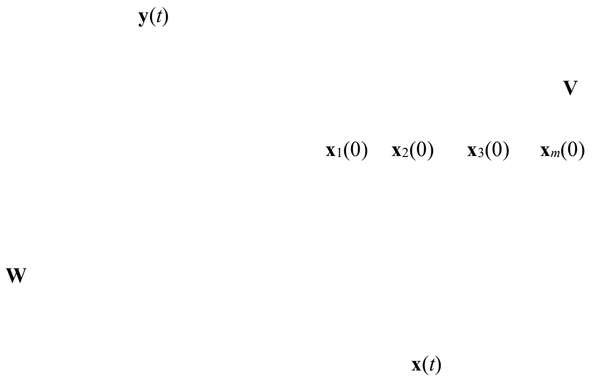


Figure 5: Architecture of the FEBAM

Transmission and learning functions

Both transmission (equation 1a and 1b) and learning functions (equation 2a and 2b) remained the same. However, since **y**(0) is not explicitly given, the information has to circulate a little longer in the network in order to get all needed inputs.

As shown in Figure 6, **y**(0) is obtained by iterating **x**(0) through its corresponding weight connections **W** using the transmission function. Subsequently, **x**(1) is obtained from **y**(0) and finally, **y**(1) from **x**(1). Through weight updates, each **x**(1) and **y**(1) will converge to a solution that will try to best reconstruct its associated initial pattern **x**(0) and/or its representation **y**(0). Thus, in the case where it is impossible for **x**(1) to equal **x**(0) (ex. information compression), weight convergence will only be guaranteed by **y**(1) and **y**(0). The number of units in the **y**–layer determines the dimensionality (level of compression) of the generated representations. The more units there are, the better the reconstruction will be (Giguère et al., 2009).

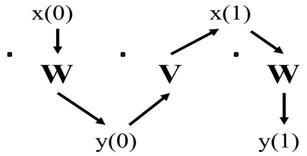


Figure 6: Iterative process used to gather inputs for learning.

FEBAM-BAM Model

Figure 7 illustrates the overall network to accomplish the task where the FEBAM is used to generate features (context).

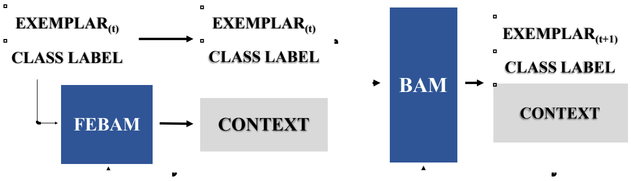


Figure 7: Overall architecture of the FEBAM-BAM

Methodology

The task consisted of learning the same three sequences from simulation I's condition 2 (Figure 3b). This time, exemplars with their class label were fed to the FEBAM first. This allowed the FEBAM to generate features which acted as a unique "signature" for the current exemplar. This representation was then concatenated to the initial input and fed to the BAM for learning. The number of y units in the FEBAM was fixed at a dimension of 98. This was determined in order to increase the probability of success (Rolon-Merette et al., 2018b). The number of y-units can be lower than the number of x-units but for the scope of this study it was not investigated. Finally, in addition to recalling condition 2's lists, a simple noisy (pixel flip) recall task was performed, where the class labels were distorted prior being presented to the FEBAM-BAM. For the noisy recall task, pixel flip ranged from 0 to 50% of the original class label to show the FEBAM-BAM's ability to deal with noise.

FEBAM Learning

The inputs were presented to the FEBAM. To maintain internal consistency, the transmission parameter (δ) and the learning parameter (η) were not change from simulation I. The weights were randomly initialized with values between -0.1 and 0.1 and were updated after one cycle ($t = 1$). Learning stopped when the network achieved a mean squared error (MSE) of less than 10^{-15} or when 5000 learning trials were reached. The learning procedure can be described as follow:

1. Selection of a list containing all three sequences of modified exemplar as seen in Figure 3b).
2. Random selection of a given exemplar from the list to obtain **x**(0).
3. Iteration through the network (as illustrated in Figure 6) using the output function (equation 1a and 1b) to obtain **y**(0), **x**(1) and **y**(1).
4. Computation of weight updates according to the learning rule (equation 2a and 2b).
5. Repetition of steps 2) to 4) until the minimum mean squared error between **y**(0) and **y**(1) or max trials is reached.

Each output was then concatenated to its associated input before being presented to the BAM for learning. The same learning and recall procedure from simulation I was used for the BAM except for the *M* and *N* layers, they were increased to 196 units due to the concatenation.

Results

All three sequences (Figure 2) were successfully learned by the combined FEBAM-BAM model (Figure 7). Furthermore, contrary to condition 2 in simulation I, every exemplar for each sequence is retrieved correctly without any distortion. These results are similar to ones obtained in machine learning (Collobert et al., 2011; Jordan, 1997; Neville, 2008).

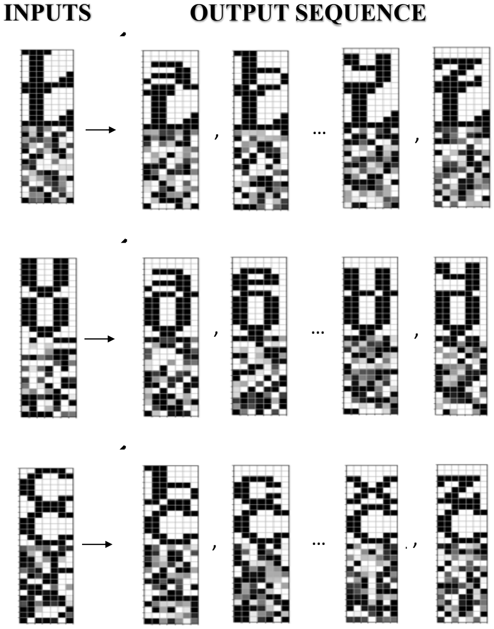


Figure 9: Recall of the learned three sequences.

Likewise, during the pixel flip recall task, correct retrieval was possible for distortion between 0 and 25 %. Figure 10 shows results for a pixel flip of 10% (10 pixels) of the original class label 'L'. This "cleaning" by the FEBAM portion of the FEBAM-BAM allowed to obtain the same retrieval results (Figure 9) while dealing with noisy class labels (inputs).

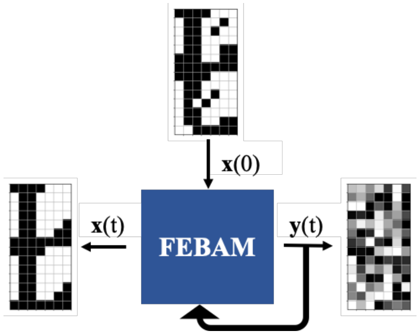


Figure 10: Noisy recall (10 % pixel flip) of class label 'L'

Discussion

In Simulation I, the goal was to learn a simple one-to-many association task (condition 1) by using the context to discriminate the exemplar in a BAM. Results showed that joining the input with fixed contextual information allows the network to solve a simple OMA task using a cognitively plausible neuronal implementation. However, condition 2 showed that this strategy alone is not sufficient if the task contains multiple OMAs. In this last case, a non-linear classification is then required.

To remedy this problem, in simulation II, the BAM was employed in combination with the FEBAM. This addition allowed the network to create its own generated features and when combined with the initial input, allowed to solve non-linear task. The network was able to achieve a perfect learning and recall while maintaining internal consistency; the same transmission, learning functions and the same general bidirectional architecture were used. Furthermore, when faced with distorted exemplars, the model was able to "clean" noisy inputs and reconstruct the appropriate class label while retrieving the associated generated feature. This allowed the FEBAM-BAM to solve the enumeration task despite being presented noisy inputs. This is an important feature towards having a model deal with real world stimuli.

This combination (FEBAM-BAM) is an interesting solution because it avoids the current task specific problem (Marcus, 2018). In approaches where context is given through time delay units (Chartier & Boukadoum, 2006; Collobert et al., 2011; Elman, 1990) the network must know beforehand how many of those units will be necessary for the task, limiting its versatility and plausibility.

That being said, in this model, the proposed mechanism is sequence specific. In other words, although the sequences themselves were arbitrary and could be replaced by any sequences of exemplars, the network outputs will always be in the same order. This is accurate in the case of learning multi-step patterns like motor outputs. However, a future desired property would be the inclusion of more flexibility where the order of outputs is determined from the frequency of occurrence or the success rate of past experience using reinforcement learning. Furthermore, it would be interesting to follow up on the inherent characteristic task (Hattori & Hagiwara, 1998) while using the FEBAM-BAM's ability to modify the exemplars with pseudo-contextual compartments (Clarke, 2017; Spillers & Unsworth, 2011; Stoet & Snyder, 2007). This would open the door towards a cognitively plausible artificial neural-network capable of combining knowledge acquisition and knowledge transfer, increasing even further the model's versatility. Additionally, it is known that the number of y-units must be greater or equal to the number of unique exemplars for feature extraction in the FEBAM (Tremblay et al., 2013). That being said, it would be advantageous to investigate the probability of success for this multi-OMA task while controlling for the dimensionality of the generated context (FEBAM y-unit). This could determine if the number of exemplars in a list or the number of intertwined exemplars have an impact on the number of y-units needed for the non-linearly separable OMA task. Finally, it would be interesting to account for exemplars in a list representing a single exemplar or a whole category in itself. This would allow the model to perform an important semantic memory task while being a simple neuronal model (free association task; Nelson et al., 2004)

In sum, it was shown that a simple bidirectional recurrent associative memory with a hebbian/anti-hebbian learning algorithm is sufficient to solve a complex task requiring the enumeration of all associated arbitrary exemplars from a class by the sole presentation of a class label. These findings are an important step towards developing a neural implementation of semantic networks in order to shift from narrow intelligence to artificial general intelligence (Bengio et al. 2015; Marcus, 2018).

References

Bengio, Y., Lee, D. H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Chartier, S., & Boukadoum, M. (2006). A sequential dynamic heteroassociative memory for multistep pattern recognition and one-to-many association. *IEEE Transactions on Neural Networks*, *17*(1), 59-68.

Chartier, S., Giguère, G., Renaud, P., Lina, J. M., & Proulx, R. (2007, August). FEBAM: A feature-extracting bidirectional associative memory. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on* (pp. 1679-1684). IEEE.

Clarke, A. (2017). Top-down cognitive influences on object recognition: a commentary on Perry and Lupyan (2016). *Language, Cognition and Neuroscience*, *32*(8), 947-949.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(aug), 2493-2537.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179-211.

Giguère, G., Chartier, S., Proulx, R., & Lina, J. M. (2007, January). Creating perceptual features using a BAM-inspired architecture. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 29, No. 29).

Hattori, M., & Hagiwara, M. (1998). Multimodule associative memory for many-to-many associations. *Neurocomputing*, *19*(1-3), 99-119.

Haykin, S. (2009). Neural networks and learning machines, 3rd ed., Copyright by Pearson Education. *Inc., Upper Saddle River, New Jersey*, *7458*.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554-2558.

Jones, M. N., Willits, J., Dennis, S., & Jones, M. (2015). Models of semantic memory. *Oxford handbook of mathematical and computational psychology*, 232-254.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology* (Vol. 121, pp. 471-495). North-Holland.

Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, man, and Cybernetics*, *18*(1), 49-60.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint* arXiv:1801.00631.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, *36*(3), 402-407.

Neville, R. (2008). Third-order generalization: A new approach to categorizing higher-order generalization. *Neurocomputing*, *71*(7-9), 1477-1499.

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in cognitive sciences*, *2*(11), 455-462.

Rolon-Mérette, D., Rolon-Mérette, T., & Chartier, S. (2018a, July). Distinguishing Highly Correlated Patterns using a Context Based Aproach in Bidirectional Associative Memory. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Rolon-Merette, T., Rolon-Merette, D., & Chartier, S. (2018b). Generating Cognitive Context with Feature-Extracting Bidirectional Associative Memory. *Procedia Computer Science*, *145*, 428-436.

Spillers, G. J., & Unsworth, N. (2011). Variation in working memory capacity and temporal–contextual retrieval from episodic memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(6), 1532.

Stoet, G., & Snyder, L. H. (2007). Task-switching in human and non-human primates: Understanding rule encoding and control from behavior to single neurons. *The neuroscience of rule-guided behavior*, 227-254.

Tremblay, C., Myers-Stewart, K., Morissette, L., & Chartier, S. (2013, July). Bidirectional Associative Memory and Learning of Nonlinearly Separable Tasks. In R. West & T. Stewart (Eds.), Proceedings of the 12th International *Conference on Cognitive Modeling*, Ottawa, Canada, pp. 420-425

Different Brain, Same Prototype? Cognitive Variability within a Recurrent Associative Memory

Thaddé Rolon-Mérette (trolo068@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Damiem Rolon-Mérette (drolo083@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Matias Calderini (mcald052@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Sylvain Chartier (sylvain.chartier@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Abstract

When learning similar stimuli, we tend to group them together. This categorization is a behaviour which all humans share. Yet, the pathways undertaken by the brain differs between individuals. To investigate this phenomenon, a Feature Extracting Bidirectional Associative Memory (FEBAM) was used to generate representations of various grouped stimuli. It was determined that representations created by different FEBAMs were always new. However, the learning behaviour was always the same. The generated representations were always categorized into the right category. Finally, by lowering the size of these representations, prototypes of the categories could be created. Recall tests showed that reconstructed prototypes remained the same across all FEBAMs, even if the representations themselves differed. This shows that although the encoding pathways might differ between individuals, the learned cognitive concepts do not. These findings are promising steps towards better understanding how individuals exhibit common cognitive functionality despite variability in neural activity.

Keywords: Variability; Categorization; Feature-Extraction; Associative Learning; Bidirectional Recurrent Neural Networks, Cognition.

Introduction

There are billions of human beings on this planet and each one of them can understand and share complex concepts such as language, games, music and much more. This common cognitive understanding is mind boggling when individuality is considered. There is no brain that is the same as another, with each containing a unique arrangement of its neural structures and connections (Sporns, Tononi, & Kötter, 2005; Thompson, Schwartz, Lin, Khan & Toga, 1996). When presented with the same learning task, different individuals will exhibit different neural activity (Churchland et al., 2010; Mueller et al., 2013). While perception can change based on certain differences in anatomical structures, surprisingly, this variability does not seem to drastically change an individual's understanding of the world and/or its relationships with others. While the neurological pathways involved are different across individuals, the behaviour remains consistent. In other words, from different neural activities, the same cognitive functionality can be observed. That being said, the mechanisms behind such commonality from variability are yet to be fully understood.

An encouraging avenue to better understand this would be to explore the concept of associative learning and categorization. Associative learning can be seen as linking two or more stimuli together (Rescorla & Wagner., 1972). One of its interesting characteristics is the ability to recall one stimulus when only presented with a partial cue (McCleland, McNaughton & O'reilley, 1995). This process forms the basis of categorization whereas similar patterns are grouped together to form a category (Shields, Rovee & Collier, 1992). However, how these "grouped" patterns are represented in our brain remains a mystery. Are the encoded representations of stimuli different across individuals? If so, do they respect the relationship between stimuli, i.e. correctly categorized? In other words, if stimuli are similar, are their representations also similar?

In cognition, such questions can be explored using formal models (Forstmann, Wagenmakers, Eichele, Brown & Serences, 2011). Specifically, artificial neural networks (ANNs) have been an exciting approach to study various key cognitive concepts such as associative learning and categorization (Mareschal, French, & Quinn, 2000). One of the many interesting properties of ANNs dwells in the initialization of weight connections. By randomly initializing the connection weights, each individual instance of a network will be different, analogous to the variability found in human brains. However, what would be interesting is that different instances of a network would display the same behaviour when presented with the same learning task.

Among ANNs are Recurrent Associative Memories, which are designed to implement associative learning (Acevedo-

Mosqueda, Yáñez-Márquez, & Acevedo-Mosqueda, 2010). Particularly, there is the Feature Extracting Bidirectional Associative Memory, or FEBAM (Chartier, Giguère, Renaud, Lina & Proulx, 2007), which can create perceptual features from input patterns via feature extraction (Rolon-Merette, Rolon-Merette & Chartier, 2018). This property allows the FEBAM of category development (grouping similar patterns together based on their correlation). However, a question remains. When presented with the same stimuli, will the FEBAM always generate new representations and if so, will it exhibit the same learning behaviour? In other words, will the representations be categorized in the same manner even if they are always new? This would shed light on the mechanisms allowing common cognitive functionality found between individuals.

The next section gives a short description of the FEBAM and a cluster analysis, followed by three simulations. In simulation I, it was investigated if the representations created by different instances of the FEBAM are always new. In simulation II, the exemplar categorization was observed with a learning task consisting of grouped patterns. In simulation III, under the same learning task, the size of representations was varied to examine prototype categorization. Finally, this paper ends with a short discussion.

Model

The FEBAM is a completely unsupervised recurrent ANN, meaning it does not have any explicit teacher. The entirety of the model can be described by its architecture, transmission function and learning function.

Architecture

The FEBAM architecture is illustrated in Figure 1. The model has two layers of interconnected units in a bidirectional fashion, where the **W** and **V** layers return information to each other. Contrary to traditional bidirectional associative memories, there is only one explicit connection, **x**(0), to allow the network to perform feature extraction.

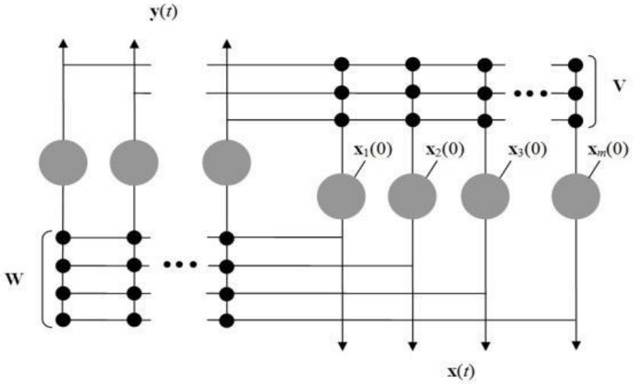


Figure 1: Architecture of the FEBAM

Output function

The transmission function is defined by the Equations 1a and 1b:

$$\text{(1a) } \forall i, \dots, N, y_{i(t+1)} = f\left(a_{i(t)}\right) = \begin{cases} 1, \; if \; a_{i(t)} > 1 \\ -1, \; if \; a_{i(t)} < -1 \\ (\delta+1)a_{i(t)} - \delta a_{i(t)}^3, \text{Else} \end{cases}$$

$$\text{(1b) } \forall i, \dots, M, x_{i(t+1)} = f\left(b_{i(t)}\right) = \begin{cases} 1, \; if \; b_{i(t)} > 1 \\ -1, \; if \; b_{i(t)} < -1 \\ (\delta+1)b_{i(t)} - \delta b_{i(t)}^3, \text{Else} \end{cases}$$

Where N and M are the total number of units in each layer, i is the index unit, δ is the general transmission parameter and a and b are the activations. These activations are obtained the following way: $\mathbf{a}(t)=\mathbf{W}\mathbf{x}(t)$ and $\mathbf{b}(t)=\mathbf{V}\mathbf{y}(t)$.

Learning rule

The connection weights are modified following a hebbian/anti-hebbian rule:

$$\text{(2a) } \mathbf{W}(k+1) = \mathbf{W}(k) + \eta\left(\mathbf{y}(0) - \mathbf{y}(t)\right)\left(\mathbf{x}(0) + \mathbf{x}(t)\right)^{\mathrm{T}}$$

$$\text{(2b) } \mathbf{V}(k+1) = \mathbf{V}(k) + \eta\left(\mathbf{x}(0) - \mathbf{x}(t)\right)\left(\mathbf{y}(0) + \mathbf{y}(t)\right)^{\mathrm{T}}$$

Where **x**(0) and **y**(0) are the initial inputs, η is the learning parameter and k is a given learning trial. Equation 2a and 2b shows that the matrix weights will converge when **x**(0) = **x**(t) or **y**(0) = **y**(t). To reduce the simulation time the number of iterations was set to t = 1. It is guaranteed that the learning will converge if the learning parameter (η) is smaller than the following value (Chartier & Boukadoum, 2006):

$$\text{(3) } \eta < \frac{1}{2(1-2\delta)\text{Max}[M,N]}, \delta \neq \frac{1}{2}$$

FEBAM learning process

As previously mentioned, in the FEBAM, there is only one explicit connection **x**(0), meaning the **y**(0) inputs are not initially available. Instead, they are obtained after a first iteration through the network. As shown in Figure 2, **y**(0) is obtained by the iteration of **x**(0) through its corresponding weight connections **W** using the transmission function. Subsequently, **x**(1) is obtained from **y**(0) and finally, **y**(1) from **x**(1). Through the weight updates, each **x**(1) and **y**(1) will converge to a solution that will try to best reconstruct its associated initial pattern **x**(0) or its initial output **y**(0). Thus, in the case where **x**(1) does not equal **x**(0), weight convergence will be granted by **y**(1).

The number of units in the **y**–layer determined the dimensionality (size) of the generated representation.

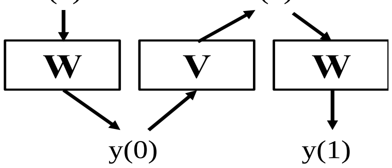


Figure 2: Iterative process for weight updates during learning in the FEBAM.

FEBAM Learning Procedure

The transmission function's parameter (δ) was set to 0.2 and the learning parameter (η) respected Equation 3 for all the simulations. Weights were randomly initialized with values between -0.1 and 0.1. Learning stopped when the network achieved a mean squared error (MSE) of less than 10^{-10} or when 5000 learning trials was reached. Learning was conducted following this procedure:

1. Creation of a list of inputs respecting preset conditions.
2. Random selection of a given exemplar from the list to obtain $\mathbf{x}(0)$.
3. Iteration through the network (as illustrated in Figure 2) using the output function to obtain $\mathbf{y}(0)$, $\mathbf{x}(1)$ and $\mathbf{y}(1)$.
4. Computation of weight updates according to the learning rule.
5. Repetition of steps 2) to 4) until the minimum mean squared error between $\mathbf{y}(0)$ and $\mathbf{y}(1)$ or maximum trials is reached.

Cluster Analysis

In order to partition the generated representations into categories, k-mean clustering was used. For a chosen number of clusters *k*, the algorithm randomly sets *k* centroids in feature space and assigns each data point to the category of its nearest centroid. The positions of each centroid are then iteratively readjusted such that the within-category distance of the resulting categories is minimized. Lloyds algorithm and K-means++ initialization were implemented with the SickitLearn library on Python (Arthur & Vassilvitskii, 2007; Kanungo, Mount, Netanyahu, Piatko, Silverman & Wu, 2002). The sum of the squared distances between data points and their centroid is presented by distortion. *A priori*, the number of clusters that would most appropriately divide the data cannot be known and its high dimensionality makes it prohibitive to determine it visually. Instead, the elbow method was applied to select the optimal number of clusters (Kodinariya & Makwana, 2013). Cluster analysis was conducted under two different scenarios. Scenario A will be used to examine variability across all FEBAMs (Simulation I and IIIb). Scenario B allows to find the average behaviour of an individual FEBAM (Simulation II and IIIa).

Scenario A

1. Creation of input patterns respecting preset conditions.
2. FEBAM learning as specified in the learning procedure.
3. Repetition of steps 1) and 2) for all FEBAMs.
4. K-Means cluster analysis on generated representations of all FEBAMs at once from step 3).

Scenario B

1. Creation of input patterns respecting preset conditions.
2. FEBAM learning as specified in the learning procedure.
3. K-Means cluster analysis on generated representations of each individual FEBAM.
4. Repetition of steps 1) to 3) for all FEBAMs.
5. Calculate average distortion and number of clusters from step 4).

Simulation I: new representations

The number of different generated features was studied when the inputs were kept constant. The task consisted of three learning conditions of different input patterns and generating their associated representations. In each condition, the patterns were fed to multiple FEBAMs, mimicking the learning process of different individuals. The generated outputs, or representations, were then analyzed with k-means clustering using Scenario A.

Methodology

Three different learning conditions were studied using pixelated bipolar inputs patterns of dimension 50, where black pixels represent the value of +1 and white pixels -1. The "pattern" condition consisted of a single pattern. The "category" condition consisted of two categories of five highly correlated patterns. Each pattern within a category exhibited a correlation of 0.95 and the correlation between patterns of both categories was set to 0.15. Finally, in the "random" condition, ten inputs were generated with low correlations varying from 0.01 to 0.30. All three conditions are presented in Figure 3.

In order to have a good estimate of the behaviour, the input patterns were presented to 1000 different FEBAMs, each with a different set of randomly initialized weight connections. The size of the generated representations was kept constant at a dimension of 50. Finally, for each condition, k-means clustering analysis was conducted on the generated representations of all the FEBAMs at once, as stated in Scenario A.

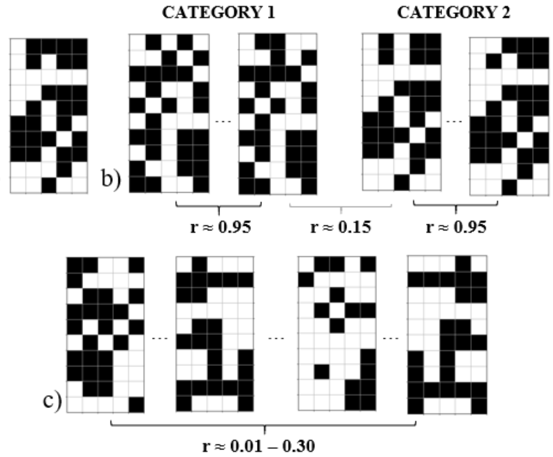


Figure 3: Input patterns for the "pattern" (a), "category" (b) and the "random" (c) conditions.

Results

Different FEBAMs generated different representations when presented with the same pattern(s). Figure 4 illustrates an example of this process. Figure 5 shows the results of k-means clustering for each condition. As the number of clusters created increased, the distortion decreased. However, for all three conditions no elbow was observed.

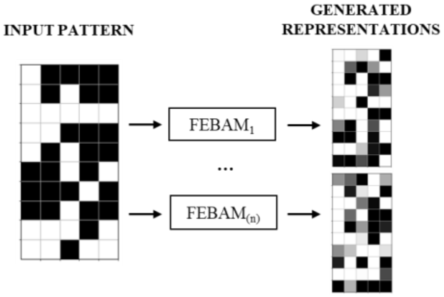


Figure 4: Generating representations for the "pattern" condition with different FEBAMs.

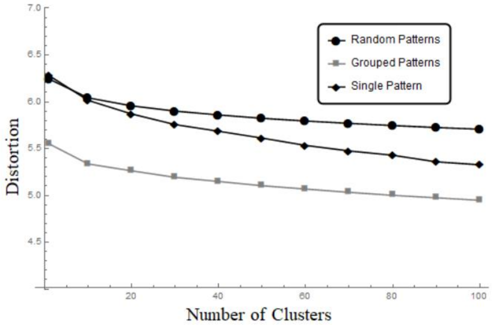


Figure 5: Cluster analysis on representations formed across 1000 different FEBAMs.

Simulation II: Exemplar categorization

In this section, we further investigate whether different FEBAMs respect the same behaviour during exemplar categorization. To do this, we extended the condition 2 of simulation I to five categories. However, in this case, clustering analysis will be conducted on individual FEBAMs and not all at once, as stated is Scenario B.

Methodology

The same method described in simulation I was used to generate input patterns. Here, two to five categories were generated. Each category contained five patterns. The correlation of patterns within the categories was approximately 0.95 and the correlation of patterns between the categories was set to approximately 0.15. The dimensionality of representations (outputs **y** units) was set again to 50. Each set of patterns were presented to 1000 different FEBAMs with the same learning procedure and parameters as previously described. Subsequently, following Scenario B, k-means cluster analysis was conducted on the generated patterns of individual FEBAMs only. Within-category and between-category correlation of generated representations were also examined. Finally, a recall test was performed to verify that patterns were correctly categorized.

Results

In Figure 6, an example of exemplar categorization is presented. In Figure 7, the mean number of clusters and distortion for the 1000 FEBAMs are presented. Results show that the generated representations respected the number of categories found in the input patterns (e.g. two categories, two clusters of generated representations). Furthermore, the average within-category correlation of generated representations was 0.75 and the average between-category correlation was <0.05. Lastly, the recall test yielded a performance of 100% correct pattern categorization.

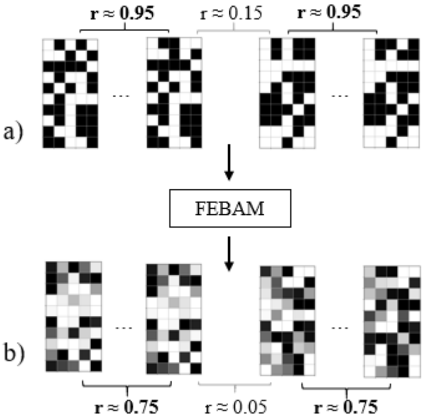


Figure 6: Example of exemplar categorization. Within category (black) and between category (gray) correlation for input (a) and output (b) patterns are presented.

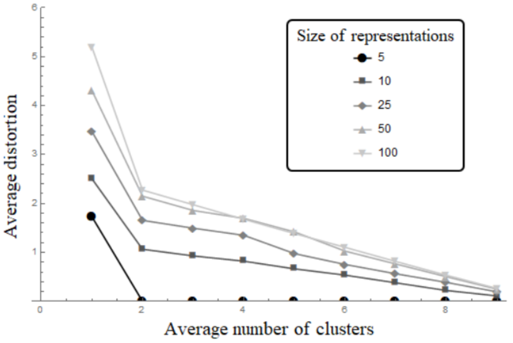


Figure 7: Average distortion and number of clusters for generated representations in function to the number of categories.

Simulation III: Prototype categorization

In this last simulation, the goal was to examine the behaviour of the FEBAM during prototype categorization. A previous study showed that if the dimensionality of the representations is small enough when compared to the number of patterns, prototypes are formed (Giguère, Chartier, Proulx & Lina, 2007). However, the variability of recalled prototypes formed across different FEBAMs was not investigated.

Methodology

Two categories of input patterns, each containing five patterns, were generated in the same fashion as in Simulation Ib and II. The dimensionality of generated representations (number of **y** units) was varied from 5, 10, 25, 50 to 100 dimensions. The patterns were presented to 1000 different FEBAMs using the same learning procedure and parameters as in simulation I and II. Two clustering analyses were conducted.

Simulation IIIa. First, to determine the relationship between distortion and size of representations, k-means cluster analysis was conducted on generated representations from individual FEBAMs. This was done following the procedure described in Scenario B.

Simulation IIIb. Second, to determine the variability of recalled patterns, k-means clustering analysis was conducted on generated representation of dimension 5 and their recalled patterns for all FEBAMs at once. This was done following the procedure described in Scenario A.

Results

Figure 8 shows the first cluster analysis. The average number of clusters and distortion is presented. In all cases, two clusters were formed. Additionally, by lowering the dimensionality of the representations, clusters with lower distortion began to appear. With representations of dimension 5, two clusters accounted for all the distortion, suggesting that prototypes were created.

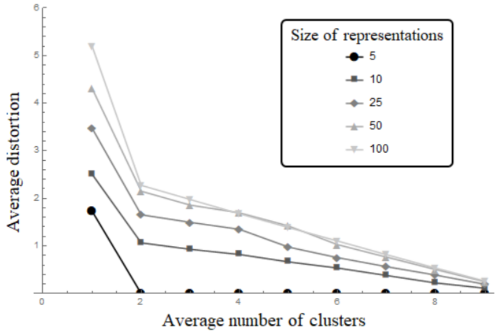


Figure 8: Simulation IIIa. Relationship between the number of clusters and number of y units.

In Figure 9, the second k-means clustering is shown. When looking at the generated representations across the 1000 FEBAMs, it is quickly noted that no clusters were observed. This is consistent with the results from Simulation I, different FEBAMs will always generate different representations. Furthermore, when looking at the recalled patterns, two clusters are shown. However, these accounted for almost all the distortion. This suggests that although coming from 1000 different FEBAMs, the same two patterns were recalled. Figure 10 illustrates this process.

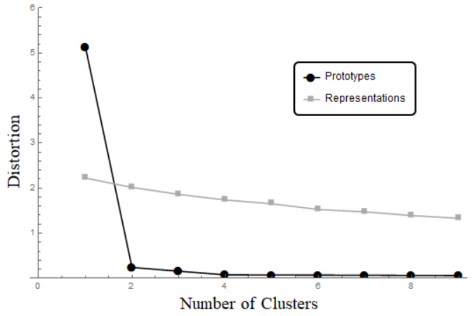


Figure 9: Simulation IIIb. Recalled prototypes and generated representations clustering across 1000 FEBAMs.

Discussion

The goal of this paper was to determine if the FEBAM could shed light on the categorization process found within and between individuals. Results from simulation I showed that when learning the same stimuli, different FEBAMs will generate diverse representations of these input patterns. As seen by the absence of clusters during a k-means clustering analysis. This result was expected since connection weights were initialized randomly.

However, in simulation II, it was found that although different FEBAMs generate different representations, their learning behaviour remained the same. This was first shown by looking at the correlation of generated representations from each FEBAM. The within-category correlation (≈ 0.75) was far greater than the between-category correlation (≈ 0.05). This was further shown with a k-means clustering analysis on the representations. The analysis put forward the fact that the number of clusters corresponded to the number of categories. In addition, recalled patterns were correctly categorized into individual exemplars. These findings are keys since it proposed that the FEBAM will have the same encoding behaviour even if the initial connection weights are different. This also contributes to previous work by showing that both representations and reconstructed patterns are categorized in the same manner (Giguère, Chartier, Proulx & Lina, 2007).

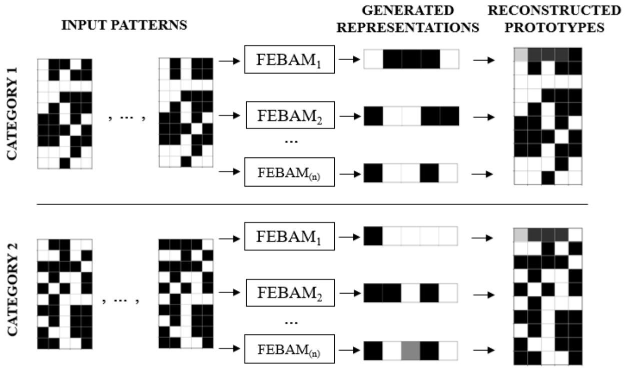


Figure 10: Example of Prototype categorization. Two categories of input patterns are presented to (n) different FEBAMs. These generate representations of dimension 5. Although representations are always different, the same two patterns are recalled. These recalled patterns act as a prototype for each input category.

This characteristic was further explored with prototype categorization. Simulation III showed that different FEBAMs constructed the same prototypes even if the stored representations were different. If the size of the representations is equal or lower than the number of patterns of a given category, then the same pattern was always recalled. This recalled pattern was a prototype of all input patterns within a given category. Thus, even if the initial learning conditions and subsequent generated representations are different, the network will still create the same prototypes.

To sum up, this study showed that the FEBAM is a good model for categorization, capable of both exemplars and prototypes encoding while also accounting for individual differences. The findings are a promising step towards better understanding how individuals exhibit common cognitive functionality despite variability in neural activity and may help in defining the optimal conditions to perform a classification task.

Future work could focus on how manipulating weight initialization may influence learning. A change in initial weight connections between different FEBAMs could result in a corresponding change in their generated representations. Furthermore, depending on the size of the network, the FEBAM exhibits different behaviours during reconstruction of the input patterns (prototype or exemplar recall). An interesting property would be to grow (increase y-units) or prune (decrease y-units) the network based on a task. This would help to surpass the current task specific problem and allow the model to be more generalized.

Achnowledgements

This research was partly supported by the Natural Sciences and Engineering Research Council of Canada.

References

Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & Acevedo-Mosqueda, M. A. (2013). Bidirectional associative memories: Different approaches. *ACM Computing Surveys (CSUR)*, *45*(2), 18.

Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual *ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Chartier, S., & Boukadoum, M. (2006). A bidirectional heteroassociative memory for binary and grey-level patterns. *IEEE Transactions on Neural Networks*, *17*(2), 385-396.

Chartier, S., Giguère, G., Renaud, P., Lina, J. M., & Proulx, R. (2007, August). FEBAM: A feature-extracting bidirectional associative memory. *In Neural Networks, 2007. IJCNN 2007. International Joint Conference on* (pp. 1679-1684). IEEE.

Chervyakov, rolA. V., Sinitsyn, D. O., & Piradov, M. A. (2016). Variability of Neuronal Responses: Types and Functional Significance in Neuroplasticity and Neural Darwinism. *Frontiers in human neuroscience*, *10*, 603.

Churchland, M. M., Byron, M. Y., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., ... & Bradley, D. C. (2010). Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, *13*(3), 369.

Forstmann, B. U., Wagenmakers, E. J., Eichele, T., Brown, S., & Serences, J. T. (2011). Reciprocal relations between cognitive neuroscience and formal cognitive models: opposites attract*? Trends in cognitive sciences*, *15*(6), 272-279.

Giguère, G., Chartier, S., Proulx, R., & Lina, J. M. (2007). Category development and reorganization using a bidirectional associative memory-inspired architecture. In *Proceedings of the 8th international conference on cognitive modeling* (pp. 97-102). Ann Arbor, MI: University of Michigan.

Kanai, R., Bahrami, B., & Rees, G. (2010). Human parietal cortex structure predicts individual differences in perceptual rivalry. *Current biology*, *20*(18), 1626-1630.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE *Transactions on Pattern Analysis & Machine Intelligence*, (7), 881-892.

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, *1*(6), 90-95.

Mareschal, D., French, R. M., & Quinn, P. C. (2000). A connectionist account of asymmetric category learning in early infancy. Developmental psychology, 36(5), 635.

McClelland, J. L., McNaughton, B. L., & O'reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, *102*(3), 419.

Mueller, S., Wang, D., Fox, M. D., Yeo, B. T., Sepulcre, J., Sabuncu, M. R., ... & Liu, H. (2013). Individual variability in functional connectivity architecture of the human brain. *Neuron*, *77*(3), 586-595.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, *2*, 64-99.

Rolon-Merette, T., Rolon-Merette, D., & Chartier, S. (2018). Generating Cognitive Representations with Feature-Extracting Bidirectional Associative Memory. *Procedia computer science*, *145*, 428-436.

Shields, P. J., & Rovee-Collier, C. (1992). Long-term memory for context-specific category information at six months. *Child Development*, *63*(2), 245-259.

Sporns, O., Tononi, G., & Kötter, R. (2005). The human connectome: a structural description of the human brain. *PLoS computational biology*, *1*(4), e42.

Thompson, P. M., Schwartz, C., Lin, R. T., Khan, A. A., & Toga, A. W. (1996). Three-dimensional statistical analysis of sulcal variability in the human brain. *Journal of Neuroscience*, *16*(13), 4261-4274.

An Architectural Integration of Temporal Motivation Theory for Decision Making

Paul S. Rosenbloom (Rosenbloom@USC.Edu)
Institute for Creative Technologies & Department of Computer Science, University of Southern California
12015 Waterfront Dr., Playa Vista, CA 90094 USA

Volkan Ustun (Ustun@ICT.USC.Edu)
USC Institute for Creative Technologies, 12015 Waterfront Dr., Playa Vista, CA 90094 USA

Abstract

Temporal Motivation Theory (TMT) is incorporated into the Sigma cognitive architecture to explore the ability of this combination to yield human-like decision making. In conjunction with *Lazy Reinforcement Learning* (LRL), which provides the inputs required for this form of decision making, experiments are run on a simple reinforcement learning task, a preference reversal task, and an uncertain two-choice task.

Keywords: Motivation; cumulative prospect theory; reinforcement learning; cognitive architecture

Introduction

Temporal Motivation Theory (TMT) weaves together threads from economics, decision making, sociology, and psychology concerned with modeling human motivation and its role in decision making (Steel & König, 2006). Although other forms of motivational theories have previously been incorporated into cognitive architectures (Bach 2009; Sun & Wilson, 2010), TMT provides a particularly intriguing point of departure due to how it already integrates together so many critical aspects. Its implementation within an architecture does, however, present several challenges, including the demands it places on how probabilities, utilities, and time are represented and processed.

The work described here incorporates TMT into the Sigma cognitive architecture (Rosenbloom, Demski & Ustun, 2016), which is capable of stretching to accommodate its various demands. TMT is then deployed in a *Lazy Reinforcement Learning* (LRL) context – where the policy is computed as needed from a learned fractional representation – for use in determining the values of actions/operators considered for selection. This is not the only context in which TMT could be applied. For example, it is likely also relevant in projection, a context that may better match a number of experimental setups, but reinforcement learning was chosen as the initial target due to its centrality in procedural learning.

The resulting combination was explored in a simple RL task plus two tasks that characteristically reveal non-rational choice behavior in humans: a preference reversal task and an uncertain two-choice task.

In the remainder of this article, the relevant aspects of TMT, the TMT/LRL combination, and Sigma are first introduced, and then followed by the implementation of TMT/LRL in Sigma, experimental results and a conclusion. The core result concerns how TMT can be integrated into a cognitive architecture to yield results that, at least qualitatively at this point, enable producing some of the major phenomena that motivate this theory.

Temporal Motivation Theory (TMT)

Temporal Motivation Theory arose as a combination of four prior theories from across multiple disciplines. *Picoeconomics* models the undervaluing of future rewards via hyperbolic discounting rather than classical exponential discounting (Ainslie, 1992). *Expectancy Theory* specifies the overall worth of a value in terms of its product with its expectancy, or probability (e.g., Vroom, 1964). *Cumulative Prospect Theory (CPT)* (Tversky & Kahneman, 1992), like *Prospect Theory (PT)* (Kahneman and Tversky, 1979) before it, models an approach/avoidance aspect of behavior by nonlinearly transforming both values (Figure 1; Equation 1) and expectancies (Figure 2; Equation 2), with different weights when positive (i.e., gains) versus negative (i.e., losses). However, it also goes a step beyond PT to handle stochastic dominance and large numbers of outcomes by using cumulative distribution functions (i.e., transforming the cumulative probabilities and then taking differences of the resulting neighbors). *Need Theory* shares some aspects of the previous theories, but also concerns itself with the specific needs that yield rewards. It furthermore proposes that, like the probabilities and utilities in Prospect Theory, the weighting of temporal distances differs for positive versus negative values (e.g., Dollard & Miller, 1950).

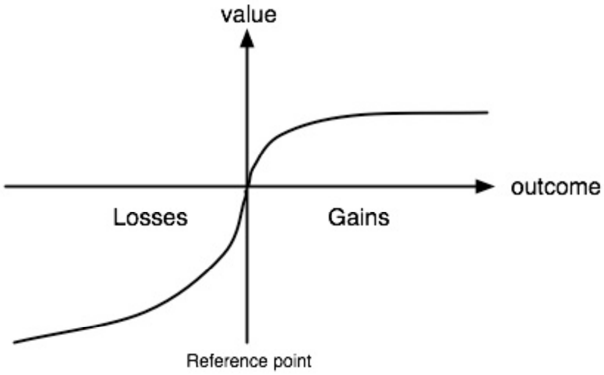


Figure 1: Shape of the (C)PT value transformation.
(from https://upload.wikimedia.org/wikipedia/commons/4/4e/Valuefun.jpg)

$$V_{CPT}^{+} = V^{\alpha}; \qquad V_{CPT}^{-} = -\lambda(-V)^{\beta} \tag{1}$$

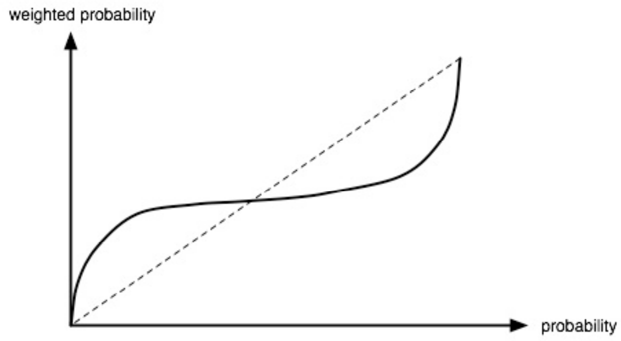


Figure 2: Shape of the (C)PT expectancy transformation.
(from https://upload.wikimedia.org/wikipedia/commons/9/90/Weightingfun.jpg)

$$E_{CPT} = \frac{E^C}{\left(E^C + (1-E)^C\right)^{1/C}} \quad \text{If } V \geq 0 \; C = \gamma \text{ else } C = \delta \qquad (2)$$

Putting this all together, except for the identification of specific needs and how they generate values, yields Equation 3 (Steel & König, 2006). It has two terms, for the sum over the gains and losses, respectively. The numerators blend Expectancy Theory and Cumulative Prospect Theory by multiplying the differences among nonlinearly transformed CPT expectancies (i.e., cumulative probabilities) times the nonlinearly transformed CPT values (i.e., rewards or utilities). The denominators blend Picoeconomics and Need Theory by linearly transforming temporal distances to rewards. All of the parameters other than the additive hyperbolic constant (Z) in the denominators are potentially distinct for gains versus losses.

$$Utility = \sum_{i=1}^{k} \frac{E^{+}_{CPT} \times V^{+}_{CPT}}{Z + \Gamma^{+}(T-t)} + \sum_{i=k+1}^{n} \frac{E^{-}_{CPT} \times V^{-}_{CPT}}{Z + \Gamma^{-}(T-t)} \qquad (3)$$

TMT is also sometimes displayed in a simpler form, as the *procrastination equation* (Steel, 2010), as shown in Equation 4; however, it is Equation 3 that has been implemented in Sigma, along with a lazy form of reinforcement learning that provides the expectancies, values and times needed by it. Although Sigma has the ability to appraise the desirability of situations based on goal specifications (Rosenbloom, Gratch & Ustun, 2015), modeling of specific human(-like) needs has not yet been undertaken; so, instead, the results here are based on whatever rewards are appropriate for the tasks at hand.

$$Motivation = \frac{Expectancy \times Value}{Impulsiveness \times Delay} \qquad (4)$$

TMT and Lazy Reinforcement Learning (LRL)

To use Equation 3 in choosing among actions, it must be applied to each action being considered. This in turn requires tracking probabilities and temporal distances for each reward received as a result of choosing each action. The policy – or *Q function* – learned via normal RL loses most of this information, just tracking a single number – corresponding to the projected discounted future utility – for each action.

An approach has been proposed for incrementally (or recursively) performing hyperbolic discounting so that time delays need not be explicitly tracked (Alexander & Brown, 2010), as with standard exponential discounting; however, it is complex enough on its own to justify putting off its consideration to future work. Still, keeping distinct the probabilities (E) and utilities (V) until they can be non-linearly transformed by CPT before being multiplied adds an extra layer of representational elaboration that is not supported by traditional Q functions.

A pair of algorithmic approaches to integrating (C)PT directly into RL can be found in Andriotti (2009) and L.A., et al. (2016), but here this additional complexity is handled by using *Lazy Reinforcement Learning (LRL)* to acquire a fractional *TMT-Q function* for states and actions, as well as a fractional *TMT-V function* for state values. LRL is like Lazy Q-Learning (Touzet, 2004) in avoiding eager combination of information for RL, but the former does learn distributions over rewards rather than using a full instance-based memory.

Both of these fractional functions explicitly represent a distribution over rewards received at each future temporal distance, while keeping separated values, expectancies and times – as necessary for TMT – and compressing the tree-structured searches into a linear structure that converts all rewards found at one temporal distance into a single distribution over those rewards (Figure 3a).

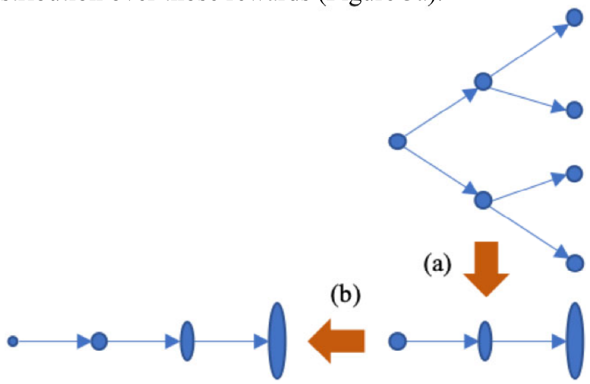


Figure 3: (a) Compressing tree of rewards into sequence of distributions over rewards; (b) Propagating sequence backward one state while shifting distributions forward.

The learning of these fractional functions occurs in an incremental manner analogous to what happens in standard RL, but the whole temporal structure is propagated backwards over action applications to be used in updating the memories for previous states (Figure 3b). In the process, the existing distributions are shifted one step forward in time, to reflect that they now correspond to states one step further in the future, and the temporal location 0 is opened up for learning about the reward at the current state.

Action selection then occurs via Equation 3, by computing Q from TMT-Q and using Boltzmann selection. In Sigma, Equation 3 is implemented in the architecture, whereas LRL involves changing parts of the knowledge traditionally used for RL. This latter works because RL in Sigma, although it leverages architectural mechanisms for things like gradient-descent learning (Rosenbloom et al., 2013) and translation of mental images (Rosenbloom, 2011), is not itself an architectural mechanism (Rosenbloom, 2012).

Sigma and Learning `TMT-V`/`TMT-Q`

Sigma is defined in terms of two architectures, a cognitive architecture at the bottom of Newell's (1980) cognitive band and a graphical architecture at the top of his biological band. The latter started as a straightforward implementation of factor graphs with the sum-product algorithm (Kschischang, Frey & Loeliger, 2001), a general form of probabilistic graphical model (Koller & Friedman, 2009) that yields efficient computation over complex multivariate functions by decomposing them into products of simpler factors, mapping them onto graphs that have bidirectional links among variable and factor nodes, and computing over these graphs via message passing. However, in Sigma this has since been extended, e.g., by allowing unidirectional message passing to support additional cognitive structures such as rules and neural networks (Rosenbloom, Demski & Ustun, 2017).

The functions stored at factor nodes and sent along links are *regular region tensors*; i.e., *n*-dimensional structures in which each cell spans one or more values along each dimension, and every cell along any row in one dimension shares the same boundaries along the others. So, for example, `TMT-V` is a 3D tensor with dimensions for the state, reward, and temporal distance between the state and the reward (Figure 4); and `TMT-Q` is a 4D tensor with the addition of an action dimension. In both, there is a full distribution over the reward for each combination of values of the other dimensions. To slide the rewards to the right, the mental imagery operation of translation is used, while the now empty current (0) time step is initialized to uniform (Figure 5).

Future Time (steps)

Reward		0	1	2	3-*max*
	-1	.5	.1	.4	.333
	0	.25	.6	.5	.333
	1	.25	.3	.1	.333

Figure 4: A nominal 2D slice from `TMT-V` for future reward distributions at a single state. The final temporal regions at the right specify a uniform distribution over all possible but as yet unexplored future times.

Future Time (steps)

Reward		0	1	2	3	4-*max*
	-1	.333	.5	.1	.4	.333
	0	.333	.25	.6	.5	.333
	1	.333	.25	.3	.1	.333

Figure 5: 2D Slice from Figure 4 translated to the right, with the open column (0) initialized to uniform for learning from the perceived reward at the current state.

In Sigma's cognitive architecture, *predicates* are used to define particular types of tensors for which working memory (WM) and/or long-term memory (LTM) factor nodes are to be created. The 3D tensor underlying Figures 4 and 5 is defined, in simplified syntax, as `TMT-V(`***`x:location, f:future, r:reward`***`):1`. There is one argument for each dimension, with a type for each that determines whether it is continuous, discrete (i.e., integers) or symbolic, and what its span is. All of the types are discrete here, but with differing spans. Argument `r` is underlined because the distributions are defined over it; that is, for each region defined by the other dimensions, there is a distribution over the rewards. The `:1` denotes an LTM node should be defined with an initial uniform value of 1 (before normalization). There is no WM factor node here, so there is no temporary latching of values for this tensor. Instead, gradient-descent learning at the LTM node learns reward distributions from experience. Left for future work is consideration of the potential relationship of such LTM functions to Sigma's episodic memory (Rosenbloom,), which also learns histories of values at states from experience.

To define the overall structure of the graph in which such LTM nodes exist, Sigma supports the notion of a *conditional*, which blends the conditionality found in rules and in both probabilistic and neural networks. Conditionals are built from variablized patterns plus functions, with the functions yielding an additional form of LTM node from those in predicates. The patterns are defined over predicates and may take on the unidirectional forms found in rule *conditions* and *actions* and in neural networks, or the bidirectional form found in probabilistic and constraint networks (*condacts*). Figure 6, e.g., shows in simplified form a conditional that projects backward the state reward distributions – i.e., `TMT-V` – while using the mental imagery operation of translation (`f+1`) to convert Figure 4's distribution to Figure 5's.

```
CONDITIONAL TMT-V-Translated
  Conditions: Selected(operator:o)
              Location(x:x)
              Location*Next(x:nx)
              TMT-V(x:nx, f:f, r:r)
  Actions: TMT-V(x:x, f:f+1, r:r)
```

Figure 6: Conditional to translate from Figure 4 to Figure 5. The italicized argument values are variables.

Learning `TMT-V` here occurs via gradient descent at its LTM factor node, as driven by the messages reaching it from the conditional's action. The remainder of the LRL algorithm then includes an additional conditional that yields `TMT-Q` from `TMT-V`, and which is identical to the conditional in Figure 6 except for the action, and a simpler conditional that copies the reward at the current state to location 0 of `TMT-V`.

Computing `Q` from `TMT-Q`

The `Q` function isn't directly learned in Lazy RL, although by providing an LTM function for it, it would be possible to cache an evolving representation of it. Instead, the architecture has been extended to compute Q from TMT-Q at decision time. The algorithm for this in Figure 7 implements TMT but also allows a full space of options: (1) nonlinearly transforming rewards according to PT, CPT, or not; (2) nonlinearly transforming probabilities as in CPT, or in PT, or not; and (3) discounting future rewards hyperbolically (as in TMT), exponentially (as in classical RL), or not. It thus provides a combinatoric space for experimentation.

```
1.  P ⟵ P - min(P)        ; Remove uniform aspect
2.  F,R ⟵ Normalize(F,R)
3.  When Transform?(R) or Transform?(P):
        R ⟵ Shatter(R)  ; Partition into unit regions
4.  When Cumulate?(P):
        P ⟵ Cumulate(P)
5.  When Transform?(P):
        P ⟵ Transform(P)
6.  When Cumulate?(P):
        P ⟵ P – $P_{prev}$     ; Uncumulate P
7.  R ⟵ Expected(P, [If Transform?(R) then
                         Transform(R) else R])
8.  When Discounted = H
        F ⟵ Hyperbolic(F)
9.  When Discounted = E
        F ⟵ Exponential(F)
10. ∑F
11. P ⟵ Scale-Down(P)  ; Scale down extremes
12. P ⟵ Exponentiate(P) ; Ensure values positive
13. Q ⟵ Remove-Unneeded-Slices
```

Figure 7: Algorithm to transform TMT-Q into Q. Rewards (R) may be transformed or not. Probabilities (P) may be cumulated or not and transformed or not. Futures (F) may be discounted hyperbolically (H), exponentially (E) or not.

Several aspects of this algorithm could do with a bit more explanation. In line 1 the removal of the uniform signal, by subtracting from each distribution the minimum probability in it, is not a standard part of TMT or CPT, but is necessary to appropriately use Sigma's learned distributions in transformations that are asymmetric around zero. Without this step, and with the standard loss-averse CPT parameters, decision making would be biased away from actions for which little has been learned, and which thus retain more of their initial uniform distribution. Ultimately, such a bias may prove to be appropriate as an implicit preference for actions about which the system is more certain, but it raises issues for Boltzmann selection in providing a strong bias for exploitation over exploration, so for now this bias is subtracted out, with further investigation of its possibly appropriate use left for future consideration.

The shattering along the reward dimension in line 3 and the later removal of unnecessary slices in line 13 both relate to how Sigma's regular region tensors work. The shattering ensures that there is a separate region for each distinct reward value – even when they have the same probabilities – enabling computations for cumulative probabilities to happen appropriately. The removal of unneeded slices at the end then reaggregates regions when all adjacent pairs of them along any dimension have the same probabilities.

The core of (C)PT is transforming both probabilities (line 5) and rewards (which occurs in line 7, in the process of determining the expected value for each future region). With full CPT, the probabilities need to be cumulated before they are transformed (line 4) and then uncumulated after (line 6). This is then followed, if desired, by discounting, either hyperbolically (line 8) or exponentially (line 9); and summing the expectations over future times (line 10).

The key remaining steps are required by Sigma rather than TMT or CPT. The core is the need to exponentiate the results (line 12) so that Sigma's decision procedure receives a fully non-negative Q distribution. In support of this, distributions with extreme values that would lead to either underflow or overflow when exponentiated are scaled down (line 11).

The elements added in this algorithm to fit TMT into Sigma may affect the exact numerical values yielded, in comparison to pure TMT, but the intent, as evaluated in the next section, is that the same qualitative phenomena will result.

Experiments

Tversky and Kahneman (1992) provided a standard set of parameters for use in CPT: $\alpha=\beta=.88$; $\gamma=.61$; $\delta=.69$; and $\lambda=2.25$. These values are used throughout these experiments, without attempting to tune them further to any differences resulting either from the implementation in Sigma or the specific tasks or models. We are not aware of standard values for the other TMT parameters in Equation 3, so the following values were used: $\Gamma^+=\Gamma^-=1$; and $Z=.5$. Although Z is often shown as 1, it was found to be easier to display preference reversal on a small grid with a smaller value. The one remaining parameter is a value of .95 for exponential discounting, when used.

RL Task

The first experiment used a simple RL task – of finding a goal location in a 1D corridor (Figure 8) – to determine whether RL would still work when TMT is injected into the decision-making process; and, further, to see if there might be any interesting differences among 12 of the variations enabled by the algorithm in Figure 7, omitting only those with no discounting, as they would not learn anything of interest here.

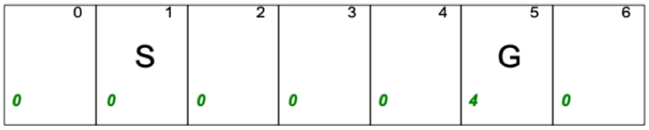


Figure 8: Simple RL Task Domain with starting and goal locations, and a reward of 4 at the goal location.

Each variation was run for 5 repetitions of 200 trials each. The average number of decisions per repetition ranged from 4428 (with either form of expectancy/probability transformation, no value transformation, and exponential discounting) up to 5510 (with no expectancy/probability transformation, a value transformation, and hyperbolic discounting). However, all of the variations did learn, with even the slowest yielding a Q function favoring movement to the right over movement to the left at locations 1-4. The faster cases simply had greater differences between the values for moving right versus moving left.

To explore whether there were any interesting differences among the 12 variations, a three-factor ANOVA was run over the number of decisions per repetition for the three dimensions/factors that define the space of variations. This yielded four results with $p<.05$, for the three single factors plus the interaction between expectancy/probability transformations and discounting strategy.

There are three levels of expectancy/probability transformation, with CPT and PT having almost identical means, of 4911 and 4909, but with the no transformation case jumping up to 5410. There are two levels for value transformation, with transformation (5130) being slightly outperformed by no transformation (5023). There are also two levels of discounting, with hyperbolic discounting (5373) being outperformed by exponential discounting (4780). The one significant pairwise interaction is due to exponential discounting significantly outperforming hyperbolic discounting with expectancy/probability transformations, but performing similarly with no transformation.

Given the rationality, and thus the a priori expectation of optimality, of standard RL – that is, exponential discounting with no transformations – the one real surprise here is the gain over this from expectancy/probability transforms. Although these results are very preliminary, they do suggest further investigation may be worthwhile from an RL perspective.

Preference Reversal

The second experiment concerned a temporal form of *preference reversal* (Ainsley, 1992), an example of humans engaging in non-rational decision making, where shifting a pair of possible rewards further away in time, even when their relative distances to the starting point remain the same, reverses which reward is preferred. A typical example compares two decision situations in which the subject can either receive \$10 on one day versus \$100 one year from that day. If the decision occurs on that first day, then the immediate gratification of \$10 may be preferred to \$100 in a year. However, if the decision occurs 10 years prior to the first day, the \$100 will be preferred. With exponential discounting, a year's worth of discounting has the same proportional impact whether or not that year is now or 10 years in the future. However, with hyperbolic discounting the effect is smaller the further into the future the rewards are.

An analog of this task was developed in a slightly wider version of the 1D corridor in Figure 8, with the start location at 5, and rewards of 1 and 3 either at "near" locations (4 and 7) or "far" locations (2 and 9). For 1 repetition of 200 trials, every variant with exponential discounting failed as expected to exhibit preference reversal, with all showing a strong preference for the larger reward whether near or far. For example, for standard RL – although still in the form of LRL – with no transformations, the left versus right Q values at location 5 were <.14, .86> for near and <.16, .84> for far. For hyperbolic discounting, all cases instead showed preference reversal, except for those with no value transformation. For full TMT, for example, the left versus right Q values at location 5 were <.62, .38> for near and <.41, .59> for far. In the exceptional cases – i.e., with no value transformation – the constant preference for the larger value is likely due to there being a wider difference between their untransformed values versus their transformed values, and thus requiring more distant "far" locations to show preference reversal. In a follow-on experiment with rewards of 2 and 4 and locations of 1 and 10, this case did in fact exhibit preference reversal.

Two-Choice Task

The third experiment compares a choice between a single fixed reward versus a gamble between two rewards with known probabilities. It stresses the non-rational consequences of transforming expectations/probabilities and values/utilities, and thus is particularly relevant to (C)PT.

A narrower adaptation of the 1D corridor in Figure 8 is used here, with a start location of 1, a move left leading to the gamble at location 0, and a move right leading to the fixed reward at location 2. As both payoffs are at the same distance, the form of temporal discounting becomes irrelevant here. However, instead of using the Q function for decision making, the probability of selecting the gamble is set to .9, with only a .1 chance of selecting the fixed reward, so that, even when the gamble is very skewed, such as .99 versus .01, there will be enough experience with the rare alternative.

Figure 9 shows the results for those cases from Table 3 of Tversky & Kahneman (1992), where the uncertain choice has one 0 reward and one positive one (either 50, 100, 200 or 400). The Human data is taken directly from that table for these fifteen data points, whereas the Sigma data is from the simulation, and the TMT data is directly from the equations. It can be seen from this figure that the shapes of the three curves are roughly the same, although both the Sigma and TMT curves are somewhat lower than the Human curve throughout much of the midrange of the probabilities.

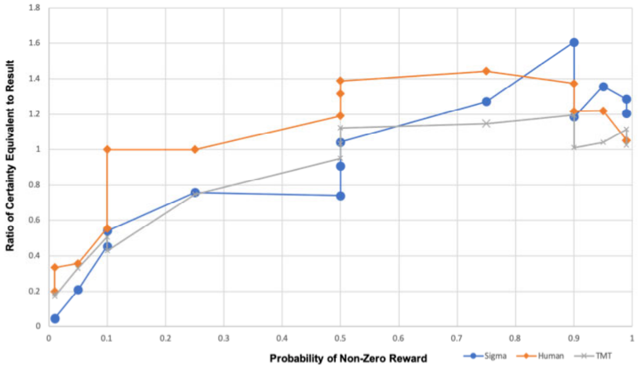


Figure 9: Comparison of Sigma (simulated), Human and TMT (analytical) data for uncertain alternatives of 0 and a non-zero (positive) reward.

Conclusion

In this article, one key aspect of motivation – how it impacts decision making in human(-like) intelligence – has been approached architecturally, by adding Temporal Motivation Theory (TMT) to Sigma's decision procedure. A lazy form of reinforcement learning (RL) that is implemented by modifying Sigma's standard knowledge-driven (plus gradient-descent learning) approach to RL provides the values, expectancies and times required by TMT.

Experiments explored whether this combination could still learn appropriately in a simple RL task, and whether it yields human-like results in preference reversal and two-choice tasks. The answer to these questions is yes, although only qualitatively at this point. The one big surprise was that in the simple RL task, adding in the expectancy/probability transformation improved the learning. Although this is a very preliminary result, it is worth looking into further.

In addition to what has already been mentioned with respect to relevant future work, there are a number of other issues worth further follow up. One is exploring more complex RL tasks that require the learning of longer sequences of future time steps, and thus considering whether Lazy Reinforcement Learning (LRL) continues to be sufficiently efficient. A second is the use of TMT in projection – i.e., lookahead search – as an alternative to its use in reinforcement learning. A third is looking more deeply at a broader range of human tasks, and specifically at whether with appropriate parameter searches good quantitative fits can be produced. A fourth and final topic is incorporating architectural models of specific motivations/needs that can appropriately and automatically provide many of the rewards/values required for human-like decision making.

Acknowledgments

The work described in this article was sponsored by the U.S. Army. Statements and opinions expressed may not reflect the position or policy of the United States Government, and no official endorsement should be inferred.

References

Ainslie, G. (1992). *Picoeconomics: The Strategic Interaction of Successive Motivational States within the Person.* New York: Cambridge University Press.

Alexander, W. H. & Brown, J. W. (2010). Hyperbolically discounted temporal difference learning. *Neural Computation*, *22*, 1511-1527.

Andriotti, G. K. (2009). *Prospect Theory Multi-Agent Based Simulations for Non-Rational Route Choice Decision Making Modelling*. PhD Thesis, University of Würzburg.

Bach, J. (2009). *Principles of Synthetic Intelligence, Psi: An Architecture of Motivated Cognition.* New York, NY: Oxford University Press.

Dollard, J., & Miller, N. E. (1950). *Personality and Psychotherapy: An Analysis in Terms of Learning, Thinking, and Culture.* New York, NY: McGraw-Hill.

Kahneman, D. & Tversky, A. (1979). Prospect Theory: An Analysis of Decision under Risk (PDF). *Econometrica*, *47*, 263–291.

Koller, D. & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press.

Kschischang, F. R., Frey, B. J. & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*. *47*, 498-519.

L.A., P., Jie, C., Fu, M., Marcus, S. & Csaba, S. (2016). Cumulative Prospect Theory meets Reinforcement Learning: Prediction and Control. In *Proceedings of Machine Learning Research*, *48*, 1406-1415.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Rosenbloom, P. S. (2011). Mental imagery in a graphical cognitive architecture. *Proceedings of the 2nd International Conference on Biologically Inspired Cognitive Architectures* (pp. 314-323).

Rosenbloom, P. S. (2012). Deconstructing reinforcement learning in Sigma. *Proceedings of the 5th Conference on Artificial General Intelligence* (pp. 262-271).

Rosenbloom, P. S., Demski, A., Han, T., & Ustun, V. (2013). Learning via gradient descent in Sigma. *Proceedings of the 12th International Conference on Cognitive Modeling* (pp. 35-40).

Rosenbloom, P. S., Demski, A. & Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*, 1-103.

Rosenbloom, P. S., Demski, A. & Ustun, V. (2017). Toward a neural-symbolic Sigma: Introducing neural network learning. *Proc. of the 15th Annual Meeting of the International Conference on Cognitive Modeling.*

Rosenbloom, P. S., Gratch, J. & Ustun, V. (2015). Towards emotion in Sigma: From Appraisal to Attention. *Proceedings of the 8th Conference on Artificial General Intelligence* (pp. 142-151).

Steel, P. (2010). *The Procrastination Equation: How to Stop Putting Things Off and Start Getting Stuff Done*. Toronto, Ontario: Random House Canada.

Steel, P. & König, C. J. (2006). Integrating Theories of Motivation. *Academy of Management Review*, *31*, 889-913.

Sun, R. & Wilson, N. (2010). Motivational processes within the perception-action cycle. In V. Cutsuridis, A. Hussain & J. G. Taylor (Eds.), *Perception-Action Cycle: Models, Architectures, and Hardware*. New York, NY: Springer.

Touzet, C. F. (2004). Distributed lazy Q-learning for cooperative mobile robots. *International Journal of Advanced Robotic Systems*, *1*, 5-13.

Tversky, A. & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, *5*, 297–323.

Vroom, V. H. (1964). *Work and Motivation.* New York, NY: Wiley.

(A)symmetry × (Non)monotonicity: Towards a Deeper Understanding of Key Cognitive Di/Trichotomies and the Common Model of Cognition

Paul S. Rosenbloom (Rosenbloom@USC.Edu)
Institute for Creative Technologies & Department of Computer Science, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094 USA

Abstract

A range of dichotomies from across the cognitive sciences are reduced to either (a)symmetry or (non)monotonicity. Taking the cross-product of these two elemental dichotomies then yields a deeper understanding of both two key trichotomies – based on control and content hierarchies – and the Common Model of Cognition, with results that bear on the structure of integrative cognitive architectures, models and systems, and on their commonalities, differences and gaps.

Keywords: Dichotomies; control; memory; learning; Common Model of Cognition; cognitive architectures.

Introduction

The cognitive sciences embody many dichotomies, with a broad range of work focused on either making a case for one side versus the other of individual dichotomies or on finding a hybrid approach that spans both sides. Here, the focus is on two general clouds of dichotomies – one that is fundamentally reducible to (*a*)*symmetry* and the other to (*non*)*monotonicity* – with the overall aim of understanding them better both individually and jointly. (A)symmetry concerns whether processing – whether conceived of as memory access, derivation, inference or computation – is valid in a single direction versus in arbitrary directions. (Non)monotonicity in its essence concerns whether processing accumulates results versus alters them.

These are not necessarily the most familiar formulations of either dichotomic cloud, but each fundamentally captures the nature of its own cloud in a manner that enables a simple definition and a clear path for mapping the other dichotomies from the same cloud onto it. Although such mappings may at times lose nuances, the main message concerns the commonality at their heart rather than the range of subtleties.

Once the clouds are reduced to the two elemental dichotomies, their cross product yields a 2×2 framework that enables additional analyses. It is first applied to two key cognitive trichotomies that are based, respectively, on control hierarchies – including one implicit in AlphaZero, a system that learns to best humans at challenging board games (Silver et al., 2018) – and content hierarchies. Each trichotomy spans only three of the four cells but together they span all four.

The framework is then applied to the Common Model of Cognition – an attempt to build a community consensus over the structures and processes that define a human-like mind – plus three cognitive architectures that heavily influenced its initial form (Laird, Lebiere & Rosenbloom, 2017): ACT-R (Anderson, 2007), Soar (Laird, 2012) and Sigma (Rosenbloom, Demski & Ustun, 2016). The initial focus here will be on memory and control, with results highlighting one of the major capabilities missing from the Common Model, while clarifying the distinct ways the three architectures span the di/trichotomies. This is followed by an analysis of learning that also includes AlphaZero.

The methodology here is akin in general to the one behind the Common Model – based on abstract analysis and synthesis rather than detailed experiments and models – but the goal is to provide a start at a yet deeper understanding of key parts of cognition at a yet more abstract level of analysis and synthesis. The overall structure of this paper is simple, focused on dichotomies, then trichotomies, and then the Common Model. The results suggest new ways of thinking about existing architectures, models and systems, while also highlighting key commonalities, differences, and gaps.

Dichotomies

(A)symmetry

(A)symmetry fundamentally concerns whether the processing of memory structures is valid in only one direction versus omnidirectionally. For example, consider a rule versus a logical implication. Both can be denoted by arrows, but the former only works moving forward whereas the latter works in both directions, and in fact, can even be replaced by a symmetric connective. Or, consider a feedforward neural network versus a Bayesian network. Here the former also only yields valid results moving forward whereas the latter can be used to infer values in any direction. When reverse processing does happen in asymmetric structures – whether for abduction, planning or learning – it is of a fundamentally different form than the forward processing.

In addition to rules and feedforward (including recurrent) neural networks, additional asymmetric forms include both traditional procedural programs plus more recent AI formulations such as arithmetic circuits (Darwiche, 2009) and sum-product networks (Poon & Domingos, 2011). Beyond logics and graphical models – such as Bayesian or Markov networks and factor graphs – additional symmetric forms also include constraints and Boltzmann machines.

With respect to actual dichotomies, rules versus logics (with, for example, model-based semantics) is a traditional symbolic AI one that maps directly onto (a)symmetry. In expert systems, a more abstracted variant occurs as rules versus first-principles reasoning (Davis, 1983), with the latter focusing on flexible use of small amounts of general knowledge, whether logical or not, to yield a wide variety of results that might otherwise require many rules. Abstracting this even further, but still within expert systems, yields shallow (or surface) versus deep reasoning (e.g., Hart, 1982).

Function-based versus model-based approaches – where the former may, for example, comprise feedforward neural networks or arithmetic circuits and the latter graphical models such as Bayesian networks – expresses a related dichotomy that arises in probabilistic AI (Darwiche, 2018). Likewise, within neural networks, we get the dichotomy of heteroassociative versus autoassociative networks (e.g., Rizzuto & Kahana, 2001). Feedforward networks are heteroassociative, generating outputs from inputs but not vice versa, whereas Boltzmann machines are autoassociative. It may seem jarring to view this distinction between types of neural networks in a manner akin to that between rules and logics, but that is a clear conclusion from this analysis.

In (machine) learning more broadly, we see classification versus clustering, supervised versus unsupervised learning, and discriminative versus generative learning (e.g., Ng & Jordan, 2001). The first element in each pair acquires a structure that is to be used in only one direction, whereas the second enables processing in arbitrary directions.

A dichotomy familiar in both symbolic AI and cognitive science is procedural versus declarative memory. In a classical cognitive architecture, such as ACT-R or Soar, procedural memory is based on rules and declarative memory on facts. Rules are asymmetric structures. Facts are static structures that don't themselves mandate a direction of processing. However, they do mandate a means for accessing them. Typically, this involves a mechanism for retrieving the best candidate(s) given any set of cues; a form of symmetric processing, whether as partial match, spreading activation, a holographic memory, or an autoassociative network.

It may even be that it is this symmetric processing rather than the nature of the facts themselves that defines declarative memory and distinguishes it from procedural memory; an idea worth capturing as an explicit hypothesis.

> *(A)symmetric Memory Hypothesis*: Procedural and declarative memory are fundamentally distinguished by differences in processing symmetry rather than content.

Particularly attractive about this hypothesis is how simple yet fundamental the underlying distinction is, and how it thus obviates the need for a messier attempt at distinguishing procedural versus declarative content. It also enables directly mapping varieties of neural networks (e.g., heteroassociative versus autoassociative), symbolic structures (e.g., rules versus logics), and probabilistic structures (e.g., arithmetic circuits versus Bayesian networks) onto procedural versus declarative memory, respectively.

Although a difference in (a)symmetry has long been recognized in how knowledge is retrieved from procedural versus declarative memories, the key difference here is that (a)symmetry is proposed as definitional rather than ancillary, yielding a bottom-up mechanistic definition rather than a top-down content-based one. In the process, the hypothesis has direct implications that would be difficult to derive from distinctions concerning memory content.

Given that procedural and declarative memory fully cover the (a)symmetry dichotomy, and that it appears to be a true dichotomy rather than just the endpoints of a more graduated dimension, the possibility is also raised that there is no further conceptual room for other forms of memory along this dimension. There may, however, be variations of these along other dimensions; for example, image memory may simply be a subsymbolic form of symmetric memory, and thus in a deep sense akin to declarative memory. The two may also be combined; for example, both episodic memory and analogy combine symmetric access to memory structures with subsequent asymmetric processing of the structures, via mapping or succession, respectively. One of these memories may even be used to implement or emulate the other, such as when a rule description is stored in declarative memory, retrieved and interpreted to yield procedural behavior; or when an autoencoder is implemented via a pair of feedforward networks. Still, none of this fundamentally changes the essential nature of the dichotomy.

Two additional dichotomies that are sometimes associated with procedural versus declarative memory are procedural versus declarative semantics (in AI) and implicit versus explicit representations (in cognitive science). The former concerns whether or not structures have fixed, a priori semantics, whereas the latter concerns whether or not there is awareness of the structures during processing. Declarative memory does appear to more naturally support both fixed meanings and awareness, but neither is actually inherent to it, nor does either derive directly from symmetry, so an in depth understanding of these dichotomies is left for future work.

(Non)monotonicity

(Non)monotonicity fundamentally concerns whether processing is additive, cumulative or increasing versus modifiable, retractable or reducible. For example, one of the core pieces of the Common Model is a cognitive cycle that runs at ~50 msec in humans. In Soar and Sigma this cycle is structured as a (mostly) monotonic elaboration phase during which new information is added about the current situation, followed by a nonmonotonic decision (or adaptation) phase during which the situation is actually changed.

This dichotomy also maps to a distinction in cognitive science between automatized versus controlled behavior (Schneider & Shiffrin, 1977), with monotonic processing safely allowed to proceed automatically, while controlled decisions are needed to determine which nonmonotonic change to make. Taking this a step further, it maps onto the dichotomy of parallel versus serial processing, where the absence of interactions or conflicts in monotonic processing authorizes parallelism whereas the need for control and the possibility of interactions among nonmonotonic options implies a need for seriality The mapping for both of these dichotomies is not perfect, as control may be needed to limit parallelism and parallelism may be possible for noninteracting nonmonotonic components; however, the essential commonalities are again what matter here.

Aligning these last two dichotomies yields one form of processing that is autotomized and parallel, plus a second that is controlled and serial. This aggregate dichotomy clearly maps onto both the dichotomies of reactive versus deliberative behavior in cognitive control and fast (System 1) versus slow (System 2) behavior in Kahneman (2011). It has also been characterized in terms of knowledge versus search, or a bit more precisely, as knowledge (K) search versus problem space (PS) search, with the former being monotonic search over what is already known and the latter nonmonotonic problem-space search over the space of combinatoric possibilities (Newell, 1990).

A key takeaway for cognitive science from this is again worth capturing as an explicit hypothesis.

> *(Non)monotonic Control Hypothesis*: Reactive (System 1) and deliberative (System 2) are fundamentally distinguished by differences in processing monotonicity.

Shifting from cognitive science to the cognitive sciences more broadly, and in particular to various subfields of AI, a number of additional variations on this same dichotomy can be found. In constraint solving, there is monotonic propagation (where existing constraints on some variables induce additional constraints on others) versus nonmonotonic conditioning (where hypothetical commitments are made to particular variable values) (Dechter, 2003)). In causal reasoning, the first two steps on the Ladder of Causation (Pearl & Mackenzie, 2018) are association (monotonic probabilistic reasoning) and intervention (nonmonotonic action changes). In logic, the distinction between monotonic and nonmonotonic logics depends on whether inferences made remain valid forevermore versus being retractable. Finally, in search over multimodal spaces, making monotonic moves that never decrease the current value only guarantees a local optimum whereas reaching a global optimum may require interim nonmonotonic moves to lower-valued states.

(A)symmetry × (Non)monotonicity

The cross product of these two elemental dichotomies yields the 2×2 framework outlined in Table 1. Other such cross products have previously been explored in cognitive science, such as one in ACT-R and CLARION (Sun, 2016) that spans (a)symmetry – under two different names – and (sub)symbolic. However, replacing (sub)symbolic with (non)monotonicity in the analysis yields new opportunities for a deeper understanding.

Table 1: 2×2 Framework.

	Asymmetry	*Symmetry*
Monotonicity		
Nonmonotonicity		

Of particular interest here is how this 2×2 framework structures cognitive architectures, models and systems, and how it reveals commonalities and differences among them. With one last explicit hypothesis, it also helps reveal gaps in them.

> *(A)symmetric×(Non)monotonic Necessity Hypothesis*: General intelligence necessitates appropriate processing and learning in all four cells of the (a)symmetry × (non)monotonicity framework.

Initial evidence for this hypothesis will, in what is to come, take the form of how all four cells are required to handle both trichotomies, plus the three architectures that most influenced the initial form of the Common Model of Cognition.

Trichotomies

Tri-level Control Hierarchy

The (non)monotonic dichotomy by itself provides a classic two-level control hierarchy, whether one thinks of it as reactive versus deliberative or System 1 versus System 2. However, a number of approaches go beyond this to three levels. One canonical form spans reactive (immediate response), deliberative (action sequences), and reflective (metacognition), which when mapped to the 2×2 framework bends the normal linear trichotomy into an L shape (Table 2).

Table 2: 2×2 Mapping of Tri-Level Control Hierarchy.

	Asymmetry	*Symmetry*
Monotonicity	Reactive	
Nonmonotonicity	Deliberative	Reflective

The vertical leg retains the general mapping from earlier of reactive onto monotonic and deliberative onto nonmonotonic but restricts them to the corresponding asymmetric cells. The reactive level in control hierarchies unsurprisingly focuses on procedural rather than declarative memory, due to the former's focus on control, and thus maps to the top-left cell. Declarative memory can clearly play a role in control, but this is typically ignored in control trichotomies.

At the elbow of the L is deliberative processing, consisting of a controlled action sequence that yields a single asymmetric path through situations in the world. Following the horizontal leg to the right yields reflective use of action models to explore simulated paths between arbitrary states – that is, models of situations – thus yielding the ability to search omnidirectionally in a metacognitive problem space.

Tables 3-4 show how this all works for two tri-level control hierarchies from very different contexts: a classical robot control approach (Bonasso et al., 1997); and the AlphaZero approach to board games. Although these examples are, respectively, from robotics and (neural) ML/AI, and each implements the cells in the hierarchy differently, they both fit this same trichotomic framework, as do also the three cognitive architectures that are analyzed later.

Table 3: 2×2 Mapping of the 3T Architecture.

3T Architecture	*Asymmetry*	*Symmetry*
Monotonicity	Skill Manager	
Nonmonotonicity	Sequencer	Planner

Table 4: 2×2 Mapping of AlphaZero.

AlphaZero	*Asymmetry*	*Symmetry*
Monotonicity	Neural Networks	
Nonmonotonicity	Game Moves	Monte Carlo Tree Search

Tri-level Content Hierarchy

Tri-level content hierarchies are less common than tri-level control hierarchies, but they do exist, and bear an interesting relationship to the other. One version of this can be seen in Table 5, for affective content (Ortony, Norman & Revelle, 2005). The development of this hierarchy began with a tri-level control hierarchy, but then the distinct nature of the emotional content at each level was identified. As in control, both nonmonotonic cells are filled, but with emotional content. The larger difference, however, is that the monotonic level is now symmetric rather than asymmetric, corresponding to declarative rather than procedural memory.

Table 5: 2×2 Mapping of the Affect Hierarchy.

Affect	*Asymmetry*	*Symmetry*
Monotonicity		Proto-Affect
Nonmonotonicity	Primitive Emotions	Cognitively Elaborated Emotions

Another tri-level content hierarchy, but from AI, is the Ladder of Causation mentioned earlier (Table 6). The tri-level content hierarchy here includes Bayesian reasoning (association level), reasoning about actions (intervention level), and hypothetical, or metacognitive, reasoning (counterfactual level). One major point of Pearl's work is that causal reasoning isn't all just (monotonic) Bayesian.

Table 6: 2×2 Mapping of the Ladder of Causation.

Causality	*Asymmetry*	*Symmetry*
Monotonicity		Association
Nonmonotonicity	Intervention	Counterfactuals

The asymmetric monotonic cell, where procedural memory resides, is unsurprisingly blank in both of these content hierarchies. As with the corresponding gap in control hierarchies, the missing memory could be used, but it is at best of secondary importance, and thus not typically a focus.

Common Model of Cognition

The Common Model of Cognition is being developed as an evolving community consensus concerning the structures and processes that yield human-like minds, in service of creating a cumulative reference point for the field while guiding efforts to both extend and break it. The question of interest here is to what extent the 2×2 framework can help to better understand the Common Model. The first step involves a mapping of its memory and control aspects (Table 7), followed by corresponding mappings of ACT-R, Soar and Sigma (Tables 8-10). Learning is then mapped, with AlphaZero added to the mix for this analysis.

Like the earlier trichotomies, the Common Model is incomplete, spanning only three of the framework's cells. However, in contrast to the two trichotomies, the Common Model spans both monotonic cells while omitting a metacognitive, or reflective, capability in the symmetric nonmonotonic cell. This lack. However, reflects that a consensus is needed rather than that there is a consensus against such a capability (Kralik, et al., 2018).

Table 7: 2×2 Mapping of the Common Model.

Common Model	*Asymmetry*	*Symmetry*
Monotonicity	Procedural	Declarative
Nonmonotonicity	Action Selection & Execution	

Three Cognitive Architectures

The Common Model, as a partial consensus over cognitive architectures, lacks aspects such as metacognition that may exist in the architectures from which it is derived. So, as a follow up step, it is useful to extend this analysis to the three architectures that heavily influenced its initial development – ACT-R, Soar and Sigma (Tables 8-10) – each of which includes some form of metacognition.

Table 8: 2×2 Mapping of ACT-R.

ACT-R	*Asymmetry*	*Symmetry*
Monotonicity	Rule Match	Facts
Nonmonotonicity	Selection & Execution	Imaginal Buffer

Table 9: 2×2 Mapping of Soar.

Soar	*Asymmetry*	*Symmetry*
Monotonicity	Parallel Rule System	Facts & Episodes
Nonmonotonicity	Selection & Execution	Reflection

Table 10: 2×2 Mapping of Sigma.

Sigma	*Asymmetry*	*Symmetry*
Monotonicity	Asymmetric Graphs	Graphical Models
Nonmonotonicity	Selection & Execution	Reflection

In conjunction with Table 7, these mappings show how the three architectures fulfill the Common Model's requirements for its three cells, and fill in its blank cell, while highlighting the diverse ways they implement such capabilities.

All three architectures support rules in procedural memory, but in ACT-R it is only their match process that is monotonic, with a rule then being selected to yield a nonmonotonic action execution. Soar matches and fires its rules in parallel, making the whole rule system – but not final action (or operator) selection – part of procedural memory. Sigma uses a unidirectional extension of its graphical models that subsumes not only parallel rules but also feedforward neural networks (Rosenbloom, Demski & Ustun, 2017) and sum-product networks (Joshi, Rosenbloom & Ustun, 2018); with action/operator selection also separated out.

One implication of this analysis of procedural memory is that the 2×2 framework may draw boundaries that are somewhat askew from those found in standard analyses. The approach here splits off nonmonotonic aspects that would traditionally be considered part of procedural memory and includes them instead as part of action selection and execution. Whether this is ultimately the correct view remains to be seen, but either way, such boundary shifts are an important part of what falls out of these analyses.

In declarative memory, all three architectures can represent facts, although Soar decomposes this general memory capability into distinct semantic and episodic memories, and Sigma's factor graphs provide a broader range of possibilities that includes not only these two but also other forms of hard and soft constraints. All three architectures also support selecting the best partial match from declarative memory, but this does involve asymmetric and nonmonotonic processing.

Soar and Sigma are similar in the nonmonotonic layer, due to Sigma's approach being based on Soar's, with the asymmetric cell being action related and impasse-driven reflection providing the ability to leverage models for search within the symmetric cell. However, Sigma's selection process for declarative memory shares much with its procedural selection, whereas in Soar they are distinct, including an asymmetrical form of spreading activation. In ACT-R, rule selection and action execution provide its asymmetric component, while its symmetric component is based on an imaginal buffer that can represent hypotheticals.

Learning

Table 11 shows an abstract mapping of forms of learning that blends terms from the tri-level control hierarchy and the Common Model. Combining this with an extension, to all four cells, of the Common Model's notion that structure and parameter learning are needed in both procedural and declarative memory, we can jointly analyze learning in the Common Model, ACT-R, Soar, Sigma and AlphaZero to better understand its overall structure, how the approaches compare and contrast, and what gaps may show up in them.

Table 11: 2×2 Mapping of Learning.

Learning	*Asymmetry*	*Symmetry*
Monotonicity	Procedural	Declarative
Nonmonotonicity	Deliberative	Reflective

Procedural – i.e., asymmetric monotonic – learning includes rule creation via composition/chunking (Common Model, ACT-R and Soar) and parameter learning via backpropagation (Sigma and AlphaZero). None of these models/systems are thus complete with respect to procedural learning. The Common Model is described in a way that appears to be complete, but that is due to considering reinforcement learning (RL) – which learns to select actions from experience with action sequences – as procedural. But, by the analysis here, RL is an asymmetric nonmonotonic form of learning, and thus belongs instead in that cell.

On the positive side, by including RL, all five models/systems do thus span asymmetric nonmonotonic parameter learning. None of them, however, learns new primitive actions, although Soar at least learns new high-level actions by combining primitive ones (Mohan & Laird, 2014).

For declarative – i.e., symmetric monotonic – learning, the Common Model acquires facts and the quantitative metadata that facilitates their use. Both ACT-R and Soar directly implement such a combination. In Sigma, facts are instances of predicates with typed arguments. The only actual structure learning at present is type extension, whereas quantitative metadata is learned via Hebbian-style symmetric learning. Adding facts to declarative memory occurs not by structure learning but by raising probabilities above 0. AlphaZero has no declarative memory, and thus no role for its learning.

Symmetric nonmonotonic, or reflective, learning can be thought of as the acquisition of models and their parameters. The Common Model does not include these forms of learning due to its general lack of metacognition, even though all three of the architectures mentioned do embody some form of it. AlphaZero uses action models in model-based RL, but it does not appear to learn these models.

Conclusion

The first step in this paper was to reduce two distinct clouds of dichotomies down to simple computational forms. In the process it was hypothesized that the distinction between procedural and declarative memory – along with many other dichotomies (Table 12, left column) – can be grounded in the more elemental terms of (a)symmetry. The possibility was even raised that although there may be other variants or combinations of these two basic types of memory, there may be no further basic types along this dimension. It was then also hypothesized that the distinction between reactive and deliberative behavior can be grounded in the more elemental terms of (non)monotonicity, also along with many other dichotomies (Table 12, right column).

The cross product of these dichotomies yields a 2×2 framework that enables analyzing two key trichotomies and the Common Model of Cognition, providing a common means for understanding and comparing across divergent integrations of cognitive capabilities. It also identifies gaps, when accompanied by a hypothesis relating to the processing and learning that is necessary in all four cells. It further helps understand how apparently ad hoc but highly successful systems such as AlphaZero can fit within the same coherent framework for memory, control and learning as more traditional cognitive architectures, models and systems.

Table 12: Summary of Dichotomy Mappings.

(A)symmetry	*(Non)monotonicity*
Rules vs. Logic	Elaboration vs. Decision
Rules vs. First Principles	Automatized vs. Controlled
Shallow vs. Deep	Parallel vs. Serial
Function vs. Model	Fast vs. Slow
Hetero. vs. Autoassociative	System 1 vs. System 2
Classification vs. Clustering	K vs. PS Search
Supervised vs. Unsupervised	Propagation vs. Conditioning
Discriminative vs. Generative	Association vs. Intervention
Proc. vs. Decl. Memory	Mon. vs. Nonmon. Logic
	Local vs. global

In future work, this analysis needs to be extended to more systems and architectures, to more precise mappings onto the framework, and to a deeper level of understanding of the full dichotomic clouds. A complete analysis of cognition should also ultimately provide a coherent story over all relevant dichotomies and their combinations. Additional dichotomies of relevance may include discrete versus continuous, central versus peripheral, explicit versus implicit, symbolic versus subsymbolic, conscious versus subconscious, and short-term versus long-term. Additional combinations of dichotomies will also be of central importance; possibly even eventually up to a full combination of all relevant, distinct dichotomies.

Acknowledgments

The work described in this article was sponsored by the U.S. Army. Statements and opinions expressed may not reflect the position or policy of the United States Government, and no official endorsement should be inferred.

References

Anderson, J. R. 2007. *How Can the Human Mind Exist in the Physical Universe?* New York: Oxford Univ. Press.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. & Slack, M. (1997). Experiences with an architecture for intelligent reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence*, *9*, 237-256.

Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks.* New York: Cambridge University Press.

Darwiche, A. (2018). Human-level intelligence or animal-like abilities? *Communications of the ACM*, *61*, 56-67.

Davis, R. (1983). Reasoning from first principles in electronic troubleshooting. *International Journal of Man-Machine Studies*, *19*, 403-423.

Dechter, R. (2003). *Constraint Processing*. San Francisco, CA: Morgan Kaufmann.

Hart, P. (1982). Directions for AI in the eighties. *ACM SIGART Bulletin*, *79*, 11-16.

Joshi, H., Rosenbloom, P. S. & Ustun, V. (2018). Exact, tractable inference in the Sigma cognitive architecture via sum-product networks. *Advances in Cognitive Systems*, *7*, 31-47.

Kahneman, D. (2011). *Thinking Fast and Slow*. New York: Farrar, Straus and Giroux.

Kralik, J. D., Lee, J. H., Rosenbloom, P. S., Jackson, P. C., Epstein, S. L., Romero, O. J., Sanz, R., Larue, O., Schmidtke, H. R., Lee, S. W. & McGreggor, K. (2018). Metacognition for a Common Model of Cognition. *Procedia Computer Science*, *145*, 740-746.

Laird, J. E. 2012. *The Soar Cognitive Architecture*. Cambridge, MA: MIT Press.

Laird, J. E., Lebiere, C. & Rosenbloom, P. S. (2017). A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, *38*, 13-26.

Mohan, S. & Laird, J. E. (2014). Learning goal-oriented hierarchical tasks from situated interactive instruction. *Proc. 28th AAAI Conf.* (pp. 113–130).

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Ng, A. Y. & Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Proc. 14th International Conf. on Neural Information Processing Systems* (pp. 841-848).

Ortony, A., Norman, D.A. & Revelle, W. (2005). Affect and proto-affect in effective functioning. In J.M. Fellous & M.A. Arbib (Eds.), *Who Needs Emotions? The Brain Meets the Machine*. New York: Oxford Univ. Press.

Pearl, J. & Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. New York: Basic Books.

Poon, H., & Domingos, P. (2011). Sum-product networks: A new deep architecture. *Proc. IEEE International Conf. on Computer Vision Workshops* (pp. 689-690).

Rizzuto, D. S. & Kahana, M. J. (2001). An Autoassociative Neural Network Model of Paired-Associate Learning. *Neural Computation*, *13*, 2075–2092.

Rosenbloom, P. S., Demski, A. & Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*, 1-103.

Rosenbloom, P. S., Demski, A. & Ustun, V. (2017). Toward a neural-symbolic Sigma: Introducing neural network learning. *Proc. 15th Annual Meeting of the International Conference on Cognitive Modeling*.

Schneider, W. & R. M. Shiffrin. (1977). Controlled and automatic human information processing: 1. Detection, search, and attention. *Psychological Review*, *84*, 1-66.

Silver, D., Hubert, J., Anonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *363*, 1140-1144.

Sun, R. (2016). *Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture*. New York: Oxford Univ. Press.

Towards a Cognitive Model of the Takeover in Highly Automated Driving for the Improvement of Human Machine Interaction.

Marlene Scharfe (m.scharfe@campus.tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23, 10587 Berlin, Germany

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23, 10587 Berlin, Germany

Abstract

In this study, an ACT-R cognitive model, that depicts cognitive dynamics during a takeover in highly automated driving is developed. Such a model is inevitable to explain strong differences between studies and display and represent cognitive dynamics. Thus, the goal of the model is, to validly display the time sequence of the steps, that are undertaken to build up situation awareness during the takeover task. It is developed based on video analysis and pertinent literature. Empirical data of a real traffic study show takeover patterns that are applied after engaging into a non-driving related task. Correlations between model predictions and empirical data evaluation show, that the model is able to display cognitive dynamics. It serves as solid basis, but still further development is aspired, concentrating on the impact of traffic complexity.

Keywords: highly automated driving; HAD; cognitive modeling; ACT-R; takeover; TOR; conditional automation; NDRT; non-driving related tasks; real vehicle study; resource model; situation awareness; SA;

Introduction

The development of technological innovations in the field of highly automated driving is growing rapidly. The next level of automation (SAE Level 3; SAE, 2014) enables the driver to engage into non-driving related tasks (NDRT) during the automated drive. Still, the driver needs to respond properly to a takeover request (TOR; SAE, 2014). Thus, the driver has to be enabled to take over the driving task in a safe and comfortable manner. Several studies have investigated takeover times (Feldhütter, Gold, Schneider, & Bengler, 2017; Gold, Damböck, Lorenz, & Bengler, 2013; Naujoks & Neukum, 2014; Walch, Lange, Baumann, & Weber, 2015). As various factors influence the takeover, current results concerning takeover times and behavior in Level 3 illustrate incomparable results that range from 1,14s (Zeeb, Buchner, & Schrauf, 2015) to 15s (Merat, Jamson, Lai, Daly, & Carsten, 2014). It is of scientific relevance to understand underlying cognitive dynamics that lead to these differences. Also, to improve the human-machine interaction (HMI) and the product development in the indusrty, it is inevitable to unravel the black box of cognitive dynamics and gain an understanding of how the human processes the takeover and builds up situation awareness (Endsley, 1995). According to Endsley (1995), situation awareness encompasses three main stages: *perception*, *comprehension* and *projection* (included in Figure 1). The basis is formed by a persons visual perception of the environment. Based on the perception, the meaning of the current situation has to be understood (comprehension) and a future status is projected (Endsley, 1995). The aim of this study is to develop a cognitive model, representing cognitive dynamics during the takeover task. Yet, cognitive models for the driving task exist (e.g. Salvucci, 2006), but no renowned cognitive model explicitly displays the takeover and includes surrounding traffic.

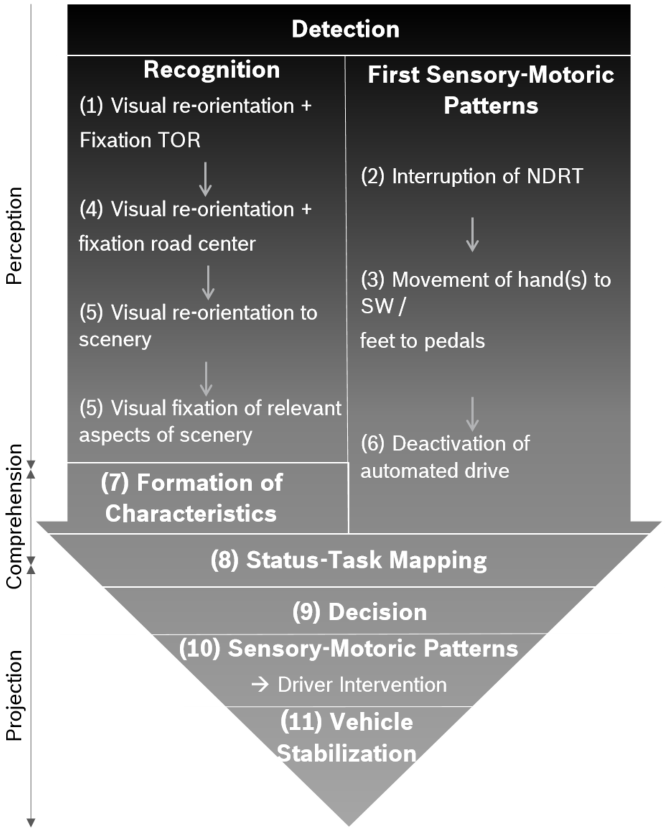

Perception
Comprehension
Projection
Detection
Recognition
(1) Visual re-orientation + Fixation TOR
(4) Visual re-orientation + fixation road center
(5) Visual re-orientation to scenery
(5) Visual fixation of relevant aspects of scenery
(7) Formation of Characteristics
First Sensory-Motoric Patterns
(2) Interruption of NDRT
(3) Movement of hand(s) to SW / feet to pedals
(6) Deactivation of automated drive
(8) Status-Task Mapping
(9) Decision
(10) Sensory-Motoric Patterns
→ Driver Intervention
(11) Vehicle Stabilization

Figure 1: Definition of the Takeover Process and the Phases of Situation Awareness for the cognitive model approach (Source: own figure).

In this paper a first cognitive modeling approach of the takeover task is introduced and a comparison of model results to empirical data is drawn. The goal is to create an understanding of cognitive mechanisms during the separate steps of the takeover process (visualized in Figure 1). Cognitive processes are later described in detail. The focus of the model lies on the perception mechanisms rather than the interruption process which has been modeled by Borst, Taatgen, and van Rijn (2015). To better understand perception mechanisms, the interruption is implemented here simply as a new goal setting, which can later be expanded. Here, the steps to update situation awareness and perform the takeover task are modeled. As soon, as the the basic cognitive dynamics are understood, aspects of complexity will be included in future work.

For the current approach, it is investigated whether a cognitive model is able to a.) explain cognitive mechanisms during the takeover according to empirical data, b.) depict mechanisms that arise due to NDRTs and c.) illustrate cognitive processes to update SA using the visual component.

Methods

The approach of this study is to gain an understanding of cognitive dynamics during the takeover task. Thereby, behavior during a takeover can be predicted and empirical results of past studies can be explained. Based on pertinent concepts and results of relevant studies, the takeover process in conjunction with the three stages of situation awareness (SA; Endsley, 1995) is defined (Figure 1) and a cognitive model established for the takeover task. In order to validate overall predictions of the model, these are compared to empirical data of a driving study in a real traffic environment. After validating the overall model performance, different conditions of the NDRT are closer examined to extract potential improvements. Following, the data acquisition is described and methods as well as the functions of the model outlined.

Data Acquisition

The utilized data derives from a real traffic study of another project (Ko-HAF, 2017) in 2017 with a Wizard of Oz vehicle in the area of Stuttgart, Germany. The vehicle allows the experience of highly automated driving in a real driving environment. The co-driver is able to drive the car covertly via a control at the passengers seat and thus simulate highly automated driving (Level 3). This results in the drivers ability to turn away from the driving task (Ko-HAF, 2017) and engage into a non-driving related task (NDRT). For the current approach, the type of NDRT that is performed, is not relevant. As soon as a TOR is triggered, participants have to take over. For this study's purpose $N = 28$ participants are evaluated. Half of the subjects ($N = 14$) are used for qualitative video analysis concerning gaze- and motoric behavior. This data is used for the model development. Two raters independently view the videos and assess the participants' behavior in a standardized way. Four behavioral relevant steps are extracted (1. Gaze TOR, 2. Interruption NDRT, 3. Hands steering wheel (SW) and 4. Gaze Road), that define the time course of the takeover for the current purpose. In order to validate resulting model predictions, the other half ($N = 14$) is quantitatively evaluated concerning the four steps and compared to model results.

Cognitive Model

The cognitive model displays sub-steps that are undertaken during the takeover and cognitive dynamics for visual perception, motoric reactions and decision making. The aim is, to have model predictions in accordance with empirically found reaction times. Motoric, visual and cognitive steps to update SA are included into the model (Figure 1). To realize the implementation, the cognitive architecture ACT-R (Anderson et al., 2004) is used. It allows the modeling of a wide range of higher cognitive processes (Taatgen, 1999) and provides an accurate representation of human abilities (Salvucci, Boer, & Liu, 2001). ACT-R contains various modules (intentional, declarative, visual and manual) that communicate with each other through buffers (Anderson et al., 2004). The opportunity to model processes of the particular modules separately and gain insights of their interaction in certain situations is essential for modeling the takeover. Cognitive modules form the mental representation elaboration and decision-making processes on attentional and automatic levels (Bellet et al., 2012). Visual and aural modules describe the visual and aural processing of stimuli (Anderson et al., 2004). The manual module is responsible for the execution of motoric responses.

For this study's' purpose, three steps are undertaken. Based on literature (Endsley, 1995; Salvucci, 2006) and on qualitative analysis of video material from the driving study ($N = 14$), first the process that occurs during a takeover is defined for the current model (Figure 1). These steps are then used for the definition of the cognitive model. Third, the model is tested and compared to empirical data. It is important to clearly define the takeover task and steps that are undertaken to regain control. (Figure 1) outlines how this is addressed. Additionally, it has to be understood how the different stages of SA (Endsley, 1995) are realized to implement corresponding cognitive structures in ACT-R. The focus of the current model lies on the visual *perception phase* of SA as it is the most important of the three SA stages in driving, the first to occur (Ratwani, McCurry, & Trafton, 2010) and forms the basis for the subsequent steps. Following, the takeover process and the corresponding realization in the model are described.

The scenario, that the model illustrates, describes a highly automated driving situation (Level 3) on a three lane highway. The model starts with a NDRT during the automated drive and ends with an action decision after the takeover. Figure 2 illustrates the model productions.

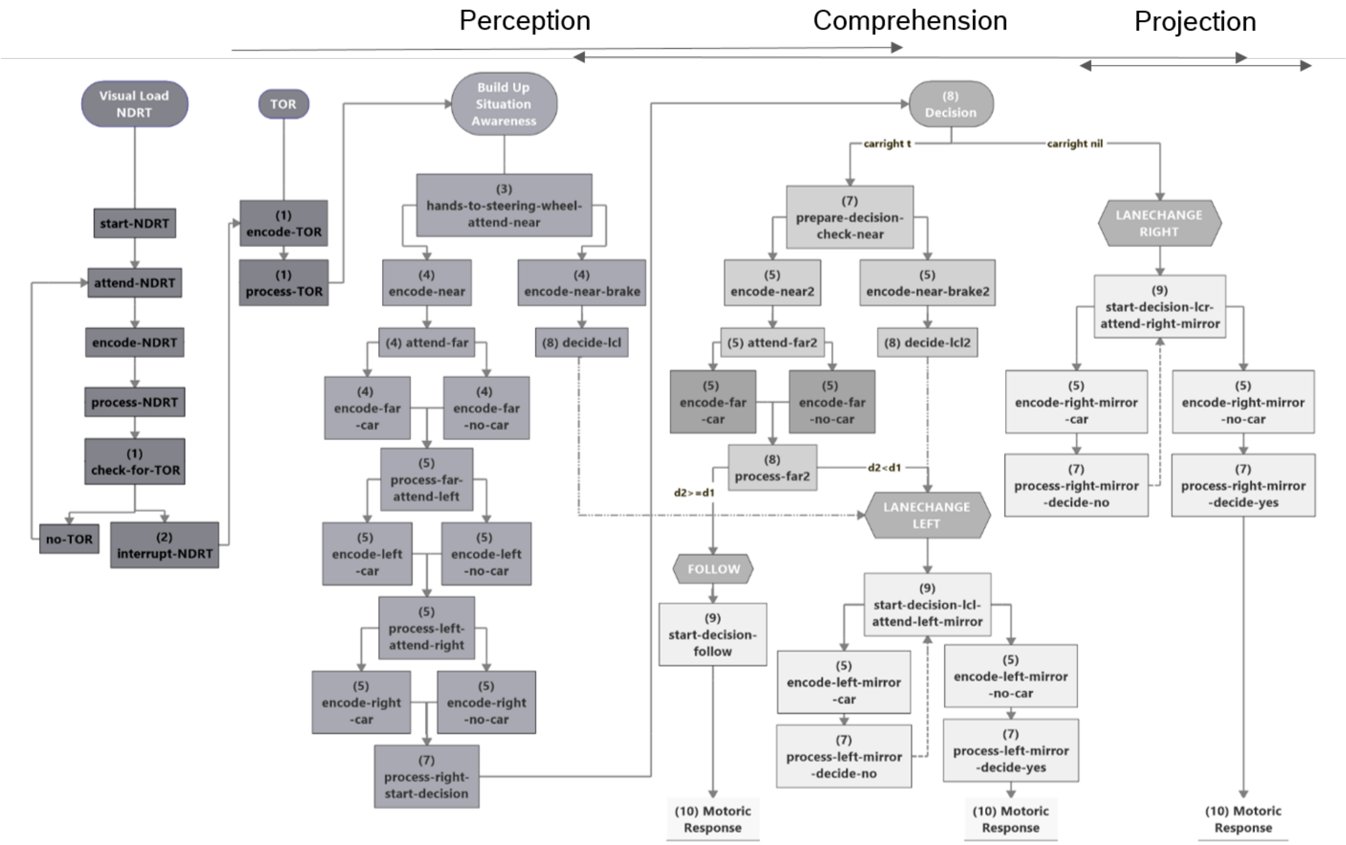


Figure 2: Cognitive Model of the Transition Process after an Automated Drive. Representation of productions from the NDRT to the motoric response (Source: own figure).

Figure 3 represents the environment, the model interacts with and the areas that are attended by the model. The visual perception undergoes three main steps in the *perception phase* (Endsley, 1995). While focusing on the NDRT, the model is constantly checking for a takeover request (TOR) in the visual-location. This is due to the given task of taking over the driving as soon as a TOR appears. Due to the perception of a visual or aural stimulus, the gaze is moved to the TOR (1.; Visual reorientation and fixation of the TOR) and the goal set to attend the TOR. In case no TOR can be detected, the NDRT is started again. Otherwise as soon as the TOR is triggered, the NDRT (Visual Load NDRT) has to be interrupted and a retrieval request for the meaning of the TOR is made. At the same time, the visual interruption of the NDRT (2.) is executed. It automatically occurs with the attentional shift to the TOR in the *perception phase* (Endsley, 1995; Salvucci, 2006). They come along with first automated sensory-motoric steps, encompassing the movement of hands to the steering wheel and feet to the pedals (3.). In the model only the movement of the hands to the steering wheel can be implemented, as ACT-R does not include the feet yet. Nevertheless, a movement action is carried out by the manual module. These are based on an automatic reaction rather than focused decision making and fluently merge into the *comprehension phase* (Endsley, 1995).

In separate productions, the TOR is attended, encoded and processed. After the TOR has been processed, SA has to be build up (Build Up Situation Awareness). Hence, the gaze is oriented to the road center and the front lane (near and far area) is checked for objects (4.; Salvucci, 2006). In case an object is detected in the near area (encode-near-brake), a strong brake is carried out (motoric module) and a direct decision to change to the left lane is made, setting the goal to lanechange left. This case pictures a critical event and should not apply when a non-critical TOR is triggered. Thus, in the current context this case is postponed, as the study focuses on non-critical takeover scenarios. Nevertheless, for completeness of the model it is still necessary to cover the scenario. In the majority of cases though, no object is in the near area and the far front lane is attended. It is encoded whether there is a car or not (attend-far) and the result processed. After this, the rest of the scenery is attended (5.), starting with the left lane (process-far-

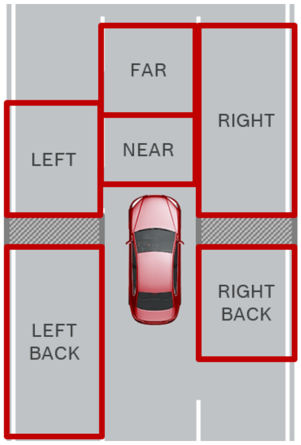


Figure 3: Focus Areas of the Cognitive Model for Action Decision (Source: own figure).

attend-left). Again, the model encodes whether there is a car or not. The result is processed and stored in a chunk in the imaginal. This process is repeated to perceive the status of the right lane likewise (process-left-attend-right). Each status (object or not) of the attended areas is stored in the imaginal to form a representation of the traffic environment. Here, characteristics of the situation are formed and the current status is recognized (7.; Endsley, 1995) while the automation is deactivated (6.). The deactivation of the automation is not represented in the model, as there is no common function yet and operates by oversteering. After a mapping between the perceived status and the task (8.), an action decision is made (9.; *comprehension phase* of SA; Endsley, 1995). The decision productions (Decision) are based on the status of the surrounding traffic in the focus areas (Figure 3) and the underlying law. In case no car has been detected on the right lane, a lane change to the right is triggered (LANECHANGE RIGHT). Thus, the right mirror is attended by the visual module to check, whether the right back is free. If it is free, the lane change is executed by a motoric response of the manual module (10.). In case, there is a car in the right back, the mirror is attended repeatedly, until the lane change can be performed safely. If the right lane is not free, the front road is attended again, applying the same mechanisms as before. Now, distances from the first allocation that are stored in the imaginal are compared to the current perception (process-far2). If the distance to the car in the front stays the same or is increasing, car following is decided and performed (FOLLOW). If the distance to the front car is decreasing, a lane change to the left is triggered (LANECHANGE LEFT). Again, the mirror is repeatedly checked for occupation of the left back. Equally to the right lane change, a left lane change is only performed by the manual module, when it can be carried out safely (no object in the visual-location for the left back; Figure 2). The model runs until this stage, predicting type and time of action execution in different traffic conditions. Still, the focus of this paper is to validate the model steps in order to produce valid predictions. After the driver intervention (10.), the vehicle is stabilized (11.; *projection phase* of SA; Endsley, 1995).

Results

For the statistical evaluation, only data that the model does not base on, is used. Four timestamps were defined for the individual takeover behavior (1. Gaze TOR, 2. Interruption NDRT, 3. Hands SW and 4. Gaze Road). In a first step, to validate the model in general, the timestamps were plotted for each individual ($N = 14$) without distinguishing between the different NDRTs (Figure 4). As the figure shows, the variance of the measurement values is high and outliers can be detected. For statistical evaluation, median values rather than the arithmetic mean were used, as the median is more robust towards outliers (Schillinger, 2002). Data was tested on non-linearity, normal distribution, homoscedasticity and influential outliers using residual vs. fitted-, normal-Q-Q-, scale-location and residuals vs. leverage plots.

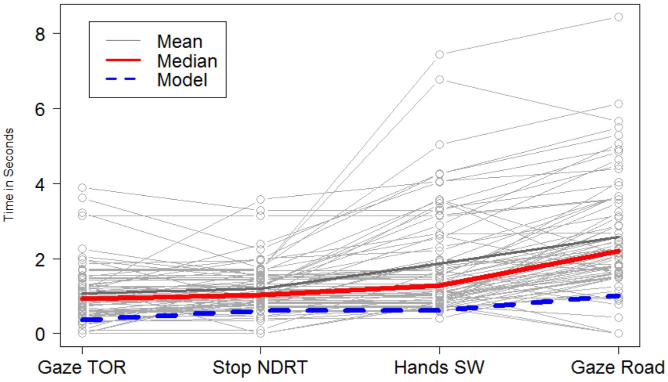


Figure 4: Course of the Participant Data and Model Prediction of Takeover Behavior Patterns ($N = 14$; Source: own figure).

Overall, the gaze to the TOR was performed with a median of 0.96s ($M = 1.07$s), ranging from a minimum (min) of zero seconds (s; zero seconds occur due to no NDRT condition, in which the gaze did not rest on a NDRT) to a maximum (max) of 3.3s. The NDRT was at the median of 1.08s interrupted ($M = 1.25$s, $min =$ 0s, $max = 3.58$s), the hands were after a median of 1.32s ($M = 1.91$s, $min = 0.66$s, $max = 7.44$s) at the steering wheel and after 2.28s ($M = 2.7$s, $min = 0$s, $max =$ 8.44s) the gaze was on the road. The model performs the sequence with the following times: (1) Gaze to TOR

Table 1: Correlations of NDRT with the model predictions and the correlation of accumulated empirical data with model predictions (significance codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1; source: own table).

NDRT	Correlation Coefficient and p-Value Arithmetic Mean	Correlation Coefficient and p-Value Median
Baseline	$r(2) = .96$, $p = .038$ *	$r(2) = .99$, $p = .006$ **
Listening	$r(2) = .96$, $p = .035$ *	$r(2) = .99$, $p = .012$ *
Searching	$r(2) = .86$, $p = .136$	$r(2) = .9$, $p = .1$
Reading	$r(2) = .8$, $p = .2$	$r(2) = .83$, $p = .17$
Tetris	$r(2) = .89$, $p = .1$	$r(2) = .95$, $p = .048$ *
Accumulated	$r(2) = .91$, $p = .086$ *	$r(2) = .95$, $p = .048$ *

0.37s, (2) Interruption of NDRT 0.62s, (3) Hands at SW 0.62s and (4) Gaze on Road 1.02s. It is noticeable, that the model is overall faster in the reaction times. As this applies for the overall time course, this difference is supposed to be justified by some patterns, that apply before the gaze is directed to the TOR. This is an important fact, that is further addressed in the discussion. It does not affect the overall evaluation though, as a time shift beforehand would shift the overall sequence. Statistical analysis of the data shows a significant bivariate pearsons correlation between the median behavior of the participants ($n = 14$) and the model predictions ($r(2) = .95$, $p = .05$). Figure 4 shows the course of the individuals over all takeover conditions, including the median course (bolt-red line), the mean course (dark-gray line) and the course of the model predictions (striped-blue line). The correlation shows that predictions of the model represent empirical data validly which strengthens proceeding model predictions of the action decision. In order to understand where the variances come from, further analysis concerning the different NDRTs is made. Table 1 shows that only NDRT conditions without additional physical movements show significant correlations with the cognitive model using mean as well as median values. These movements apply mainly for reading the newspaper (occupied hands) and searching something in the back (torso turned, hands occupied). The tetris on the mounted tablet also involves the hands, but no holding is necessary, hence the correlation is significant for median, but not for mean values. Overall, no significant correlation of model predictions can be found for conditions, in which additional motoric processes were necessary before taking over. This indicates, that the model can already display the cognitive occupation of NDRTs, but lacks to include motoric complex tasks. In further studies, additional cognitive processes will be investigated more detailed with eyetracking analysis.

Discussion

Results indicate, that the cognitive model is able to validly represent the time course during a takeover (a.). Only conditions, in which the hands are occupied by holding something (reading a newspaper) or turning the torso (searching something in the back), model predictions do not correlate significantly with empirical time courses. This is not a surprise however, as the ACT-R motor module does not yet include complex movements. This finding illustrates that as soon as the hands are occupied during a NDRT or the body is turned away, additional movement patterns are necessary (e.g. removing reading glasses, folding newspaper, turning body to front) before the defined takeover steps can be performed. Mechanisms that arise due to NDRTs can only partly be depicted (b.). These resource depending circumstances should be considered in the model for better time estimation of NDRT, depending on the occupied resource and NDRT cancellation criteria. As mentioned before, although significant correlations are found for the course of the takeover, the model is in its overall performance still faster. The cause of that may lie in the fact, that data was collected in real traffic environments. Thus, more environmental cues than the model includes are probably attended by participants before moving the gaze to the TOR (c.).

However, the current model provides a good basis for predictions of the takeover. It still is a general model and does not depict individual differences. The model will be refined next, incorporating the impact of the complexity of a situation. Eye movement patterns will be captured for more particularized understanding of processes in the vision module. Along with this, further development of the SA update as well as of the action decision will be validated. The current cognitive model is going to be expanded to enable the prediction of individualized cognitive processes depending on the situation complexity. These predictions will serve as groundwork for further specification of cognitive processes in complex traffic situations and resulting action decisions, that should be linked to eye-movement patterns. Resulting predictions of human cognition consequently serve for the adaption of HMI concepts.

Conclusion

The present model allows the prediction of takeover patterns in highly automated driving. It is able to illustrate several steps that are undertaken during the takeover process validly. This approach provides a solid groundwork for a more specific development of the model. The impact of complexity aspects on cognitive processes during the takeover will further be included. In the next step, research on complex situations will be done. To understand how the cognitive system and especially the visual perception handles complex situations, eye tracking data will be gathered. By implementing and comparing this approach to empirical data the underlying cognitive processes can be elucidated and resulting necessities during a takeover understood. This will later serve as guideline for the development of suitable HMI systems.

Acknowledgements

I wish to acknowledge the help provided by my supervisors Michael Schulz and Kathrin Zeeb from Robert Bosch GmbH and the public promoted projects PAKoS and Ko-HAF.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Bellet, T., Bornard, J.-C., Mayenobe, P., Paris, J.-C., Gruyer, D., & Claverie, B. (2012). Computational simulation of visual distraction effects on car drivers'situation awareness. In *Iccm 2012*.

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2015). What makes interruptions disruptive?: A process-model account of the effects of the problem state bottleneck on task interruption and resumption. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 2971–2980). ACM.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human factors*, *37*(1), 32–64.

Feldhütter, A., Gold, C., Schneider, S., & Bengler, K. (2017). How the duration of automated driving influences take-over performance and gaze behavior. In *Advances in ergonomic design of systems, products and processes* (pp. 309–318). Springer.

Gold, C., Damböck, D., Lorenz, L., & Bengler, K. (2013). "take over!" how long does it take to get the driver back into the loop? In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 57, *1*, pp. 1938–1942). SAGE Publications Sage CA: Los Angeles, CA.

Ko-HAF. (2017). Ko-haf - wizard-of-oz-konzept. YouTube. Retrieved from https://www.youtube.com/watch?v=4mm3xaBfQZc

Merat, N., Jamson, A. H., Lai, F. C., Daly, M., & Carsten, O. M. (2014). Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation research part F: traffic psychology and behaviour*, *27*, 274–282.

Naujoks, F., & Neukum, A. (2014). Timing of in-vehicle advisory warnings based on cooperative perception. In *Proceedings of the human factors and ergonomics society europe chapter annual meeting* (pp. 193–206). HFES Torino.

Ratwani, R. M., McCurry, J. M., & Trafton, J. G. (2010). Single operator, multiple robots: An eye movement based theoretic model of operator situation awareness. In *Proceedings of the 5th acm/ieee international conference on human-robot interaction* (pp. 235–242). IEEE Press.

SAE, T. (2014). *Surface vehicle information report. taxonomy and definitions for terms related to on-road motor vehicle automated driving systems.* SAE International.

Salvucci, D. (2006). Modeling driver behavior in a cognitive architecture. *Human factors*, *48*(2), 362–380.

Salvucci, D., Boer, E., & Liu, A. (2001). Toward an integrated model of driver behavior in cognitive architecture. *Transportation Research Record: Journal of the Transportation Research Board*, (1779), 9–16.

Schillinger, M. P. (2002). *Flächenhafte schätzung mit classification and regression trees und robuste gütebestimmung ökologischer parameter in einem kleinen einzugsgebiet* (Doctoral dissertation).

Taatgen, N. (1999). The atomic components of thought. *Trends in Cognitive Sciences*, *3*(2), 82.

Walch, M., Lange, K., Baumann, M., & Weber, M. (2015). Autonomous driving: Investigating the feasibility of car-driver handover assistance. In *Proceedings of the 7th international conference on automotive user interfaces and interactive vehicular applications* (pp. 11–18). ACM.

Zeeb, K., Buchner, A., & Schrauf, M. (2015). What determines the take-over time? an integrated model approach of driver take-over after automated driving. *Accident Analysis & Prevention*, *78*, 212–221.

Perspectives on Computational Models of Learning and Forgetting

Florian Sense[1, 2, 3] (f.sense@rug.nl)
Tiffany S. Jastrzembski[4] (tiffany.jastrzembski@us.af.mil)
Michael C. Mozer[5] (mozer@colorado.edu)
Michael Krusmark[3] (michael.krusmark.ctr@us.af.mil)
Hedderik van Rijn[1, 2] (d.h.van.rijn@rug.nl)

[1] Department of Experimental Psychology, University of Groningen, The Netherlands
[2] Behavioral and Cognitive Neuroscience, University of Groningen, The Netherlands
[3] L-3 Technologies, Wright-Patterson Air Force Base, Dayton, OH, USA
[4] Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, OH, USA
[5] Department of Computer Science & Institute of Cognitive Science,
University of Colorado, Boulder, CO, USA

Abstract

Technological developments have spawned a range of educational software that strives to enhance learning through personalized adaptation. The success of these systems depends on how accurate the knowledge state of individual learners is modeled over time. Computer scientists have been at the forefront of development for these kinds of distributed learning systems and have primarily relied on data-driven algorithms to trace knowledge acquisition in noisy and complex learning domains. Meanwhile, research psychologists have primarily relied on data collected in controlled laboratory settings to develop and validate theory-driven computational models, but have not devoted much exploration to learning in naturalistic environments. The two fields have largely operated in parallel despite considerable overlap in goals. We argue that mutual benefits would result from identifying and implementing more accurate methods to model the temporal dynamics of learning and forgetting for individual learners. Here we discuss recent efforts in developing adaptive learning technologies to highlight the strengths and weaknesses inherent in the typical approaches of both fields. We argue that a closer collaboration between the educational machine learning/data mining and cognitive psychology communities would be a productive and exciting direction for adaptive learning system application to move in.

Keywords: learning; forgetting; computational models; recurrent neural networks; process models; naturalistic data; educational application

Introduction

Imagine leading cognitive scientists came together for a conference—in Montreal, for example—and decided to build the best possible adaptive system to support student learning. A successful adaptive learning system would draw upon our theoretical understanding of human memory and its temporal dynamics: How does knowledge and skill develop with practice? How do memory traces decay over time? Which individual differences in these processes can be exploited to best adapt to individual learners?

Taking this hypothetical endeavor seriously is a productive thought experiment because it makes explicit the gap between our theoretical understanding—based primarily on research conducted in psychology laboratories—and practical applications—worked on primarily by computer scientists.

These two disciplines have largely operated in parallel, and both fields could benefit greatly from collaborating more closely. Mutual benefits for coming together will likely include an enhanced theoretical understanding of learning and memory through access to big, naturalistic data; and improved practical applications achieved through exploitation of robust and well-studied psychological principles.

Here, we will discuss a number of recent efforts to help bridge this interdisciplinary gap. We will present promising approaches to build adaptive learning systems from both the computer science and cognitive science/psychology fields, highlighting the strengths and weaknesses afforded by each type of approach. We will focus and structure the discussion around two recent reports stemming from real-world, educationally relevant use-cases: (1) the second language acquisition modeling (SLAM) challenge put forward by Duolingo, and (2) a comparison of the utility of different computational models to personalize review in a middle school classroom.

Duolingo's SLAM Challenge

The well-known online language-learning platform Duolingo[1] recently posed a challenge to the scientific community. They made data available from more than 6,000 users who independently studied English, Spanish, or French at their own pace, across a duration of 30 days on their platform. Using a corpus of 7+ million annotated words, Duolingo invited research teams to submit computational models to predict users' performance at a later point.[2] In their report of the competing models, they frame this approach—

[1] https://www.duolingo.com/

[2] Interestingly, Settles et al. describe the task as: "Given a history of errors made by learners of a second language, the task is to predict errors that they are likely to make at arbitrary points in the future" (2018, p. 56).

second language acquisition modeling (SLAM)—as a new computational task (Settles, Brust, Gustafson, Hagiwara, & Madnani, 2018). Settles et al. elaborate that educational software has made advances in simpler domains but that less is known about how beginners acquire second languages in realistic settings. As such, their challenge is a special case of "building the best possible adaptive learning system."

Fifteen research teams responded to Duolingo's challenge, encompassing a multitude of approaches used to submit predictions. Most competing teams came from the field of natural language processing due to the fact that Duolingo posed the challenge in the context of a large computational linguistics conference.[3] An analysis of the types of algorithms used to power the predictions suggested that non-linear algorithms—recurrent neural networks (RNNs)—were especially successful, while linear models—item response theory variants—were least successful (Settles et al., 2018; Table 3). In fact, the top models demonstrating the highest predictive validity were all considered non-linear, suggesting that SLAM was mainly approached as deep knowledge tracing (Piech, Bassen, Huang, & Ganguli, 2015) in which RNNs are used to trace student performance over time.

It is interesting to note that none of the teams who submitted model predictions explicitly accounted for the cognitive processing mechanisms involved or how those processes unfold over time. These types of *process models* have been the focus of study in cognitive psychological research, but that research has remained largely in the realm of controlled, laboratory tasks.

In the following two subsections, we will discuss RNNs and process models respectively, to highlight the strengths and weaknesses of both types of models.

Recurrent neural networks (RNNs)

The dominance of RNNs in the Duolingo challenge is not surprising, given their flexibility in discovering useful representations from large amounts of data (LeCun, Bengio, & Hinton, 2015), enabling these models to leverage the rich meta-data available for each instance in the corpus (see Figure 3 in Settles et al., 2018). What is surprising, however, is that these models do not have a clear representation of time, which is of course a crucial dimension of learning (Bloom, 1974). Settles et al. state that none of the models explicitly considered that the passage of time affected acquisition and/or forgetting. This disregard for the temporal dynamics of learning and retention seems surprising given what is known about the spacing effect (e.g., Bahrick, Bahrick, Bahrick, & Bahrick, 1993). It is further surprising to glean that Duolingo itself explicitly models time non-linearly, taking the shape of the forgetting curve into account (Settles & Meeder, 2016).

An analysis of the features that the different models used (see Section 5.2 and Table 4 in Settles et al., 2018) suggests that only the *response time* and *days in course* features had marginally significant effects on the quality of predictions—the modeling architecture (RNN or additive IRT) was the main driver of the differences between the teams. Notably, the *days in course* information for each entry could have been translated to the time that elapsed since the last encounter with an item (i.e., lag-time) in order to explicitly model forgetting as a non-linear function of lag-time. Instead, however, "forgetting was either modeled through engineered features (e.g., user/token histories), or opaquely handled by sequential RNN architectures" (Settles et al., 2018; Section 4).

Simply considering the *sequence* in which events occurred (rather than lag-time) is common in knowledge tracing models (Corbett & Anderson, 1995) and often works well because student behavior is usually modeled in a single session. Consequently, most Bayesian knowledge tracing models do not assume that forgetting takes place at all (see Khajah, Lindsey, & Mozer, 2016 for a BKT variant that does consider forgetting). The benefit of considering lag-time between (rather than the mere sequence of) events as input might only emerge if data are modeled on sufficiently long timescales, across which accurately modeling forgetting curves should be more important.

In a recent effort, Mozer, Kazakov, and Lindsey (2017) introduced an explicit representation of continuous time (CT) in a RNN that they trained on 11 different data sets. The hypothesis behind creating the CT-RNN variant was that including certain constraints might guide the model in its learning—essentially protecting it against its own flexibility (Mozer et al., 2017). Mozer et al. motivate their approach by drawing a helpful analogy with vision, in which models constrained to take known regularities into account outperform unconstrained models (in decyphering handwriting, for example: LeCun, Bottou, Bengio, & Haffner, 1998).To the surprise of the authors, their CT-RNN did not perform any better than the RNN that did not take CT into account, but was otherwise functionally identical. What is more: removing elapsed time from the input stream altogether did not impair the default RNN's performance by more than 5% at most, suggesting that that it did not incorporate temporal information to the extent one might expect.

Their null findings are surprising in light of earlier work that demonstrated the power of taking statistical regularities in the temporal dynamics of forgetting into account. For example, Khajah, Lindsey, and Mozer (2016) extended a Bayesian knowledge tracing model and showed that it performed as well as a RNN knowledge tracing model. Their extensions were based on psychological principles—such as exponential decay of knowledge over time, which is usually not assumed in Bayesian knowledge tracing—that constrained the potential patterns that their model could learn from the data relative to the deep knowledge tracing model.

[3] Specifically, the "13th Workshop on Innovative Use of NLP for Building Educational Applications" held at NAACL-HLT 2018 (http://naacl2018.org/).

More importantly, they highlight the fact that the processes that are assumed to influence learning and forgetting are explicitly expressed in the model's specification: The model parameters correspond to psychological concepts of theoretical relevance. For example, how quickly skill X decays for student Y, or how much students vary in their abilities.

RNNs are currently the preferred choice of computer scientists because of their flexibility to learn arbitrary representations from copious amounts of data. The very architecture guaranteeing this flexibility, however, poses a risk to overfitting the data and makes it extremely difficult to interrogate the model. For adaptive learning systems to be used in practice, systems powered by RNNs may preclude the ability of the system to understand what the learner may optimally require or why the learner is struggling. For these reasons, researchers in psychology—whose main goal is to describe underlying cognitive processes—have not embraced RNNs. Instead, they have developed process models.

Process models

In process models, theoretical assumptions regarding underlying cognitive processes are hard-coded in the model itself. A prime example of an overarching architecture of process models is the Adaptive Control of Thought–Rational (ACT-R; Anderson, 2007) framework[4], which implements testable theories of human memory processing, and supports the creation of cognitive models that are capable of predicting and explaining human behavior. ACT-R has been used to successfully account for a depth and breadth of phenomena, including language comprehension, learning and memory, problem solving and decision, and even interpretation of fMRI data.

With regards to adaptive learning systems, ACT-R has been leveraged by the intelligent tutoring community to minimize the distance between student and expert models. In the case of algebra tutors, for example, ACT-R models each step for solving a problem explicitly, and functions by identifying the root cause for student errors. It then provides the appropriate assistance and mentoring for the individual student to remediate the identified error. These cognitive tutors are highly successful for helping students *acquire* knowledge (Anderson, Corbett, Koedinger, & Pelletier, 1995). Practically speaking, however, they fail to include decay mechanisms, so they lack the ability to account for maintenance or sustainment needs long-term.

A number of process models have focused on and extended ACT-R's declarative memory module to model the temporal dynamics of learning and forgetting in greater detail. Pavlik and Anderson (2005) extended ACT-R to account for effects of spacing using an activation-based decay mechanism. They applied this model iteratively and demonstrated success in making real-time predictions for a language learning task, nicely pushing the bounds of computational modeling application for real-world educational use. More recent extensions incorporated response latencies for each learning event to better trace memory strengths over time, showing promise in both laboratory (Sense, Behrens, Meijer, & van Rijn, 2016) and real-life learning tasks (Sense, van der Velde, & van Rijn, 2018; van Rijn, van Maanen, & van Woudenberg, 2009). In addition, this model fared well when evaluated against a range of *theoretical* criteria (see Walsh, Gluck, Gunzelmann, Jastrzembski, & Krusmark, 2018), however, gaps were noted in its ability to make out-of-sample predictions, particularly at long temporal horizons, or to account for the speeded benefit of relearning when initial practice was initially more spaced.

The Predictive Performance Equation (PPE) is another model that explicitly captures the spacing effect, motivated in its development to remediate limitations of existing models. PPE leverages and combines elements of the General Performance Equation (Anderson & Schunn, 2000), ACT-R, and the New Theory of Disuse (Bjork & Bjork, 1992). This novel computational account of the spacing effect has demonstrated its theoretical and applied validity across a breadth of empirical data (see Walsh et al., 2018). PPE has built upon the shoulders of giants previously described and pushed into the *prescriptive* realm for real-world applications. This means that real-time predictions are iteratively made and successive, optimal training schedules are immediately delivered to the individual learner. PPE has successfully been applied to the domain of cardiopulmonary resuscitation (CPR), demonstrating greater performance effectiveness and minimized training time to acquire and sustain proficiency through personalized, precision learning capability (Jastrzembski et al., 2017).

PPE is unusual in its focus on *prescribing* training schedules in real-life tasks and conditions (but also see, e.g.: Mozer, Pashler, Cepeda, Lindsey, & Vul, 2009), as most process models are primarily developed and evaluated for theoretical purposes. PPE exemplifies the capabilities and limitations of process models more generally: When the relevant processes in a particular domain are mapped onto the mechanics of a model, those models can extrapolate from the available data to make cognitively-plausible prescriptions. Model parameters directly map onto concepts relevant to the modelled domain and can be interpreted and communicated meaningfully (e.g., "Your ability is very high, but this is an unusually difficult fact to learn. You should rehearse this item four hours sooner than the other facts in this set.")

The downside, however, is that process models do not readily translate to new domains or even similar tasks within the same domain. Model parameters that capture individual learning and forgetting signatures often vary across domains and tasks (e.g., Sense et al., 2016). Therefore, using the parameters estimated for a person in one domain, does not mean their performance profile can automatically be accurately predicted in another domain. However, recent work with PPE showed that prior data may be used to *inform* free parameters (Collins, Gluck, Walsh, & Krusmark, 2017; Collins, Gluck, Walsh, Krusmark, & Gunzelmann, 2016),

[4] http://act-r.psy.cmu.edu/about/

indicating that the model does not have to start from scratch in every domain.

Another issue is that most process models with potential for adaptive learning are based on very sparse inputs: lag-time, sometimes response latency, and accuracy—which is often aggregated to reduce noise. Thus, the models are not inherently equipped to leverage the rich meta-data available in, for example, the Duolingo data in the way that RNNs are.

In the final section, we will discuss potential ways of "moving forward" but before, we turn from an online learning platform to the classroom in order to discuss a recent effort to deploy adaptive learning software in realistic educational settings.

Personalized Review in the Classroom

Duolingo's challenge to the scientific community is instructive because it reflects a clearly defined task that an adaptive learning system must perform: modeling second language acquisition (Settles et al., 2018), i.e., predicting future performance given a corpus of learning history. The preceding discussion of how well a number of computational models might be able to perform this task is a productive way to compare the models' theoretical assumptions. If we take the goal of *building the best possible adaptive learning system* seriously, however, we must also keep the end users in mind: the learners.

Today, learners increasingly engage with study materials in distributed learning environments and the culture of learning is changing. While lectures will be scheduled at fixed times, more and more aspects of learning are now self-directed, self-paced, and available on demand in online learning environments. Traditional, structured classroom settings, which are different from Duolingo's learning environment, progressively move towards incorporating distributed learning approaches to aid face-to-face interactions (e.g., Sense et al., 2018). The *best* adaptive learning system would function in realistic, modern educational settings, in which learners follow courses that expose them to materials in a prescribed sequence; in which there might be regular quizzes on subsets of the material; and in which the goal is to perform well on a (cumulative) exam at the end of the course. The ideal system would be able to inform each learner about their progress, the current state of their knowledge, which elements of the course they should focus on, and assist them in their self-regulated learning decisions (Bjork, Dunlosky, & Kornell, 2013).

One elucidatory effort deployed retrieval-practice software as part of the curriculum in a middle school (Lindsey, Shroyer, Pashler, & Mozer, 2014). In a semester-long Spanish course, 179 students engaged with a flashcard tutoring system during class time. Each week, they completed three 20- to 30-minute sessions: In the first and second, the week's new materials were studied to proficiency before reviewing old materials; in the third, a test of the week's new materials was administered. The authors tested three different algorithms that scheduled items during review. The personalized spacing algorithm resulted in the highest performance on the cumulative end-of-semester exam, with especially high performance for items that were introduced early in the semester (Lindsey et al., 2014). The algorithm was dubbed DASH—because it incorporated information regarding item difficulty, student ability, and study history—and the authors argue that their model is in principle agnostic with regards to the domain that is modeled as long as knowledge in that domain can be deconstructed into "primitive knowledge components" (Lindsey et al., 2014, p. 643), which is comparable to the assumptions made by ACT-R in general and PPE in particular (see above).

Mozer and Lindsey (2016) discuss the DASH framework more generally in a recent book chapter—aptly subtitled "psychological theory matters in the big data era"— in which they argue that theory-inspired models such as ACT-R and the multiscale context model (Mozer et al., 2009) can inform theory-agnostic machine learning approaches, specifically collaborative filtering. In this framework, collaborative filtering is used to estimate difficulty and ability from the study history (again: DASH) to infer a student's knowledge state. The generalized power-law of forgetting (Wixted & Carpenter, 2007) can then be used to project the decay of knowledge into the future. Mozer and Lindsey discuss variations of their DASH framework that vary with regards to the information that is considered when instantiating forgetting curves. Their simulation results suggest that for the tested scenarios, individual differences in both learning and forgetting should be considered and that models do much worse if they do not take forgetting into account at all. In two experiments, the authors provide strong empirical evidence that personalized review is more effective than other forms of spacing, which is in line with other research rejecting one-size-fits-all approaches to spacing (Mettler, Massey, & Kellman, 2016).

Conducting experiments of this kind in schools imposes additional administrative and logistic costs on a research project that are not required if large online learning platforms make their data available to researchers (e.g., Ridgeway, Mozer, & Bowles, 2017). A more accessible, educationally relevant context for most researchers might be provided by the classrooms of the universities they work at (e.g., Sense et al., 2018). Ultimately, the best possible adaptive learning system must be tested for effectiveness and usability by real learners, not on historical data.

Moving Forward

Moving forward, we believe it is crucial that cognitive scientists engage with the educational data mining community in order to test their process models with naturalistic data. This will allow cognitive scientists to demonstrate the usefulness of formulating relevant cognitive processes explicitly and to learn from approaches commonly used to model learning in computer science. A productive way forward might be to formally evaluate models of different types against each other to map out the boundary conditions for which the strengths and weaknesses of each class apply. For example: In which domains does each type of model fare best? What types of data does each type of model optimally function with? And perhaps most critically, can the strengths of one model alleviate the weaknesses of another through integration?

One potential path towards leveraging the strengths of both process models *and* RNNs is to have the models collaborate when making predictions. The DASH model proposed by Lindsey et al (2014), for example, could be simultaneously fit with a RNN using gradient descent. Instead of making independent predictions, the models would sum the two model predictions to make a single prediction. With predictions thus combined, the RNN will learn the residual between the restricted but interpretable DASH and the actual data. This would maintain the interpretable parameters of DASH and exploit the flexibility of the RNN at the same time. The specific implementation of DASH proposed by Lindsey et al. could be replaced with any other process model, of course, and the collaborative predictions could be weighted to give preferential treatment to either the process model or the RNN.

Although significant progress has been made to close the gap between computational models and educational or training practice application, it is important to realize that literature is sparse or nonexistent for timescales and contexts most keenly relevant to formal educational institutions where typical summer breaks invoke an inherent acceptance of knowledge decay each year (Cooper, Nye, Charlton, Lindsay, & Greathouse, 1996; McCombs, Augustine, & Schwartz, 2011); or for military training, where irregular delays between training and use is common and maintenance of readiness for high-risk, low-volume skills is a significant challenge. Thus, additional research must be conducted to evaluate the applied utility of any computational model that could be of practical use.

We argue that a multidisciplinary, collaborative approach bringing the power of neural network and process modeling approaches together, would be an exciting direction for adaptive learning system application to move in (also see Mozer, Wiseheart, & Novikoff, 2019). It would acknowledge the value of the human-in-the-loop by integrating our theoretical understanding of the human memory system with RNNs' ability to make sense of large data; thereby pulling their affordances together in a unified task to build the best adaptive learning system possible.

Acknowledgements

This research was supported by NSF grants SES-1461535 and DR-L1631428.

References

Anderson, J. R. (2007). *How can the human mind exist in the physical universe? Oxford Series on Cognitive Models and Architectures*. New York, NY: Oxford University Press.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of Learning Sciences*, *4*(2), 167–207.

Anderson, J. R., & Schunn, C. D. (2000). Implications of the ACT-R Learning Theory : No Magic Bullets. In R. Glaser (Ed.), *Advances in Instructional Psychology: Educational Design and Cognitive Science* (Vol. 5, pp. 1–34). Mahwah, NJ: Lawrence Erlbaum.

Bahrick, H. P., Bahrick, L. E., Bahrick, A. S., & Bahrick, P. E. (1993). Maintenance of Foreign Language Vocabulary and the Spacing Effect. *Psychological Science*, *4*(5), 316–321. http://doi.org/10.1111/j.1467-9280.1993.tb00571.x

Bjork, R. A., & Bjork, E. L. (1992). A new theory of disuse and an old theory of stimulus fluctuation. In A. Healy, S. Kosslyn, & R. Shiffrin (Eds.), *From Learning Processes to Cognitive Processes: Essays in Honor of William K. Estes* (pp. 35–67). Hillsdale, NJ: Erlbaum.

Bjork, R. A., Dunlosky, J., & Kornell, N. (2013). Self-Regulated Learning: Beliefs, Techniques, and Illusions. *Annual Review of Psychology*, *64*, 417–44. http://doi.org/10.1146/annurev-psych-113011-143823

Bloom, B. S. (1974). Time and Learning. *American Psychologist*, *29*(9), 682–688. http://doi.org/10.1037/h0037632

Collins, M. G., Gluck, K. A., Walsh, M., & Krusmark, M. (2017). Using Prior Data to Inform Initial Performance Predictions of Individual Students. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society* (pp. 1800–1805). Madison, WI.

Collins, M. G., Gluck, K. A., Walsh, M., Krusmark, M., & Gunzelmann, G. (2016). Using Prior Data to Inform Model Parameters in the Predictive Performance Equation. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society* (pp. 75–80). Philadelphia, PA.

Cooper, H., Nye, B., Charlton, K., Lindsay, J., & Greathouse, S. (1996). The Effects of Summer Vacation on Achievement Test Scores: A Narrative and Meta-Analytic Review. *Review of Educational Research*, *66*(3), 227–268. http://doi.org/10.3102/00346543066003227

Corbett, A. T., & Anderson, J. R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, *4*, 253–278.

Jastrzembski, T., Walsh, M. M., Krusmark, M., Kardong-Edgren, S., Oermann, M., Dufour, K., … Stefanidis, D. (2017). Personalizing training to acquire and sustain competence through use of a cognitive model. In D. D. Schmorrow & C. M. Fidopiastis (Eds.), *Augmented cognition. Enhancing cognition and behavior in complex environments* (pp. 148–161). Switzerland: Sprinter International Publishing AG.

Khajah, M. M., Lindsey, R. V, & Mozer, M. C. (2016). How Deep is Knowledge Tracing? *ArXiv Preprint ArXiv:1604.02416v2*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*. http://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998).

Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lindsey, R., Shroyer, J. D., Pashler, H., & Mozer, M. C. (2014). Improving Students' Long-Term Knowledge Retention Through Personalized Review. *Psychological Science*, *25*(3), 639–647. http://doi.org/10.1177/0956797613504302

McCombs, J. S., Augustine, C. H., & Schwartz, H. L. (2011). *Making summer count: How summer programs can boost children's learning*. Rand Corporation.

Mettler, E., Massey, C. M., & Kellman, P. J. (2016). A Comparison of Adaptive and Fixed Schedules of Practice. *Journal of Experimental Psychology: General*, *145*(7), 897–917. http://doi.org/10.1037/xge0000170

Mozer, M. C., Kazakov, D., & Lindsey, R. V. (2017). Discrete-Event Continuous-Time Recurrent Nets. *ArXiv Preprint ArXiv:1710.04110*.

Mozer, M. C., & Lindsey, R. (2016). Predicting and Improving Memory Retention: Psychological Theory Matters in the Big Data Era. In M. N. Jones (Ed.), *Big Data in Cognitive Science.* (pp. 43–73). Psychology Press.

Mozer, M. C., Pashler, H., Cepeda, N., Lindsey, R., & Vul, E. (2009). Predicting the optimal spacing of study: A multiscale context model of memory. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems* (Vol. 22, pp. 1321–1329). La Jolla, CA: NIPS Foundation.

Mozer, M. C., Wiseheart, M., & Novikoff, T. P. (2019). Artificial intelligence to support human instruction. *Proceedings of the National Academy of Sciences*, *116*(10), 3953–3955. http://doi.org/10.1073/pnas.1900370116

Pavlik, P. I., & Anderson, J. R. (2005). Practice and Forgetting Effects on Vocabulary Memory: An Activation-based Model of the Spacing Effect. *Cognitive Science*, *29*(4), 559–86. http://doi.org/10.1207/s15516709cog0000_14

Piech, C., Bassen, J., Huang, J., & Ganguli, S. (2015). Deep Knowledge Tracing. *Advances in Neural Information Processing Systems*, 505–513.

Ridgeway, K., Mozer, M. C., & Bowles, A. R. (2017). Forgetting of Foreign-Language Skills: A Corpus-Based Analysis of Online Tutoring Software. *Cognitive Science*, *41*, 924–949. http://doi.org/10.1111/cogs.12385

Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials. *Topics in Cognitive Science*, *8*(1), 305–321. http://doi.org/10.1111/tops.12183

Sense, F., van der Velde, M., & van Rijn, H. (2018). Deploying a Model-based Adaptive Fact-Learning System in a University Course. In *Proceedings of the 16th International Conference on Cognitive Modeling* (p. 138). Madison, WI.

Settles, B., Brust, C., Gustafson, E., Hagiwara, M., & Madnani, N. (2018). Second Language Acquisition Modeling. In *Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 56–65).

Settles, B., & Meeder, B. (2016). A Trainable Spaced Repetition Model for Language Learning. *Association for Computational Linguistic (ACL)*, 1848–1858.

van Rijn, H., van Maanen, L., & van Woudenberg, M. (2009). Passing the Test: Improving Learning Gains by Balancing Spacing and Testing Effects. In *Proceedings of the 9th International Conference on Cognitive Modeling* (pp. 110–115). Manchester, UK.

Walsh, M. M., Gluck, K. A., Gunzelmann, G., Jastrzembski, T., & Krusmark, M. (2018). Evaluating the Theoretic Adequacy and Applied Potential of Computational Models of the Spacing Effect. *Cognitive Science*, *42*, 644–691. http://doi.org/10.1111/cogs.12602

Wixted, J. T., & Carpenter, S. K. (2007). The Wickelgren Power Law and the Ebbinghaus Savings Function, *18*(2), 133–134.

Transfer effects of varied practice and adaptation to changes in complex skill acquisition

Roderick Yang Terng Seow (yseow@andrew.cmu.edu)
Shawn Betts (sabetts@andrew.cmu.edu)
John R. Anderson (ja0s@andrew.cmu.edu)
Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave
Pittsburgh, PA 15213 USA

Abstract

Varied training in comparison to consistent training has been shown to benefit transfer to novel conditions within the motor learning paradigm. However, it is unclear if these benefits of variable training extend to complex skills such as driving. Unlike simple motor skills, these complex skills require individuals simultaneously to learn the mapping between ones actions and their consequences and also to integrate this knowledge into continuous and dynamic responses to the changing demands of the environment. In the current work, we compare observed data and an ACT-R model of complex skill acquisition on a navigational video game task (Space Track). Participants trained either on one or two levels of thrust. Performance on a transfer test was better in the varied training conditions in both humans and model. Performance in both humans and model was also differentially influenced by the most recently practiced thrust level. Further analyses revealed large differences between model and human behavior on more detailed measures, which suggests that that the model achieves the same overall performance through different strategies. We discuss these findings and their implications for the ACT-R model of skill acquisition.

Keywords: varied practice; transfer; adaptation; ACT-R; complex skill acquisition

Introduction

Transfer learning is the phenomenon in which practice on one task facilitates the learning of a related task, which reduces the time needed to attain a certain level of skill. This phenomenon has been studied in various domains such as mathematical problem solving (e.g. Speelman and Kirsner, 2001), perceptual categorization and discrimination (e.g. McGovern et al., 2012), and sensorimotor learning (e.g. Goodwin et al., 1998). Within the domain of sensorimotor learning, practicing on varied task parameters has been shown to facilitate more transfer to new task parameters as compared to consistent practice. For instance, when the transfer task is to toss a beanbag to a target at a set distance, participants who trained on different target distances excluding the transfer target perform better than those who trained on just one target distance (Kerr and Booth, 1978).

In the tasks used to investigate the effects of varied practice, one common feature is that the goal of the task is often closely related to the sensory consequence of the sensorimotor mapping that the subject needs to learn. For example, the goal in a visuomotor rotation task is to maneuver a cursor towards a virtual target in the presence of perturbations (e.g. Braun et al., 2009). These perturbations cause the motion of the cursor to rotate with respect to the motion of the controller and successful participants are hypothesized to learn this new mapping between the motion of their hand and the motion of the cursor. However, it remains unclear is unclear if the benefits of varied practice extend to more complex tasks in which the acquisition of sensorimotor maps is necessary but insufficient for high performance. For example, assuming that the goal of a driver is to get from point A to B in the fastest and safest manner, successful driving involves not only more than just learning how the movement of ones foot on the accelerator translates to the cars motion, but also the ability to come up with an action plan to navigate the upcoming obstacles or road hazards. Hence, one of the goals of this study is to answer the following question: When learning complex skills, where the sensorimotor map is only a part of the skills needed to accomplish the task goal, does varied training still outperform consistent training with regard to the transfer of performance to novel task parameters?

Task

Space Track was originally a video game developed by Anderson et al. (in press) as part of a study on the transfer of complex skills. Just like driving, mastering Space Track is a complex skill because it requires one to integrate perceptual, motor, and cognitive components. Expertise arises from having gained an intuitive understanding of the physics of the game and the ability to use that knowledge towards planning sequences of key presses to overcome various situations. In Space Track, players control a space ship in a frictionless environment using three keys: thrust (W), rotate clockwise (A), and rotate counterclockwise (D). Players earn 25 points by successfully navigating the ship along each rectangular track segment and lose 100 points when the space ship crashes into the walls of the track. Figure 1 shows a schematic of the task. Finding a good speed is crucial for performance one needs to fly fast enough to cover as much distance as possible but also slow enough to avoid losing control of the ship and crashing.

To create changes in the task environment, we manipulated the amount of thrust the ship receives for the same duration that the thrust key is depressed. When the thrust key is depressed, a vector of x pixels per second in the current direction of the ship is added each game tick, which is 1/30th of a second. For the same duration of key press, a game with higher thrust would cause the ship to fly faster than a game with lower thrust. Mastery of the game relies on adequately predicting and controlling the motion of the space ship. Thus, players would have to retune their control parameters when faced with a different thrust level.

We created three game types, each with a different thrust level. High thrust games (H) added 0.6 pixels / tick to the ships velocity vector for each tick that the thrust key was depressed. Medium (M) and low (L) thrust games added 0.4 and 0.2 pixels / tick respectively. With these three game types, we created four training conditions as follows: LLLLM, HHHHM, LHLHM, and HLHLM, where each letter stands for one block of 8 x 3-minute games. Figure 2 provides a pictorial representation of the task design. For instance, a player in the LLLLM condition would play 4 blocks (32 games) of low thrust followed by 1 block (8 games) of medium thrust. For our analyses, the first 4 blocks will be referred to as the training blocks, and the last block of medium thrust in all conditions will be referred to as the test block. Participants in the consistent training group will be assigned to either LLLLM or HHHHM, while those in the varied training group will be assigned to either HLHLM or LHLHM. Our rationale for using two different conditions in the consistent training group is to separate adaptation effects due to consistent vs. varied training and effects due to training on a high vs. low thrust. We used two different conditions in the varied training group to account for possible block order effects.

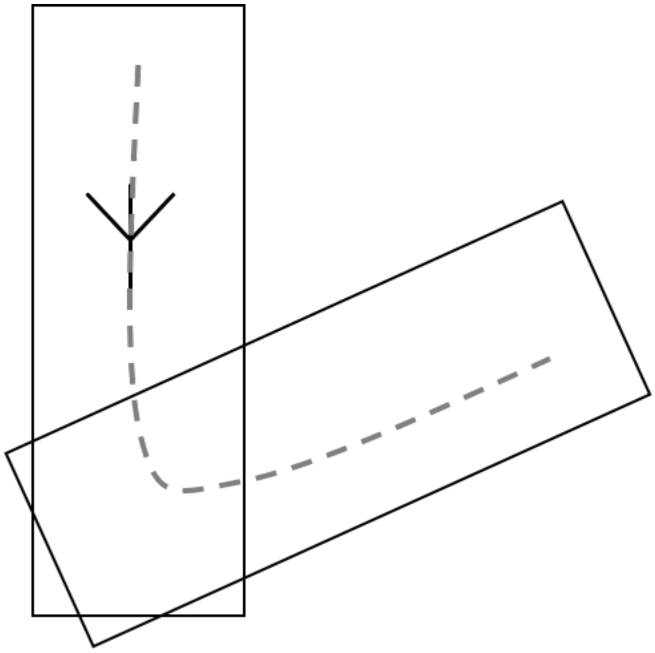


Figure 1: Schematic of Space Track. The goal is to navigate a space ship along a racetrack with rectangular segments. The dashed line displays a potential trajectory along two consecutive segments.

Experiment

80 participants, 22 females and 58 males, ranging in age from 21 to 65 years old (mean = 31.0) completed both experiment sessions through Amazon Mechanical Turk. Participants were paid $5 for the first session and $10 for the second session plus a bonus of $0.03 per 100 points.

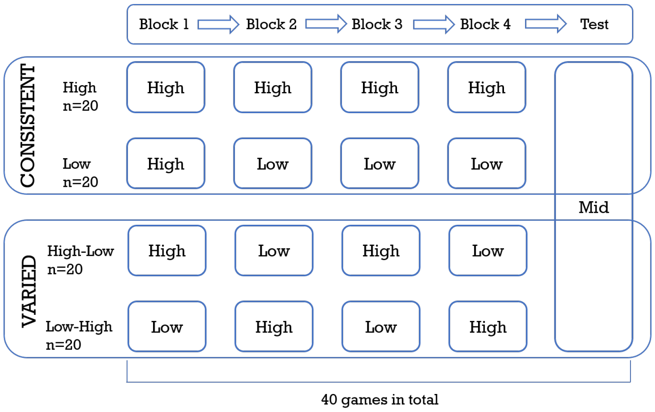


Figure 2: Task design. Each row represents one condition, and each box represents one block of 8 games.

The experiment consisted of two sessions. In the first session, participants filled out a demographic questionnaire, then proceeded to complete the first 20 games. Participants that passed a set of inclusion criteria were then invited to the second session, which consisted of another 20 games. During a game, if 20 seconds elapse without the participant pressing a key, a pop up with a ready to restart button will appear. The inclusion criteria for the second session are 1. No more than 3 resets due to inactivity and 2. Either at least 500 points in at least 3 out of the 20 games, or that the average of games 17 to 20 is at least 100 points higher than the average of game 1 to 4. These criteria were put in place to maximize recruiting only players who were sufficiently attentive and showed signs of learning. Using those criteria, 66 number of players who finished the first session were excluded from participating in the second session. Recruitment continued until 20 participants per condition successfully completed both sessions.

Behavioral results

Figure 3 displays the points per game for each condition; the following analyses will focus only on data from the human players (in red). To get a measure of test performance for each participant, we averaged each participants points across their 8 games of the test block (games beyond the rightmost dashed line in Figure 3).

We then fitted a linear regression with average test points as the dependent variable. The independent variables of interest were training group (consistent or varied) and the thrust type on block 4, which is the last practiced thrust type before the transfer test (high or low). To account for the possible effects of video gaming experience and other participant characteristics on transfer performance, we included the following as nuisance variables: age, gender, the dominant hand used to control movement in games, and the number of hours per week spent on different genres of video games.

Variable training outperforms consistent training on transfer test

From the results of the regression (adjusted $r^2 = 0.32$), varied training ($\beta = 265.46$, Std. Error = 103.11, $p < 0.05$), low thrust on block 4 ($\beta = -272.38$, Std. Error = 98.08, $p < 0.05$), hours per week spent on 2D action ($\beta = 204.88$, Std. Error =51.78, $p < 0.05$) and 3D shooter games ($\beta = 111.04$, Std. Error = 37.02, $p < 0.05$) significantly predicted test points. Notably, players who received varied training were predicted to outperform their consistent counterparts on the transfer test by 265.46 points. This advantage of varied training is aligned with the variability of practice hypothesis.

Changes in performance depends on the direction and magnitude of switch in thrust level

Thrust type on block 4 also strongly predicted transfer performance ($\beta = 185.521$, Std. Error = 76.376, $p < 0.05$), where participants trained on high thrust outperform those trained on low thrust. While we did not predict an effect of recent thrust level, it might be that training with higher thrusts is more difficult and that switching to lower thrust levels is akin to switching to an easier task, which has been shown to facilitate transfer (e.g. Barch and Lewis, 1954).

If there were behavioral differences between games of different thrust levels, one would expect the largest differences to manifest when initially switching to a new thrust. Hence, to further investigate the effect of switching thrust levels, we analyzed the point difference obtained by subtracting the points earned on the last game of a block from the points earned on the first game of the subsequent block. Point differences are then sorted by switch type. For instance, the point difference between games 33 and 32 for a HHHHM participant would be considered a H to M switch, whereas the point difference between the same numbered games for a LLLLM participant would be considered a L to M switch. Point differences for H to L (games 8 to 9 and 24 to 25 for HLHLM and games 16 to 17 for LHLHM) and L to H (games 16 to 17 for HLHLM and games 8 to 9 and 24 to 25 for LHLHM) were gathered from participants in both varied conditions. Switch types were then re-coded as thrust differences to express a quantitative difference in thrust levels (L to H = 0.6 0.2 = 0.4; L to M = 0.2; H to M = -0.2; H to L = -0.4).

A regression model (adjusted $r^2 = 0.3308$) with thrust difference as the sole predictor of point difference estimated a slope of -949.45 (Std. Error = 95.02, $p < 0.05$). This suggests that increasing thrust by 0.2 would result in a drop of 189.89 in points, providing further evidence that switching from low to high thrust decreases performance while switching from high to low thrust increases performance.

Adaptive Control of Thought – Rational

In a recent study, Anderson et al. (in press) demonstrated that an ACT-R model produced the same learning trajectory as humans do (r = 0.96) in a Space Track task where players would play 40 games at a thrust level of 0.3. There are four key features of ACT-R that enabled this successful simulation. First, there are limits on various human cognitive processes such as attention and response times that constrain how human skill acquisition proceeds. The ACT-R cognitive architecture incorporates realistic performance constraints on the speed and accuracy of perception and action. Second, human participants do not begin learning from scratch, but are informed by explicit instructions about the controls and goals of the task. Through instruction following, ACT-R models also utilize task knowledge to accelerate learning in the initial stages. Third, the improvement in performance with experience is partially governed by increased automaticity and faster deployment of knowledge. ACT-R models capture this by production compilation, a process that gradually proceduralizes declarative knowledge and reduces the time cost of having to retrieve declarative knowledge for action execution. Fourth, human skill mastery also relies on tuning the control parameters of ones actions to predictors of success or failure in the task environment. This is captured in ACT-R by a new Controller module that explores continuous dimensions of performance to identify how to control actions. For instance, one dimension that was explored in Space Track was the speed of the ship that would yield an optimal trade-off between number of segments cleared versus ship crashes.

An open question about the new controller module is whether it responds to environmental changes in the same way humans do. Thus, it becomes of interest to see how it responds to the changes investigated in our experiment.

Control Tuning

Through practice, the model learns the optimal values for 5 control variables: aim, ship speed, thrust duration, when to start making a turn, and the ship's orientation with respect to the angle of the upcoming intersection. For each control variable, the model samples values within a preset range and evaluates the mean rate of return for the sampled values according to relevant feedback. Using that feedback, the module then estimates a quadratic function that describes the relationship between rate of return and control value, which in turn influences how the module samples the next set of control values to try. This process is repeated iteratively throughout the experiment, and the model eventually converges to a truer estimate of the relationship between return and control values.

For the model, relevant feedback comes from two sources: the number of crashes and the number of segments cleared. The weights on these sources determine the contribution of each source of feedback to the estimated rate of return. Different source weights potentially relate to differences in risk attitudes; for instance, a player might adopt a riskier approach, clearing more track segments but also crashing more often than a more cautious player.

For our first set of model simulations, we compared models with different weight ratios on the control variables. The reference model weights both features equally (-1 for a crash, +1 for a cleared segment). One modified model reflects the difference in point values assigned by the game to these features and weights a crash four times the benefit of clearing a segment (-4 to +1). Another modified possibility reflects a loss aversive player by weighting crashes as being eight times a cleared segment (-8 to +1).

While points are the primary indicator of performance on the task, two players could conceivably achieve the same total points through different strategies. For instance, a player might adopt a riskier approach to the game, clearing more track segments but also crashing more often than a more cautious player. To further investigate how switching thrust types influences more fine-grained behavior in both models and humans, we also analyzed the mean speed, number of segments cleared, and the number of crashes per game. For each point of comparison, we obtained the sum of squared errors (SSE), which measures the absolute deviation the average model exhibits with respect to the average human across all 4 conditions and 40 games. These results are presented in Table 1.

Different ratios of good and bad weights

The first set of comparisons comprise of the following models: the base (reference) model with a weight ratio of 1:1, a model with a ratio of 4:1, and a (loss aversive) model with a ratio of 8:1. Of the three models, the worst performing model by far on all measures is the base model. The other two models perform comparably, with the loss aversive model performing slightly better than the 4:1 model on all measures except total points earned. The relatively small differences in model fits possibly suggest that the weighting function of some human players might be best characterized by the 4:1 ratio, which reflects the corresponding contributions of crashes and segments cleared to the total points earned, while the weighting function for other players might be better characterized a the 8:1 ratio, which reflects a disproportionately heightened sensitivity to crashes over segments cleared. For the sake of simplicity, we proceeded to incorporate the 4:1 ratio in our subsequent model simulations.

Figure 3 displays how the average points change as a function of game number for both humans and the 4:1 model in all four training conditions; notice that the model shows the same increases and decreases in performance when the thrust level switches as do humans.

Adding slowdown and a decay on past experiences

While the 4:1 ratio model does qualitatively simulate human behavior adequately on the number of crashes, segments cleared, and the overall points earned (refer to Figures 3 and 5), it does a poorer job of capturing how human players modulate their mean speed across games. Referring to Figure 6, it appears that the model drastically changes its speed whereas human players only make small changes in response to changes in the thrust level. This then motivated the next set of models, where we added slowdown, the ability for the model to actively reduce the spaceship's speed when it overshoots its desired control speed value.

Another model manipulation we investigated was to have the model discount its old experiences. Because the Space Track task used in Anderson et al. did not change its parameters over time, it was unsurprising that a model that weighted all experiences equally would be able to perform comparably with those that discounted old experiences. However, as the task used in the current study does introduce changes in the task parameters, it might be reasonable to expect that a model that decays the weight of old experiences would be able to adapt better to the changes in thrust level. When the task parameters change, it is likely that the same control value will result in different payoffs. For instance, pressing the thrust key for 1 second in a low thrust level will increase the ship's velocity by a smaller amount than pressing the key for 1 second in a high thrust level.

There is evidence from the memory literature for an exponential decay function on the retention of past items in memory (e.g. Rubin et al., 1999). Hence, we chose to discount the weight of a past experience by $.995^t$ where t is the time in seconds.

The second set of comparisons comprise of four models: the 4:1 weight ratio model, which also serves as the reference model for this comparison, an exponential decay model, a model with slowdown, and a model with both slowdown and an exponential decay.

Between the four models, there are two that best fit the human player data; the exponential decay model for total points ($SSE = 5594241$) and crashes ($SSE = 1064$) , and the slowdown and decay model for segments cleared ($SSE = 2564$) and mean speed ($SSE = 15.2$). The presence of a decay function in both models suggests that human players might adapt to their current thrust level by discounting the weight of their past experiences, especially if those experiences were obtained from a different thrust level.

Of the four measures, a model's match on total points is the least important because the total points earned is a composite of the segments cleared and crashes measures. Focusing on the other three measures, the slowdown and decay model appears to be the overall better model, especially because the pure decay model's advantage over the slowdown and decay model in crashes appears to be relatively smaller than its disadvantage in segments cleared and mean speed.

Referring to Figures 4 and 6, the slowdown and decay model shows a large reduction in both the number of segments cleared and its mean speed compared to the reference (4:1) model. This reduction is particularly apparent during the high thrust level blocks, and enables the slowdown and decay model to align better with the mean speed of human players across all conditions. Despite the model's success, it should be noted that the model still exhibits larger modulations in its mean speed in response to changes in thrust level than human players do, suggesting that human players might actively aim to maintain the ship speed within a range of values instead of completely adapting the ship speed to optimize the points earned in games of different thrust levels.

Table 1: Model comparisons

				Sum of Squared Errors (SSE)			
Model	Weight Ratio	Slowdown	Decay	Total Points	Segments Cleared	Crashes	Mean Speed
Base	-1 : +1			25647698	11064	5087	119.8
Weight=4	-4 : +1			6789958	9413	1370	69.7
Weight=8	-8 : +1			7398453	7383	1299	54.2
Slowdown	-4 : +1	✓		19404654	3178	1516	15.5
Decay	-4 : +1		Exponential	5594241	11660	1064	82.1
Slowdown + Decay	-4 : +1	✓	Exponential	15901165	2564	1327	15.2

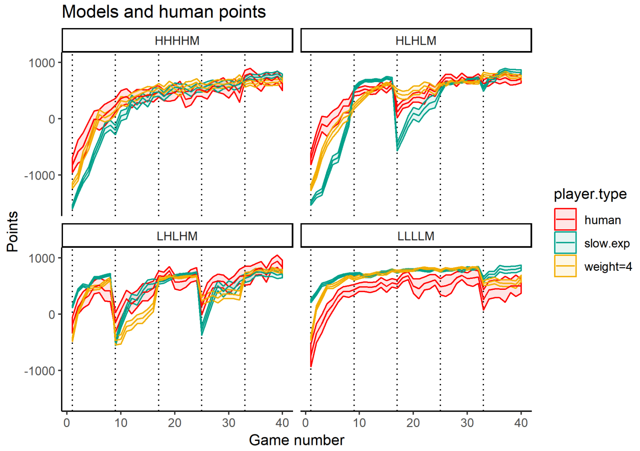


Figure 3: Points by game number for each training condition. Mean human points are in red (n=20 per condition); mean slowdown + decay model points are in green (n=100); mean -4 : +1 weight ratio model points are in yellow (n=100). Shaded areas are S.E.M. Dashed lines indicate the start of a new block.

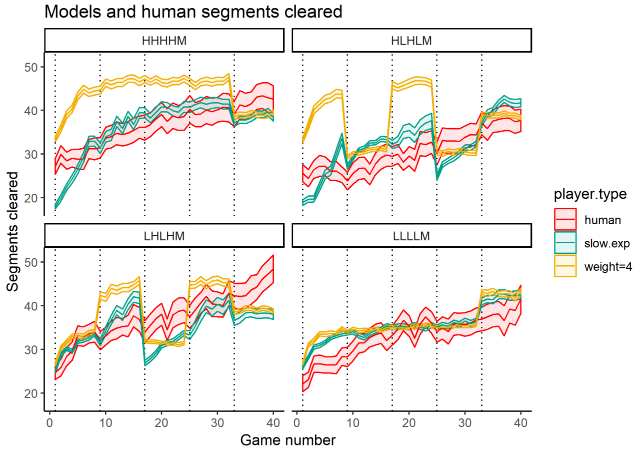


Figure 4: Segments cleared per game.

Conclusion and Further Work

The behavioral results suggest that the variability of practice hypothesis extends beyond simple motor skills to more dynamic and complex skills that require integrating perceptual, motor and cognitive components. However, our results also indicate that a person's performance on a new thrust level is influenced by their most recently experienced thrust level. Thus, transfer performance depends not only on whether one receives consistent or varied practice, but also on the specific parameters within a consistent or varied training schedule.

Switching from a high thrust to a low or medium thrust improves performance while switching from a low thrust to a high or medium thrust decreases performance. However, it is unclear why these switch effects occur, and why there is an asymmetry in these effects. One possible extension involves investigating how different thrust levels affect motor variation. Motor learning often involves the minimization of motor variation such that one is better able to precisely execute an intended action (refer to Dhawale et al., 2017 for a review). In a high thrust game, a small deviation in the duration of a thrust key press from the intended duration would cause the ship to slow down or speed up more drastically than for the same deviation in a low thrust game. Hence, it might be that players trained on high thrust games have more pressure to control and minimize their motor variation. When switching to a lower thrust level, these players easily adapt to the new thrust because they can immediately apply a suitable degree of control on their thrust key presses. In comparison, players trained on low thrust games have less pressure to minimize motor variation. When these players switch to a higher thrust level, they would be forced to grapple with learning a level of control that was previously unnecessary.

Our model comparisons reveal that the best fitting ACT-R models weight negative events more severely than positive ones. As players are rewarded depending on how many points

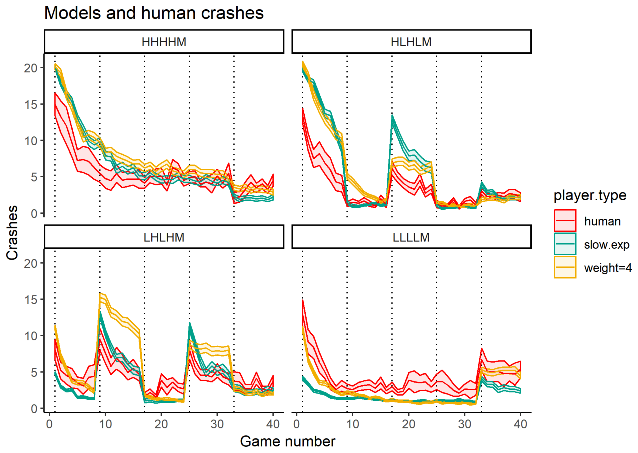


Figure 5: Crashes per game.

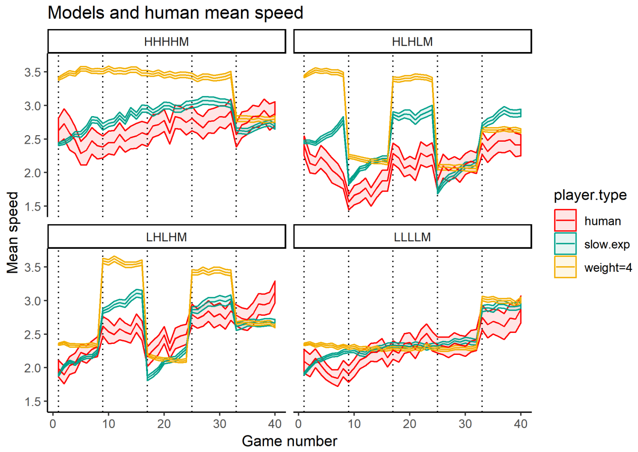


Figure 6: Mean ship speed per game.

they earn, it is reasonable that some players would weight avoiding crashes over clearing track segments in a ratio that reflects their relative contribution to points. Alternatively, as humans have been shown to demonstrate loss aversion in the face of equally valued gambles (e.g. Kahneman and Tversky, 1979), it is also reasonable that some players would place an even greater emphasis on avoiding crashes. Future work would involve investigating if the variability between individual players could be explained by models with different weight ratios.

Our comparisons also provide evidence for including a decay on past experiences. As different thrust levels likely result in different payoffs for the same control setting, a player that discounts old experiences from a previous thrust level would update their estimated payoffs faster when adapting to a new thrust level. More generally, adaptation to changes in the environment is facilitated by prioritizing information learned from recent experiences as these would better reflect the state and reward structure of the current environment.

Finally, while adding the ability to slowdown does improve the models' fit to the human players' mean ship speed, the models still exhibit larger modulations than humans players do when switching between thrust levels. One possibility is that human players are not using points as feedback for ship speed but perhaps using some sense of a comfortable speed. Further work needs to be done to see whether maintaining desired speed can be used as a feedback for the Controller module. Speed control was successfully used as a feedback signal in another video game, YouTurn, described in Anderson et al.; that YouTurn model used speed control to tune one control variable, while using point-related measures to tune other control variables.

Acknowledgments

This research was supported by ONR Grant N00014-15-1-2151 and AFOSR/AFRL award FA9550-18-1-0251.

References

Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (in press). Learning rapid and precise skills. *Psychological Review*.

Braun, D. A., Aertsen, A., Wolpert, D. M., & Mehring, C. (2009). Motor Task Variation Induces Structural Learning. *Current Biology*, *19*(4), 352–357.

Dhawale, A. K., Smith, M. A., & Ölveczky, B. P. (2017). The Role of Variability in Motor Learning. *Annual Review of Neuroscience*, *40*(1), 479–498.

Goodwin, J. E., Eckerson, J. M., Grimes, C. R., & Gordon, P. M. (1998). Effect of Different Quantities of Variable Practice on Acquisition, Retention, and Transfer of An Applied Motor Skill. *Perceptual and Motor Skills*, *87*(1), 147–151.

Kahneman, T. A., Daniel. (1979). Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, *47*(2), 263–292.

Kerr, R., & Booth, B. (1978). Specific and varied practice of motor skill.pdf. *Perceptual and Motor Skills*, 395–401.

McGovern, D. P., Webb, B. S., & Peirce, J. W. (2012). Transfer of perceptual learning between different visual tasks. *Journal of Vision*, *12*(11), 4–4.

Rubin, H. S. C., David. C., & Wenzel, A. (1999). The precise time course of retention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1161–1176.

Speelman, C. P., & Kirsner, K. (2001). Predicting transfer from training performance. *Acta Psychologica*, *108*(3), 247–281.

Less is More: Additional Information Leads to Lower Performance in Tetris Models

Catherine Sibert (siberc@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

Jacob Speicher (speicj@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

Wayne D. Gray (grayw@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

Abstract

Expert performers in complex tasks synthesize a wide variety of information to select the optimal choice at each decision point. For the task of Tetris, the synthesis includes information about the “next” piece in addition to the configuration of pieces currently on the board. While simple models of Tetris are capable of behavior similar to high level human players most (to reduce the combinatorial explosion in computation time) are only aware of the active piece and its possible placement positions. To explore how additional information contributes to expertise, when placing the current ‘on board’ piece, our model also considers placements for the “next piece” (visable to humans in the Preview Box). Though we expected this additional information to result in higher performance, we instead observed a drop in performance, and a shift in behavior away from common human patterns. These results suggest that human experts are not incorporating the additional piece information into their current decision. We speculate about the role of next piece information for expert level players.

Keywords: Expertise, Reinforcement Learning, Machine Learning, Human Performance

Introduction

Complex task environments are, almost by definition, difficult to master and, by extension, difficult to study. In this work we focus on the complex task environment created by the dynamic, decision-making game of Tetris, which we see as the *poster child* for human studies of predictive processing (Clark,2013;Engstrom et al.,2018;Rao & Ballard,1999). Although we have made considerable progress in understanding Tetris play in our laboratory (Lindstedt & Gray,2019), human play represents a confounding of various human limitations that may well be impossible to disentangle *in vivo*. This tangle has led us to machine models of Tetris (Fahey,2015;Gabillon, Ghavamzadeh, & Scherrer,2013;Szita & Lorincz,2006) where we have focused on understanding how the configural properties of the Tetris board can be interpreted by machine models as good placements or bad placements for the currently falling zoid (i.e., Tetris piece) (Sibert, Gray, & Lindstedt,2017;Sibert & Gray,2018).

The attentive reader will note that we initiated the preceding paragraph by alluding to human predictive processing but ended that paragraph by focusing on defining characteristics of good or bad placement decisions by using machine models. This shift is possible as, unlike humans, our models do not actually rotate, transpose, or drop pieces; rather, as in an episode of Star Trek, they simply beam the piece to the desired location. This trick neatly disentangles the *where to place it* decision from the *how to get it there* one (which are the concern for Tetris studies of Predictive Processing (Lindstedt & Gray,2019)).

The current work complicates our models by observing that the classic, Nintendo Entertainment Systems (NES) version of Tetris (which is the version used in the annual *Classic Tetris World Championships, CTWC*) as played by humans, always provides the “next” zoid in a Preview Box (see Figure 1). As the goals of the machine modeling community differ from ours, no prior machine model of Tetris play uses that information.[1] Hence, our current work explores two-piece placement decisions in an attempt to determine whether and, if so, how, attempts to optimize the current placement with respect to the next placement improves the game.

A Very Brief History of Games in Research

Gray (2017) distinguishes among three ways in which modern computer games have been used in psychological research. *Gamification* represents the attempt to use features of game play for more serious work such as gamifying a social media field trial (Rapp, Cena, Gena, Marcengo, & Console,2016), modeling “professional thinking” (Nash & Shaffer,2011), or teaching helicopter flight skills (Proctor, Bauer, & Lucario,2007).

Games as Treatment Conditions represents attempts to use game play as a means of changing some aspect of human behavior, health, intelligence, and so on. Examples involving Tetris include its use to reduce “flashbacks” associated with Posttraumatic Stress Disorder (PTSD) (Holmes, James, Coode-Bate, & Deeprose,2009) or as a placebo control in a

[1] While we do not know for sure why the machine modeling community does not use that information, we do note that doing so extends the search space for moves from approximately 23 placements (between 9 and 34, depending on the zoid) up to 34^{34} placements. The addition of more complex move generation functions that allow zoids to be navigated underneath other zoids further increases the number of placements to be considered at each decision point.

study of the utility of games (Belchior et al.,2013) in expanding older adult's useful field of view (UFOV).

Game-XP refers to the use of game play itself as an experimental or quasi-experimental paradigm. The earliest example of using Tetris for Game-XP (that we are aware of) was for exploring the concept of epistemic or complementary action (Kirsh & Maglio,1994;Destefano, Lindstedt, & Gray,2011). Of course, our past work (cited earlier) as well as the work presented in this paper provide other examples of the use of Tetris for these purposes; that is, an experimental paradigm which we use to seek insights into the low level mechanisms that contribute to skilled performance in dynamic tasks.

Tetris the Task

During a game of Tetris, players navigate a series of pieces, called "zoids", as they fall from the top of the screen into a pile at the bottom of the screen. When a row within the pile becomes full (all ten cells contain a part of a placed zoid) it vanishes, lowering the pile and earning points for the player. More points are earned if more lines are cleared simultaneously, with up to four lines able to be cleared in a single move. The game ends when the pile reaches the top of the screen. A game in progress can be seen in Figure 1.

Though the basic task is simple to understand, game difficulty increases as the player plays. The player is limited in the actions that they can take to move a zoid: zoids can be translated one cell left and right, or rotated 90 or 180 degrees (depending on the zoid) using a single button press, and a complete movement usually requires several button presses. As the game progresses, the pieces fall more quickly, meaning that players must make placement choices and navigate the zoids in increasingly short time periods. At the start of the game, it takes 16 seconds for a zoid to fall from the top of the screen to the bottom. At level 29 (considered by top players to be the "kill screen", and the highest playable level) pieces fall in a third of a second.

In addition to managing the ever increasing game speed, players must weigh the risks and benefits of making different types of line clears. Clearing a single line is fairly simple to do, and most low level players focus on clearing one line at a time in order to prolong the game as long as possible. However, from a purely points based perspective, this is a poor strategy. Setting up and executing a single 4 line clear (or a Tetris) is worth 7.5 times as many points as clearing a single line four times. Because the speed component of Tetris will eventually force any game to end, most high level players adopt a strategy that emphasizes making 4 line clears early and often.

Tetris is a complex, dynamic task in that the task state is constantly changing independent of any action taken by the player. Pieces will fall even if the player presses no buttons. In this kind of environment, taking no action requires a decision to do nothing, and the series of decisions made by the player at each zoid placement result in the final game score. Performance in Tetris is judged by this final game score, but because of the constant and varied game state, it is difficult to know what contributes to that performance.

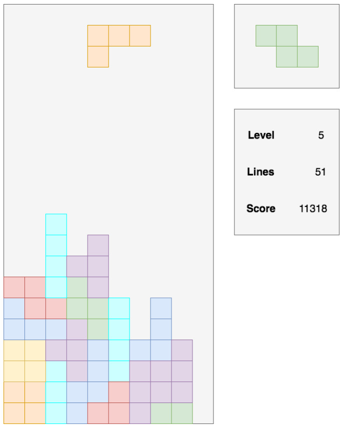


Figure 1: A Tetris game in progress. The active piece, the orange "L" is currently being placed by the player on the main game screen. The player also has access to score information, in the lower right-hand box, and one upcoming piece, the green "Z" in the upper right-hand box.

Tetris Models

Human play of Tetris is a test of human limits in dynamic decision-making and action and provides an excellent example of predictive processing (Clark,2013;Engstrom et al.,2018;Rao & Ballard,1999). Deciding where best to place a zoid becomes increasingly time-limited as the rate of fall increases. Likewise, the time available for the player to move the zoid to the chosen location also decreases.

Despite the complexity of human behavior in a task like Tetris, it is possible to build simple models capable of high level performance. Most of these come from the machine learning community, where Tetris is a popular test case for feature search algorithms.

These models function by defining a set of board features (selected by the researcher) that are believed to be important when making placement decisions. An early and commonly used set of features, defined by Dellacherie (Fahey,2015), is provided in Table 1. These are the features that we use to build the models used in this study.[2]

The models play Tetris by assigning each feature a numerical weight, the magnitude and sign of the weight indicates how desirable or undesirable a particular feature is. For a given move placement, the model generates all possible zoid positions and evaluates each one by multiplying the weight of each feature against the value produced by that move. These feature scores are added together to form a total move score, and the model ultimately selects and executes the placement with the highest move score.

The feature weights remain constant during a game, so the challenge of building a high performing model lies in choos-

[2] See Sibert et al. (2017) and Sibert and Gray (2018) for a fuller story.

Table 1: Tetris features proposed by Dellacherie, and used to construct the models used in this paper

Feature	Description
Landing Height	Height where the last zoid is added
Eroded Cells	# of cells of the current zoid eliminated due to line clears
Row Transitions	# of full to empty or empty to full horizontal transitions between cells on the board
Column Transitions	# of full to empty or empty to full vertical transitions between cells on the board
Pits	# of empty cells covered by at least one full cell
Wells	a series of empty cells in a column such that the cells to the left and right are both full

ing an optimal set of weights from a large search space. We employ the Cross-Entropy Reinforcement Learning method proposed by Szita and Lorincz (2006) and modified by Thiery and Scherrer (2009a,2009b).

Making Models More Human-Like

While traditional machine learning models are capable of high level performance, several important changes are made to the task environment that encourages models to adopt unhuman-like strategies in order to do well.

First, models tend to be unconstrained by the time pressure that is a major component of human gameplay. Second, models are reinforced for line clearing behavior, which encourages a strategy that primarily clears single lines. This is a viable strategy in the very long term (as, for example, used in Sibert and Gray, 2018), but only yields mediocre performance during the restricted time scale of a human game. Third, humans have access to additional information, like the upcoming zoid, that is not incorporated into the model decision making process.

Efforts have been made to explore how these environmental factors impact behavior. When trained on games of restricted length, models reinforced for line clearing behavior performed at a low-scoring but stable score level, while models reinforced for score reached higher scores but not as consistently. At their best, the score-reinforced models performed at the level of high performing student players, while line-reinforced models performed closer to intermediate level student players (Sibert et al.,2017).

This behavioral and strategy split was also observed in the absence of a reinforcement criteria when comparing models trained on restricted games against models trained on games of unrestricted length. The best long-game models far outperformed the short-game models by clearing single lines far beyond the point that the human game becomes unplayable. When restricted to human-length games, models adopted the higher scoring strategy of executing multiple line clears (worth far more points than a series of single line clears) early and often (Sibert & Gray,2018).

Whereas these prior studies focused on addressing the time pressure and reinforcement criteria aspects of the human Tetris environment, the current study aims to look at a third major difference between models and humans: humans have access to upcoming zoid information that models lack. Initial eye-tracking explorations (e.g., (Gray, Hope, Lindstedt, & Sangster,2015)) into human behavior show increased fixations on the next zoid box for higher level players, suggesting that this information is an important aspect of advanced play. Prior studies adjusting the model game environment led to performance levels, but only to the level of advanced human players (when equating for game length), suggesting that aspects of truly expert gameplay are still beyond the models. We hypothesized that allowing the models to consider the upcoming zoid when making placement decisions would result in higher performance. This ability to do *Two-Piece lookaheads*, thereby optimizing placements for 2 zoids rather than just 1, should also promote an increase in multiple line clears, as the models will have an increased capacity to plan ahead.

Methods

Model Development

Using the Dellacherie feature set (described in Table 1) and the cross-entropy reinforcement learning (CERL) method, we developed two models, a *One-Piece Lookahead* model and a *Two-Piece Lookahead* model.

Both models were trained on short games (a maximum of 525 zoids[3]) and were reinforced for high score. Both of these environmental conditions have encouraged more human-like behavior in our previous modeling studies. The models were developed using the same iterative CERL method (described in more detail in (Sibert et al.,2017), which can be summarized as a process that generates a set of candidate models with each model playing a single game of Tetris. The highest performing models are averaged together to create the starting point for generating the next set of candidate models. At each generation, 100 candidate models are tested, and the 10 best models were used to create the averaged model. In previous studies, this process was repeated 80 times, but here we implemented a halting condition: when the variance of the feature weights in the top performing models reached an acceptable threshold (below 0.01), the model was considered to have reached conversion and the search ended. Models tended to converge between 30 and 40 generations, greatly reducing the search time required for development.

The critical difference between the models was the amount

[3] Note that although these games are short for Machine Models, for the 300+ humans who have played an hour or more of Tetris in our laboratory, 525 zoids is the most zoids ever played by any human.

of lookahead information incorporated into the decision-making process. The *One Piece* models only information about one zoid at a time, and have no knowledge about what might be coming next in the sequence. It generates all possible placements for that zoid, and each placement is given a score by combining feature weights with the value of those features that result from the placement (i.e., if the placement creates a new pit, the score for that placement will change by the weight of the pit feature, and so on). At each placement, the model selects and executes the highest scoring move.

Two Piece models, by contrast, have access to the active zoid as well as the next zoid. Rather than calculate a score for each zoid placement, the Two Piece model evaluates the score for each pair of moves (adding together the score for the first and second move). This might cause the model to choose the second or third best move for the first zoid in order to allow a much higher scoring placement for the second zoid. Adding this capability greatly increases training time, not just in the greater computation time required for each game, but also by increasing the number of generations for convergence from approximately 30, for the One Piece models, to over 50 for the Two Piece models. However, we expected that this initial training cost would be compensated by better model performance.

Model Testing

Both models were tested using performance metrics (measured by game score) and behavior metrics (measured by types of line clears executed). Though only two models were developed, we had a total of four testing conditions. Because Lookahead was an environmental condition, it could be turned on or off for a developed model during testing. All tests were conducted on both models in both conditions: One Piece model with One Piece tests (same as training), One Piece model with Two Piece tests (alternate test condition), Two Piece model with One Piece tests (alternate test condition), and Two Piece model with Two Piece tests (same as training).

For the performance test, models were run through ten Tetris games. The zoid sequences of these games were generated using one set of ten random seeds (111, 222, 333, and on to 101010) to ensure that the models were tested in a controlled and equal environment.[4] Each model plays through this set of 10 games twice, once with only the current zoid (One Piece lookahead), and once with the current and next zoid (Two Piece lookahead).

Model performance was measured in three ways: the high score, the mean score, and the criterion score. The high score is the best score achieved on any game, and the mean is the average score of all ten games. The criterion score is a metric developed for evaluating human player skill (Lindstedt & Gray,2019), and is calculated by averaging the scores of the top four games in a testing period (for human players, this testing period is one hour, for models it is the set of ten games). The criterion score reduces the influence of a single unusually high or unusually low score on the overall measurement of player skill.

Model behavior was evaluated using the same ten test games, but rather than looking at a numerical score, the models were measured by the proportion of line clear types made during the game. Of all lines cleared during a game, some percentage are cleared using single line clears, some by two line clears, three line clears, and four line clears. The pattern of line clear types is a good measure of how the model behaves, as truly machine models tend to clear predominantly single lines, and high level humans try to emphasize 4 line clears.

Results

Table 2 shows the performance results for models trained in the One-Piece condition, tested in both the One-Piece and Two-Piece conditions. All scores were higher during testing with One-Piece lookahead (the same as the training condition), though the scaling scoring system of Tetris makes the score differences look larger than the actual performance differences that they reflect (line clears of all types are worth more points when executed at higher levels, meaning the rate of score accumulation increases as the game progresses).

Comparing the "native" training positions in the one-piece model (left column in Table 2) versus the native training position of the two-piece model (right column in Table 3) shows that the two-piece model performs worse than the one-piece model. Perhaps more surprising is the massive drop in performance when the Two Piece model is tested in the One Piece condition (left column in Table 3). These extremely low scores (left column in Table 3) represent very few line clears and in several games, these Two Piece models made no points at all.

Table 2: One-piece lookahead models tested in either the one or two piece lookahead condition

Testing Condition	One Piece	Two Piece
High Score	406000	200560
Mean Score	203766	161472
Criterion Score	323740	187965

Table 3: Two-piece lookahead models tested in either the one and two piece lookahead condition

Testing Condition	One Piece	Two Piece
High Score	1600	326180
Mean Score	220	132818
Criterion Score	540	229565

[4] See the discussion in Sibert & Gray, 2018, of the surprising differences in the variability of model performance across different random seeds.

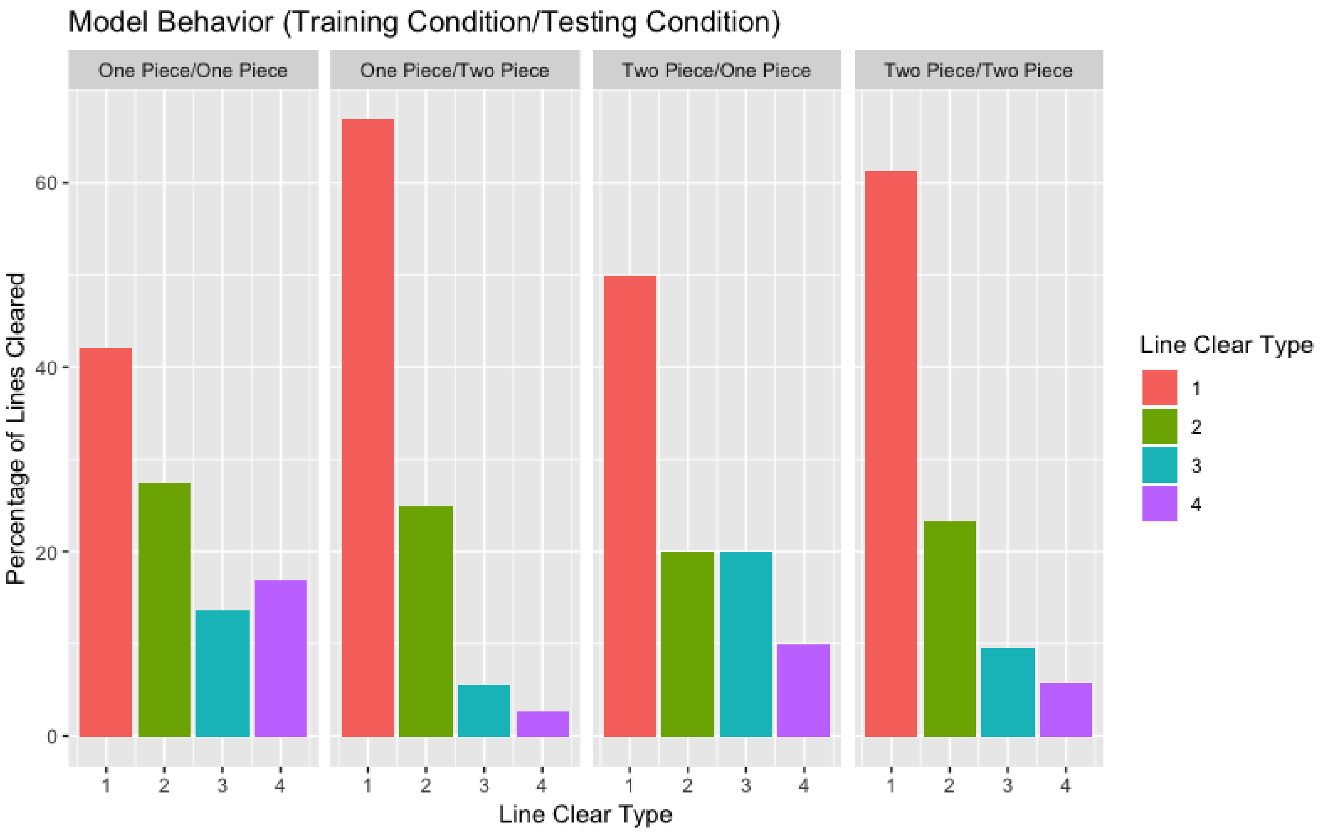


Figure 2: The behavior of models as represented by the proportion of each type of line clear made. Each set of bars represents a training/testing condition pair.

Figure 2 shows the percentage of each type of line clear averaged through the ten test games. The percentage of line clear type indicates the proportion of lines cleared using each type of clear to the total lines cleared during a game. Typical machine performance is characterized by a very high percentage of single line clears, and steadily lower percentages of each type of multiple line clear. High level human players have a more U-shaped pattern, with the highest percentage of lines being from 4 line clears, followed by 1 line clears and two line clears, with the fewest lines from 3 line clears.

The behavior pattern produced by the One Piece/One Piece condition (one-piece lookahead model playing one-piece lookahead games) is not quite the same shape as human experts, but represents a significant behavior shift toward human-like behavior. The behavior pattern has a U-shape that is similar to good human players, with more 4-line clears than 3-line clears.

Both models trained in the Two Piece condition show the much more typical machine pattern, with high percentages of single line clears, and progressively lower percentages of higher order line clears. The results from the Two Piece/One Piece model are representative of significantly fewer lines cleared, and are not as robust as the results from the other conditions.

The most unexpected result comes when the One Piece model is tested in the Two Piece condition. This resulted in lower performance, but also in a significant behavioral shift away from a humanlike pattern and toward the machine pattern.

Discussion

We expected that providing models with more information would improve model performance, and encourage a behavior pattern with higher levels of long term planning. Instead, we found that more zoid information led to lower model performance, and less human-like behavior. While there were not huge differences in performance between the one piece model tested with one piece lookahead and the two piece model tested with two piece lookahead (compare Tables 2 and 3), there were significant performance drops when a one piece model was tested with two piece lookahead. Removing two piece lookahead from the two piece model led to an even larger drop in performance, where the models were barely able to clear any lines.

The changes in the patterns of model behavior were also unexpected. Successful human players display a distinct pattern of line clear types, prioritizing four line clears. Single line clears are the next most frequent, followed by double line clears, and very low frequencies of three line clears. Many of our previous modeling efforts have tried to encourage models to follow similar patterns. During these experiments (Sibert & Gray,2018), we found that the model's behavior was determined by the training condition, and the pattern of line clears would persist in alternate testing conditions.

Adding two-piece lookahead to a high performing one-piece model (see the second set of bars in Figure 2) caused a large shift in model behavior, changing from the u-shaped pattern similar to high level humans to a sloped pattern consisting of primarily single line clears and very few four line clears. Both two-piece models displayed similar behavior patterns, but because the two-piece model tested using one-piece lookahead cleared almost no lines, few conclusions can be drawn from its pattern of line clear types.

Looking at the episode-level behavior of the models, we believe that the drop in performance and change in behavior patterns is caused by the model constantly making a suboptimal decision about the current zoid placement in service to a better placement for the upcoming zoid. However, once the upcoming zoid becomes the current zoid, there is a new upcoming zoid that may change the best placement. That is, the model is always planning to make a better move, but rarely follows through. Though we thought having additional zoid information would lead to the model making better moves, the short term optimization at the level of one or two pieces came at the cost of the generalization offered by the one piece model.

Based on these models, we can guess that if humans are incorporating upcoming zoid information into their placement decisions, it is not by making choices to facilitate specific placements for the next zoid. We do have some evidence (Gray et al.,2015) that players, particularly expert players, frequently fixate the next zoid box as they play, strongly suggesting that this information is being used in some way.

We theorize that a Tetris placement involves two stages: the decision phase and the movement phase. At low speeds, movement can be initiated before a final decision is made, but at high levels, speed is the limiting factor in performance, and placement decisions must be extremely rapid in order to maximize the available movement time. Rather than make decisions about the current zoid when it appears on the game board, we now interpret our model results as suggesting that expert players offload the decision phase to the previous episode, making a decision about the zoid placement while the zoid is still in the Next box. Once the piece appears on the Board, the player can initiate the movement phase for that zoid (now the "current" zoid in our terminology) while simultaneously initiating the decision phase for the upcoming zoid (i.e., the one that is now in the Preview box). Hence, expert players do not try to optimize two-piece placements but do try to optimize one-piece placements. The extra time for making these one-piece optimization is especially important at higher levels of Tetris; whereas maximum drop time is 16s at level 0, that decreases to 2s by level 9, to 1s by level 16, and to 0.67s at level 19. This explanation is compatible with component 2 of Lindstedt & Gray's (2019) Principal Component Analysis which suggested that better players make their placement decisions prior to moving the zoid.

We have not yet formally tested this hypothesis, but some expert Tetris players have already performed an informal experiment on their own. At 2018's Classic Tetris World Championship, 16 players engaged in a novel, ”no-next box” tournament which began play at level 18 (where it takes 1s for a zoid to drop from top-to-bottom). Although most of these players had secured a slot in the next day's playoffs for the Classic Tetris tournament, only one player scored over 30,000 points in this no-next box match with a few players scoring no points at all. The behavior of the players, usually characterized by high percentages of four line clears, was almost entirely single line clears. No four line clears were executed during the entire no-next box tournament.

Overall, the results of these models suggest that in complex, dynamic tasks, where there is rarely a single objectively correct action, the most successful behavior pattern must be general. Adding additional information serves to make model behavior more specific, which may be more optimal for a single decision point, but will be less successful over a long series of decisions. Additional zoid information, then, is likely not used to modulate individual zoid placement decisions. Instead, observation of expert players suggests that it is used to shift the time demands of a placement decision and allow more time to execute movements, making gameplay possible at very high levels. Further experiments may be able to explore how upcoming zoid information is incorporated by high level players, but the more machine-like approach of systematically exploring all options is clearly not the answer.

Acknowledgments

The work was supported, in part, by grant N000141712943 to Wayne D. Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Belchior, P., Marsiske, M., Sisco, S. M., Yam, A., Bavelier, D., Ball, K., et al. (2013). Video game training to improve selective visual attention in older adults. *Computers in Human Behavior*, *29*(4), 1318 - 1324.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181-204.

Destefano, M., Lindstedt, J. K., & Gray, W. D. (2011). Use of complementary actions decreases with expertise. In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (p. 2709-2014). Austin, TX: Cognitive Science Society.

Engstrom, J., Bargman, J., Nilsson, D., Seppelt, B., Markkula, G., Piccinini, G. B., et al. (2018). Great expectations: a predictive processing account of automobile driving. *Theoretical Issues in Ergonomics Science*, *19*(2), 156-194.

Fahey, C. P. (2015, 09). *Tetris AI.* Available from http://www.colinfahey.com/tetris/ ([Online; accessed 2015-Jan-30])

Gabillon, V., Ghavamzadeh, M., & Scherrer, B. (2013). Approximate dynamic programming finally performs well in the game of Tetris. In *Advances in neural information processing systems* (Vol. 26, pp. 1754–1762). Available from http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/881.pdf

Gray, W. D. (2017). Games-XP: Action Games as Cognitive Science Paradigm. *Topics in Cognitive Science*, *9*(2), 289–307.

Gray, W. D., Hope, R. M., Lindstedt, J. K., & Sangster, M.-D. (2015). Gaze transitions as clues to expert-novice strategy differences in a dynamic video game. In *ETVIS 2015 - First Workshop on Eye Tracking and Visualization.* IEEE VIS.

Holmes, E. A., James, E. L., Coode-Bate, T., & Deeprose, C. (2009, 01). Can playing the computer game "Tetris" reduce the build-up of flashbacks for trauma? A proposal from cognitive science. *PLoS ONE*, *4*(1), e4153. Available from http://dx.doi.org/10.1371%2Fjournal.pone.0004153

Kirsh, D., & Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, *18*(4), 513–549. Available from http://doi.wiley.com/10.1207/s15516709cog1804{_}1

Lindstedt, J. K., & Gray, W. D. (2019). Distinguishing experts from novices by the mind's hand and mind's eye. *Cognitive Psychology*, *109*, 1 - 25. Available from http://www.sciencedirect.com/science/article/pii/S0010028518300756

Nash, P., & Shaffer, D. W. (2011). Mentor modeling: the internalization of modeled professional thinking in an epistemic game. *Journal of Computer Assisted Learning*, *27*(2), 173-189.

Proctor, M. D., Bauer, M., & Lucario, T. (2007). Helicopter flight training through serious aviation gaming. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *4*(3), 277-294. Available from http://dms.sagepub.com/content/4/3/277.abstract

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79-87.

Rapp, A., Cena, F., Gena, C., Marcengo, A., & Console, L. (2016). Using game mechanics for field evaluation of prototype social applications: a novel methodology. *Behaviour & Information Technology*, *35*(3), 184-195. Available from http://dx.doi.org/10.1080/0144929X.2015.1046931

Sibert, C., & Gray, W. D. (2018). The Tortoise and the Hare: Understanding the influence of sequence length and variability on decision making in skilled performance. *Computational Brain & Behavior*, *1*(3-4), 215–227.

Sibert, C., Gray, W. D., & Lindstedt, J. K. (2017). Interrogating feature learning models to discover insights into the development of human expertise in a real-time, dynamic decision-making task. *Topics in Cognitive Science*, *9*, 1-21. Available from http://dx.doi.org/10.1111/tops.12225

Szita, I., & Lorincz, A. (2006). Learning Tetris using the noisy cross-entropy method. *Neural Computation*, *18*(12), 2936-2941.

Thiery, C., & Scherrer, B. (2009a). Building controllers for Tetris. *ICGA Journal*, *32*, 3–11. Available from http://hal.archives-ouvertes.fr/inria-00418954/

Thiery, C., & Scherrer, B. (2009b). Improvements on learning Tetris with cross-entropy. *ICGA Journal*, *32*(1), 23-33.

Cognitive-Level Salience for Explainable Artificial Intelligence

Sterling Somers (sterling@sterlingsomers.com)
Konstantinos Mitsopoulos, Christian Lebiere (cmitsopoulos; cl; @cmu.edu)
Department of Psychology, Carnegie Mellon University,
Pittsburgh, PA 15213 USA

Robert Thomson (robert.thomson@westpoint.edu)
Army Cyber Institute, United States Military Academy
West Point, NY, 10996 USA

Abstract

We present a general-purpose method for determining the salience of features in action decisions of artificial intelligent agents. Our method does not rely on a specific implementation of an AI (e.g. deep-learning, symbolic AI). The method is also amenable to features at different levels of abstraction. We present three implementations of our salience technique: two directed at explainable artificial intelligence (deep reinforcement learning agents), and a third directed at risk assessment.

Keywords: computational model; salience; artificial intelligence; reinforcement learning;

Introduction

In recent years, Deep Reinforcement Learning (RL) has gained popularity for training agents engaged in activities as diverse as playing Atari games, strategic games such as Go and chess, and controlling robotic platforms. Despite recent success, there is, perhaps, a lack of trust in applying RL in real-world scenarios. The behavior of RL systems are often qualitatively different from human behavior and they are inherently difficult to understand. Unlike traditional symbolic AI systems, they are not easy to introspect upon. Furthermore, because RL agents are largely trained without human supervision, there is often little reason to expect them to produce abstractions similar to our own. This makes the task of mapping from human conceptual space to the RL agent's conceptual space a significant challenge.

Previous Work

The common input of RL agents is usually an image. Thus, the agents are comprised of convolutional layers that map pixels to actions and reward expectations. For this reason, most of the techniques that are used for saliency calculations in image classification (Grün, Rupprecht, Navab, & Tombari, 2016) can be used in a RL setting.

One of the first and most common methods for understanding Deep RL agents is to produce gradient-based saliency. Typically, this method uses the gradient of a prediction with respect to an input image to estimate the importance of pixels. In other words, how much the change of a pixel value affects the prediction value (Simonyan, Vedaldi, & Zisserman, 2013). Other popular methods are perturbation methods (Greydanus, Koul, Dodge, & Fern, 2017). Such methods rely on comparing the resulting prediction (or decision) between a modified input with the original one. This gives insights on the importance of individual image regions.

Cognitive Salience

Although the above methods provide a visual explanation of what the agent pays attention to, they often fail to do so consistently. Frequently, the resulting saliences are challenging for a human user to use in producing a meaningful interpretation of the agent's behavior. For this reason, we propose a method that operates on a more abstract level than pixels. More precisely, our cognitive approach involves modeling of the agent's behavior but assuming non-pixel features. Instead the features are entities that a human can comprehend in order to understand the underlying causes of the RL agent's particular decision. Furthermore, the modeling process is model-agnostic and can be used with any model or even humans.

Cognitive Model

We have chosen to develop the cognitive portion of our system in ACT-R. ACT-R is a computational theory of cognition that accounts for the information processes in the human mind (Anderson et al., 2004). The mechanisms in ACT-R are task-invariant and constrained by the limitations of the brain (see Anderson (2007) for an overview). ACT-R is a hybrid architecture, composed of both symbolic and sub-symbolic processing. The hybrid nature of ACT-R is particularly compelling for our work because the symbolic level is inherently explainable, while the sub-symbolic level has the potential representation required to interface with other sub-symbolic systems like neural networks. Furthermore, because we intend to use the output as an explanation for human users, we hope to rely on the constraints of the architecture to limit the output.

Information processing occurs in ACT-R primarily through the interaction of the production system and the declarative memory. Declarative memory is represented as chunks of information. Each chunk has an associated activation level that modulates its retrieval. Chunks are compared to the desired retrieval pattern using a partial matching mechanism that subtracts from the activation of a chunk its degree of mismatch to the desired pattern, additively for each component of the pattern and corresponding chunk value. Finally, noise is added to chunk activations to make retrieval probabilistic, governed by a Boltzmann distribution.

While the most active chunk is usually retrieved, a blending process (Lebiere, 1999) can also be applied that returns a derived output reflecting the similarity between the values of the content of all chunks, weighted by their retrieval probabilities reflecting their activations and partial-matching scores.

Blending and Salience

The ACT-R blending mechanism retrieves an estimate of values based upon the previous experiences stored in memory, and is computed with the following equation:

$$V = \underset{V_t}{\arg\min} \sum_{i=1}^{n} P_i \cdot Sim(V_t, v_{it})^2 \tag{1}$$

The value, V is, therefore an interpolated value based on matching chunks i, weighted by their retrieval probability P_i. $Sim(V_t, v_{it})$ is a similarity function used to compare memory chunks v_{it} and candidate consensus values V_t. In the simplest case where the values are numerical and the similarity function is linear, the process simplifies to a weighted average by the probability of retrieval $V_t = \sum_{i=1}^{n} P_i \cdot v_{it}$.

We consider 'salience' to be the influence a factor has on a decision. The greater degree of influence, the more salient it was when the decision was being made. We model the decisions of an agent by tracing its action decisions and populating a memory. The resulting memory is used in a similar manner to instance-based learning theory (Gonzalez, Lerch, & Lebiere, 2003), except our intent is not to learn to maximize reward, but rather to mimic the behavior of an agent.

We calculate salience by taking the derivative of the blending equation (1) with respect to each feature:

$$S(D_t, f_k) = \\ c \sum_{i=1}^{n} P_i \cdot \left(\frac{\partial Sim(f_k, v_{i,k})}{\partial f_k} - \sum_{j=1}^{n} P_j \cdot \frac{\partial Sim(f_k, v_{j,k})}{\partial f_k} \right) \tag{2}$$

with $P_i = softmax(M_i/\tau)$, $c = \frac{MP}{\tau}$ and $M_i = A_i + \sum_{k=1}^{l} MP \cdot Sim(f_k, v_{i,k})$. This is a novel extension of the blending mechanism that exploits its analytical tractability to provide a closed form of the gradient-based salience of its representational features on its decisions.

Deep Reinforcement Learning Agent

In this work we are not interested in solving completely the problem that the agent is facing. Instead, a basic Deep RL architecture that receives relevantly a high score, in the domains used here, will suit our purposes. For this reason, we utilize the Advantage Actor Critic (A2C) algorithm which is the synchronous version of the A3C (Mnih et al., 2016). We adopt the same architecture and implementation details as in Vinyals et al. (2017) but removed the spatial policy for the drone domain. The agent as it is common in this setting attempts to maximize the expected return by interacting with the domain.

Methodology

The process of mapping between an agent (RL or programmatic) and the cognitive model is common in all the cases described below. Once we have an agent implemented (trained or programmed), we gather data of its performance using terms from a human ontology. In the RL cases, we create a symbolic observation for each step, and record a symbolic interpretation of the action chosen by the network. In the risk-assessment case, we gather symbolic data about the situation, and symbolic data describing the outcome. In each of these cases, this data is gathered to comprise a "memory" for the model. We treat those memories as the knowledge the model has about its respective agent. That knowledge is used by the model (through blending) to estimate what the agent will do in a new (possibly unseen) situation. The derivative of that process, as described, provides the salience. We, thereby, attempt to communicate *why* the agent chooses the action it does by allowing the user to build a mental model of what feature(s) the agent considers most important in different scenarios.

StarCraft II

StarCraft II (SC2) is a real-time strategy game in which players (human or AI) control the production and placement of buildings and the production, movement, and interaction of of militaristic units, in order to defeat opponents. An API and sample RL missions in SC2 are presented in Vinyals et al. (2017). SC2 also supports smaller, constrained missions in which points can be designated for certain achievements. These mini games are useful in the RL domain because they provide smaller tasks and straightforward rewards to be exploited by reward functions.

We used the go-to-beacon mini game presented by Somers, Mitsopoulos, Lebiere, and Thomson (2018). The objective of the go-to-beacon mini game is to move a unit to one of two beacons: a low-value green beacon or a high-value orange beacon. The beacons can be presented either individually or in pairs. When presented in pairs, the optimal solution is to prefer the orange beacon over the green beacon. Interaction in this scenario requires the selection of a unit and then a mouse click on the game map or mini map in the region of the chosen beacon. The unit will then proceed, over time, to move to the location of the mouse click. As soon the agent arrives at one of the beacons, the score is increased by the value associated with that beacon, the beacons disappears, and new beacons are generated at random with four possible configurations: 1) a sole green beacon (green-only scenario), 2) a sole orange beacon (orange-only scenario), 3) both an orange and green beacon presented in such a manner that the unit could take a direct path to the orange beacon without stepping on the green beacon (non-blocking scenario), and 4) both the orange and green beacons presented in such a manner that the green beacon is overlapping the direct path between the unit and the orange beacon (blocking scenario).

The scenarios are grouped into the four categories to accommodate a human-level ontology that might help understand the RL's behavior. Despite the simplicity of the mission, our RL agent learned a sub-optimal policy. In particular, the RL fails to guide the SC2 unit around the green beacon to the orange beacon, failing to distinguish the blocking and non-blocking scenarios. In the context of this work, we did not attempt to resolve this issue. Our aim in this paper is to explain why the RL agent acts the way it does, and to cast these explanations using a human-level ontology.

SC2 Explanation The RL agent did not learn how to go around the green beacon to reach the orange beacon. Our approach aims to explain the behavior of the RL agent in terms of its internal states: what it perceives, what it knows, and what actions it takes in response. We assume that actions consistent with a going-around action require a spatial inference: that the green beacon is 'between' the agent and the goal. In the SC2 case, we aim to communicate to a user that the network has not acquired this concept and therefore fails to act as expected.

In this particular task we are curious whether the agent has an internal representation that is functionally consistent with: a) a sole green beacon, b) a sole orange beacon, c) a non-blocking scenario, and d) a blocking scenario. We collected data of the RL agent by tracing its behavior while it played the game, collecting chunk representations that described the scenario and the action chosen in that scenario. The chunks had the following structure: *green*:value, *orange*:value, *blocking*:value, *internal representation*:vector, *select-green*:value, *select-orange*:value, *select-around*:value. Each value, aside from the vector value, are binary 0 or 1. The *green, orange, blocking* together describe the scenarios (a-d) and *select-green, select-orange, select-around* are high-level descriptions of possible actions taken. Note that these are representations of ground-truth, not representations created by the agent. We are attempting to assess when (and if) the RL agent is behaving in a manner consistent with these representations. We adopted the approach used by Somers et al. (2018) and used a vector representation to capture the internal state of the agent. The internal representation is used in the partial-match portion of the blending process. Just as in Somers et al. (2018), we use cosine as a similarity measure between two vectors. Overall, we filter the data we collected to include at least one example of each scenario and to maximize the distance between vectors. For the purposes of the present description, experience was filtered down to 20 examples.

We made a prototype of a display that could be used for explanation. It outputs the results of the blending process and salience calculation graphically, relying on the user to create the proper inferences. At each step in the game, the cognitive model makes a blend estimate of the action to be chosen by the agent, and calculates the salience of the high-level features: green, orange, blocking. A screen-capture of this display is presented in Figure 1. A screen capture of the corresponding SC2 scenario is presented in Figure 2.

The top graph in Figure 1 displays the three possible action choices a human might expect the RL agent to make and the cognitive architecture's estimate of which action the RL agent will choose. The example in the scenario is a blocking scenario and, as the display correctly indicates, the RL agent will choose to select the orange beacon.

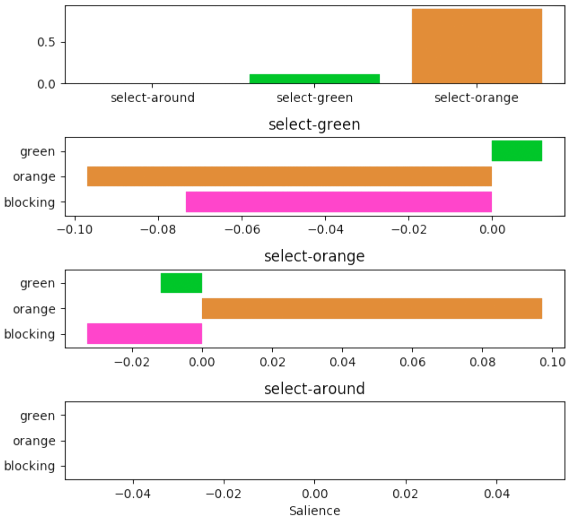


Figure 1: Explanation display. Top panel illustrates the blend value for the action decision. The three remaining panels display the associated salience for their respective decisions.

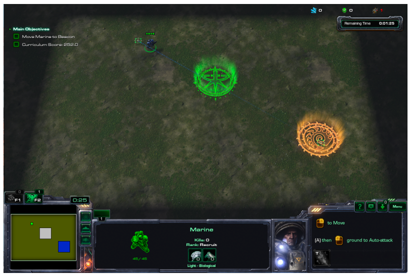


Figure 2: Screen capture of blocking scenario in StarCraft II. In this image, the marine (controlled by the reinforcement learner) is attempting to get to orange beacon.

The three bottom graphs in Figure 1 display the salience for each decision available. The top displays the salience for the select-green decision, the middle displays the salience for the select-orange decisions, and the bottom graph displays the salience for the select-around decision. As described above, the salience indicates the degree of influence each of the features (green, orange, block) have on the action decision. Given that the dominant action chosen by the drone is select-orange in the blocking scenario, discussion will focus on describing the bottom-middle graph, ”select-orange”. The graph indicates that the presence of the green beacon has a small negative salience in the action decision and the abstract concept ”blocking” has a larger negative salience, with the dominant influence being the presence of the orange beacon. This makes sense given that the RL agent always tries to get to the highest reward but appears to be unaware of the fact that the green beacon is blocking the orange. The salience and the associated lack of 'go-around' action communicate that the agent has not formed the abstract concept, 'blocking'.

Drone Domain

A second domain that we have applied the cognitive salience technique to is a drone operations domain. Currently, our drone environment is a 3D gridworld abstraction of MAVSim (Youngblood, Kravacic, & Le, 2018). Once trained, the RL agent can be deployed in MAVSim.

The aircraft we are simulating are fixed-wing drones. The missions generally include search and provisioning lost hiker(s) with any combination of food, water, first-aid, and communication devices. The rules of the environment are an abstraction that capture the constraints of flight dynamics. Although, at our current level of simulation, the specifics of the aircraft are not captured, the rules of the environment are sensitive to flight constraints more generally. The rules of the environment constrain the turning radius, maintains forward motion, and restricts elevation changes, just to name a few. Sensors on the aircraft are sensitive to altitude. Package survival (once dropped) depends on underlying terrain and will fall differently when dropped at different altitudes.

We have trained an RL agent to navigate to a hiker visible on a topological map and to drop package(s) near the hiker. Currently, programmatic solutions are used to carry out other segments of the flight including: loading the packages, searching for the hiker, and returning to the airport. Our explainable AI challenges in this domain are many and we have only begun to touch the explainability potential in the drone domain. We present here two example prototype uses of our salience technique within the drone domain: risk assessment and egocentric salience.

Risk Assessment As we move from low to high levels of fidelity, there is an increasing number of moving parts to deal with the increasing level of detail and, as a result, a large potential decision space to explain. Anticipating our needs, we have begun to prototype different explanation interfaces for different aspects of a mission. In this simplified example, we imagine the provision-loading process, and when it might need explanation. The most obvious situation where we might want some form of explanation is when there is a failure. After a large number of simulations, for example, we may want some form of risk analysis that is sensitive to the particular constraints of any given example case. We use the following simplified example, where the ground truth about a package loading module is described. In this example, the operator is unaware of the ground truth and is attempting to diagnose a problem with the package loading module.

The package loading module is given as input the needs of the hiker(s). This module has a number of flaws that we know a priori for this example but are not known by the user: 1) it always loads communications equipment, regardless of needs. 2) It always loads food, regardless of needs. 3) It never loads water, regardless of needs. We characterize success and failure in the following manner: A) if the hiker needs a provision and does not get it, there is a failure. B) If the hiker does not need a provision, and a provision is loaded, this is also considered a failure. We chose to consider this a failure because the aircraft we are modeling has limited space for provisions. Loading unnecessary provisions can prevent loading required provisions and lead to increased fuel consumption, which could lead to the drone making multiple, unnecessary trips, putting the hiker(s) in unnecessary risk. To remove any confusion, we also include a third success/failure condition: C) even though we know, a priori, what is wrong with the drone, we remember that, for the purposes of the example, when the hiker does not need a provision (in this example, water) and that provision is not loaded (because there is a flaw preventing it), we consider that a success even though, underlying that success, is a failure in the mechanism. In other words, the salience mechanism in this case only has access to the behavior of the drone, without any knowledge of its internals. These are the rules that describe the success/failure conditions in this example.

Following those rules we generated data to simulate the erroneous module described. The data includes all sixteen possible binary combinations of package-loading requirements: food needed/not needed, water needed/not needed, first-aid needed/not needed, communications needed/not needed; as well as traces of their success and failure: food-success yes/no, water-success yes/no, first-aid-success yes/no, communications-success yes/no. This is represented in ACT-R as chunks with 8 slot/value pairs.

Once we have data, we can probe the module with a new case to perform a risk assessment in a specific situation. For example: *Food: needed; Water: not needed; First-Aid: needed; Communications: not needed*

There are two aspects to the output. First the blend provides an estimate of the values (randomly generated example presented in Table 1). The values are intuitively what we might expect (given the module described above): a value of 1 for both radio and food (which is always loaded by the erroneous module). Water is estimated to be zero (which deceptively makes sense, since it was not requested). Finally, First-Aid is estimated to be 0.68, which could be rounded to 1 (given our binary example). These results are consistent with the rules described above.

Table 1: Blend Estimate of Random Example

Provision	Requested	Estimate
Food	1	1
Water	0	0
First-Aid	1	0.68
Communications	1	1

The salience provides further, useful information that could potentially be used to diagnose a faulty module. Since the blend is produced for each output factor (food, water, first-aid, communications), we generate a set of saliences for each. The salience derivative is computed with respect to each feature, so each factor has 4 saliences values associated with it. The salience values for the Food and Communications are identical but their values are so small (10^{-8}) that we do not display them. The small values are important in the explanation, however, because they indicate the blend is not strongly influenced. This makes sense, given that we know that the agent always loads food and always loads communications.

The salience of Water is zero for all features and, therefore, not displayed. This is particularly telling, a very strong suggestion that the loading of water is not sensitive to any factor. This makes sense given the ground truth about the module (always fails to load water).

Finally, the salience of First-Aid is presented in Figure 3. This display is telling, indicating salience values near zero for Water, Food, and Communications (Radio).

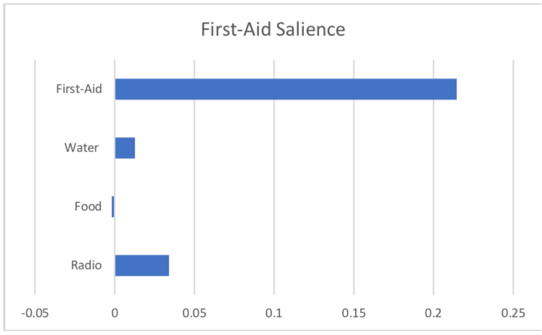


Figure 3: Display of salience for assessment of First Aid provisions.

Egocentric Salience While the risk assessment is derived from synthetic data generated by a programmatic agent, our final example, which we term, ‘egocentric salience’, results from a model trace of an RL drone agent. The input to the RL is an egocentric image and an allocentric image. Example inputs are illustrated in Figure 4 and the bottom display of Figure 5. The first image is a 20 unit by 20 unit allocentric map of our environment. The different green and brown colors represent different kinds of vegetation. The purple arrow, pointing downwards, represents the drone; with the direction of the arrow indicating the drone’s heading. The drone’s color indicates its altitude. The red cross indicates the location of the hiker for which the drone is attempting to drop off a package.

The second image (bottom of Figure 5) is a 5 unit by 5 unit egocentric view that corresponds to the first image. This image is a vertical slice of the first, in the region around the drone’s action space. At any step the drone can head directly forward, diagonally forward, or turn 90 degrees in either direction. Furthermore, any of those directions can be combined with a single change of altitude. With the options of left, diagonal left, center, diagonal right, and right, multiplied by: no altitude change, an increase in altitude, or a decrease in altitude; the drone has an action space of 15 possible grid square in a 3D gridworld. A sixteenth action to drop a package is also available. The movement actions are entirely captured by the five by five units of the egocentric input image.

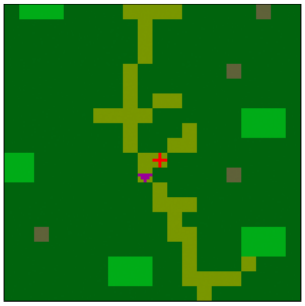


Figure 4: 20x20 unit allocentric view. Greens and browns indicate different forms of vegetation (largely trees and grass). The purple triangle indicates the drone (facing the bottom of the image). The red cross indicates the location of the hiker.

The drone is super-imposed on this picture and provides the drone an explicit representation of its altitude. The column on the left-most of the egocentric view corresponds with the grid-square directly to the right of the drone (from the perspective of Figure 4) but corresponds to the square immediately to the left of the drone (from the perspective of the drone). This is the case because the drone is actually facing downward in the allocentric view. The column second from the left in the egocentric view corresponds to the patch of grass to the diagonal left of the drone (down and to the right in the allocentric image). The three remaining columns represent obstacles (trees) surrounding the drone centre and all the way to the right. If the drone were to fly left, forward, diagonal right, or right, it would crash into a tree.

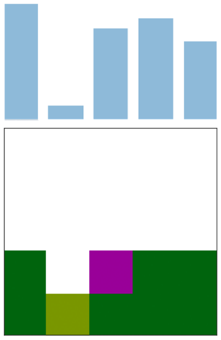


Figure 5: Top frame: the bars above the egocentric view indicate the salience of the columns in the egocentric view. Bottom frame: 5x5 unit, egocentric input to the network. This image corresponds to the allocentric image and changes each step of the simulation.

The drone exhibits a reasonable capacity to carry out the two segments of the mission it has been trained on (traversing the map and dropping the package). However, despite its success, the drone does exhibit unusual behavior: sometimes taking a bizarre path to the hiker, or circling the hiker many times before dropping a package. This successful yet unusual behavior is a good candidate for explanation because a participant viewing the mission might want to resolve why the behavior is markedly different from what a human might do. Explaining this type of behavior can furnish trust in the system if the human understands (and accepts) the reasoning.

Figure 5 depicts a prototype explanation for the scenarios presented in Figure 4. Unlike the other examples, the concepts we are associating salience with in this example are primarily spatial. The idea behind egocentric salience, is still, however, quite abstract. The data is collected under the assumption that the RL drone responds to features at altitude, trying to avoid crashing, for example, regardless of the specific identity of the object. The aim of the explanation is towards understanding how the drone responds more generally to its environment. It would be used, for example, to allow a user to rewind to a point where they thought the drone started to behave oddly, and get a sense of what it was 'attending' to considering in its decision.

The egocentric salience is a little more difficult to interpret. The bars above the egocentric view (Figure 5) are meant to communicate how salient the columns of that view are toward a single decision. This particular example could be interpreted as all dark green objects (obstacles), at altitude 1 (counting from the bottom, from zero), are highly salient, with the single safe column, with a low degree of salience. The low-degree of salience was an unexpected result but, with some interpretation, seems to make sense. The areas where objects are lower in altitude than the drone's current altitude generally will not influence the drone's behavior. Instead of thinking about the drone as going to a safe area, per se, the drone seems to be avoiding dangerous areas (the result of which is functionally the same, going to a safe area).

As with our other examples the blending process also makes an action estimate. Because the action space is so large (15 movement actions), we do not blend for each value. Instead we blend a single action value across the range. In this case, the estimate is 6.8. Rounded to 7 an action of forward and to the left, which would result in action following the path in the forest.

Discussion

The work presented here is targeted at explainable AI and is still in an early phase of development. Our goal builds upon (Kümmerer, Wallis, & Bethge, 2015) to eventually unify pixel-level, cognitive-level, and artificial intelligence-level salience computation in a model-agnostic framework. Although we have thus far concentrated on explaining AI, we are interested in exploring salience for human-generated data, specifically in the context of instance-based learning (IBL) as IBL models have been used in a wide variety of models including social dilemmas (Gonzalez, Ben-Asher, Martin, & Dutt, 2015) and two-person games (West & Lebiere, 2001).

References

Anderson, J. R. (2007). *How Can The Human Mind Occur In The Physical Universe?* New York, NY: Oxford University Press.

Anderson, J. R., Bothell, D. J., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004, oct). An integrated theory of the mind. *Psychological review*, *111*(4), 1036–60. doi: 10.1037/0033-295X.111.4.1036

Gonzalez, C., Ben-Asher, N., Martin, J. M., & Dutt, V. (2015). A cognitive model of dynamic cooperation with varied interdependency information. *Cognitive science*, *39*(3), 457–495.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635. doi: 10.1016/S0364-0213(03)00031-4

Greydanus, S., Koul, A., Dodge, J., & Fern, A. (2017). Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*.

Grün, F., Rupprecht, C., Navab, N., & Tombari, F. (2016). A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*.

Kümmerer, M., Wallis, T. S. A., & Bethge, M. (2015). Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, *112*(52), 16054–16059. Retrieved from https://www.pnas.org/content/112/52/16054 doi: 10.1073/pnas.1510393112

Lebiere, C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19. doi: 10.1007/s001970050071

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Somers, S., Mitsopoulos, C., Lebiere, C., & Thomson, R. (2018). Explaining decisions of a deep reinforcement learner with a cognitive architecture. In *Proceedings of the sixteenth annual conference on cognitive modeling* (pp. 144–149). Madison, WI: Lawrence Erlbaum Associates.

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., ... others (2017). Starcraft ii: a new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Journal of Cognitive Systems Research*, *1*(4), 221-239.

Youngblood, G. M., Kravacic, B., & Le, J. (2018). *Mavsim.* https://gitlab.com/COGLEProject/mavsim. GitLab.

Lightweight Schematic Explanations of Robot Navigation

Robert St. Amant (robert.a.stamant2.civ@mail.mil), MaryAnne Fields (mary.a.fields22.civ@mail.mil),
Brian Kaukeinen (brian.t.kaukeinen.civ@mail.mil), Christa Robison (christopher.j.robison5.civ@mail.mil)
U.S. Army Research Laboratory
Adelphi, MD, U.S.

Abstract

This paper describes a representation to support explanation of robot navigation based on image schemas, a set of abstractions closely tied to embodiment. The representation is intended to satisfy two criteria: an explanation must provide a logical or causal account of the phenomenon to be explained and be understandable by its audience. Evidence in the literature of cognitive linguistics and related fields suggests that image schemas satisfy the second criterion. We provide evidence for the first with an Answer Set Programming formalization of navigation-related image schemas. Schema-based explanation representations are generated for a robot navigating through a simple indoor environment.

Keywords: Image schema; robot navigation; explanation

Introduction

Symbolic representations in artificial agents (robots, in this paper) are often invaluable—they allow for the concise specification and communication of potentially complex behaviors. This pragmatic, engineering motivation sometimes leaves open the question of why one representation is better than another. Our interest is in a related issue that has gained increasing attention in recent years: the ability of a robot to explain its actions.

This paper describes a representation, based on *image schemas*, for the explanation of robot navigation. "Image schema" is a term coined by Lakoff (1987) and Johnson (1987): a structured, general pattern intended to capture experience "at the level of our bodily movements through space, our manipulations of objects, and our perceptual interactions." The contribution of this paper is to show how a schematic cognitive structure can be derived from patterns of navigation actions recorded by a physical robot. Image schemas appear to be a natural fit for explanation.

Navigation may at first seem to be a trivially easy domain, for human beings if not for robots, but researchers in human spatial cognition characterize navigation as being among the most complex of cognitive operations (Wiener, Büchner, & Hölscher,2009). For example, an ontology developed for urban transportation (Timpf,2002) contains concepts for path, start, goal, connection, transportation mode, map, sign, direction, distance, and time, plus names of specific entities (e.g., streets and subway stations). When we navigate, we draw on our memories and on external maps. We think about navigation at different levels of abstraction; we chain together navigation plans and paths; we combine navigation with other activities. Many navigation tasks may be easy to understand, but they can grow complex enough to require explanation of why a specific route is followed or avoided.

What counts as a good explanation? In the philosophy of science, the standard definition of an explanation has two parts (Hempel & Oppenheim,1948): a description of a phenomenon to be explained (the *explanandum*) and the explanatory account itself (the *explanans*). Both parts can be formulated as sentences in logic, with the explanandum being one sentence and the explanans being a set of sentences for which the explanandum is a consequence. The sentences in the explanans are of two types, general laws and antecedent conditions specifying applicability, and for a sound explanation they must be true. In other words, a scientific explanation captures the causes or logic of a phenomenon.

Lombrozo and Carey (2006, p. 169) broaden the definition above: the logical or causal process identified in an explanation "can be subsumed under some kind of pattern or causal schema" that is already understood by a questioner or introduced as part of the explanation itself. That is, a *psychological explanation* (which we will henceforth call simply an explanation) is a bridge between the explanans and the explanandum that has a foundation in a questioner's prior knowledge. An explanation should be understandable by the audience to whom it is directed (as well as the explainer). Further, explanations do not exist in a vacuum but are are typically part of a larger context, which includes the goals of the audience and the situation in which the explanation is offered (Mueller, Hoffman, Clancey, Emrey, & Klein,2019).

In the next section we briefly review the relevant literature on image schemas, as a partial account of spatial reasoning. Schematic structures alone do not satisfy the causal/logical requirement of explanations, however; for this we depend on a formalization in terms of Answer Set Programming (Gebser et al.,2011), which supports inferences to necessary components of an explanation in an abstract, symbolic representation. We then describe the conversion of sensor and control data on a physical robot into the abstract representation and walk through an example.

Related work

In the infant cognition literature, Mandler and Pagán Cánovas (2014, p. 519) outline a set of spatial primitives that act as image schemas: "By themselves or in combination they structure the conceptual representations that describe events."

These image schemas plausibly underlie infant spatial cognitive capabilities, gained in the first six to seven months after birth: PATH, START-PATH, END-PATH, PATH-TO; LINK; THING; ±CONTACT; CONTAINER, OPEN, INTO, OUT OF; LOCATION; ±MOVE, ANIMATE MOVE, BLOCKED MOVE; BEHIND; APPEAR, DISAPPEAR, EYES. It will be useful for our purposes to expand the set of spatial relationships represented by BEHIND, to include NEAR/FAR, LEFT/RIGHT, IN FRONT

OF/BEHIND. This is not a complete set of image schemas, even for navigation (Croft & Cruse,2004), but provides a reasonable starting point.

For clarity, a PATH is "the way to get [somewhere];" the most general version of the schema is commonly called SOURCE-PATH-GOAL (Lakoff,1987), with its components as given by its name. BLOCKED MOVE is sometimes referred to as BLOCKAGE (Cervel,1999), with components that include a path, a moving entity, and another entity acting as an obstacle. THING is an entity that can be perceived in space, which we will interpret here as a physical object. LINK is a general, contingent relationship that may come into being between objects or schemas.

Image schemas exist in a specialization hierarchy. For example, a path component of a SOURCE-PATH-GOAL might involve continuous motion, or it might consist of a sequence of discrete steps (each of which can be thought of as a PATH itself, with an atomic transition between locations). Image schemas can also be related by composition. The end of a PATH can be a LOCATION; a BLOCKAGE applies to a PATH.

Image schemas are implicitly associated with activities or events. We call these *characteristic operations*—informally, what an image schema is for. For example, a characteristic operation of the SOURCE-PATH-GOAL schema is for some agent to traverse it in a specific direction. One characteristic operation of a CONTAINER is that it can contain other objects, or that an agent can MOVE INTO or OUT OF one. (Some characteristic operations are schemas themselves, or incorporate schemas.) In a navigation context, regions are a type of CONTAINER: one can enter INTO or exit OUT OF a REGION. Buildings, rooms, and even deadends are CONTAINERs in the same way.

Image schemas have been adopted as conceptual primitives in many other fields aside from infant spatial reasoning. Geographic information systems are one example (Walton & Worboys,2009). In an extensive discussion, Frank and Raubal (1999, p. 67) observe that image schemas capture geographical concepts in a way that "comes close to how people use them in their everyday lives."

Another area of related work is AI planning. Robots are closely associated with planning, from the abstract level of classical planning down to the detailed level of planning paths. Our own past work used image schemas to capture patterns of behavior in planning agents (St. Amant et al.,2006). Navigation planning is typically handled as a search problem with a single operator for moving between locations, rather than distinct planning actions for reaching different states. Korpan and Epstein (2018)'s WHY-PLAN explains navigation plans, by examining differences between a robot's objective function and that of a human being, mapping their components to natural language phrases. For example, if a robot's objective function is sensitive to crowd density, it can contrast its solution plan with a human's: "This path is ⟨slightly⟩ ⟨less crowded⟩ than the alternative."

In an area closely related to explanation, Rosenthal et al. (2016) describe a system that generates narrative "verbalizations" to describe a navigation path. Rosenthal et al.'s navigation paths are representative of most such work we are familiar with: a path contains a goal location, a starting location, and an ordered list of intermediate waypoints (plus collinear subsequences of waypoints, to facilitate the identification of turns). Such representations map naturally onto the SOURCE-PATH-GOAL image schema. Landmarks, with appropriate semantic tags, and their spatial relationships to the robot can also be easily interpreted in schematic terms. None of this is surprising. Most of the other image schemas remain only implicit in navigation plan representations, however, though they could plausibly contribute to explanations. In the next section, we show how they can be made explicit.

Schemas for Navigation Paths

In this section we describe a formalization of image schemas to support explanation. Our target is a logical representation sufficient for an explanation, following the lead of other work in planning (Chakraborti, Sreedharan, Zhang, & Kambhampati,2017;Fox, Long, & Magazzeni,2017). Generation of the text and narrative of explanations, such as carried out by systems above, is part of the task that we leave for future work.

Answer Set Programming, with roots in knowledge representation and reasoning, has become a popular paradigm for declarative problem solving. ASP has shown promise in spatial and temporal reasoning (Li,2012) and commonsense reasoning (Balduccini,2009). Our implementation relies on the Potassco set of tools for ASP; rules are encoded using the input language of Gringo (Gebser et al.,2011).

A problem specification is separated into two parts: a specific problem instance, expressed as predicates; and an encoding, a general set of inference rules that apply to any problem instance. For explanation of a navigation path, a problem instance consists of locations and objects in environment, the initial location of the agent, the path it follows, and the commands it issues to follow the path. (We refer to an "agent" in this context as a reminder that this is a high-level abstraction of a robot—the agent is not even explicitly represented.) Time and space, in the form of a set of locations, are discretized; a time limit M for execution of the planned path (moves(M)) is also provided.

Locations are named by unique constants, e.g., loc(x_1). A path has a source location and a goal location, path(x_0,x_f), and a sequence of waypoints to be traversed in order. These are expressed as steps with paired locations; if P is a path, then step(P,x_0,x_1), step(P,x_1,x_2), ..., step(P,x_{f-1},x_f). The agent begins at(x_0,1), where 1 is the starting time. The environment may also include landmarks, obstacles, or demarcated regions that occupy specific locations, e.g. land(l_1,x_6), obst(o_1,x_3). Landmark locations are disjoint from path locations; obstacles have a non-empty intersection with path locations; regions have neither restriction.

An ASP problem encoding is further divided into separate parts. In the generation part we can specify candidate solutions, actions taken by the agent. The agent may move from one location to another along the path by taking a step, a move action; moves are possible at any time T = 1..M, but no more than one move can be carried out at a given time.

```
{ move(X_i,X_j,T) :
      P = path(X_s,X_f), step(P,X_i,X_j) } <= 1 :-
  path(X_s,X_f), moves(M), T = 1..M.
```

The definition part of an encoding defines predicates for inferences contributing to a solution. For example, a change of location can be inferred based on volitional movement from the present location.

```
at(X_j,T+1) :- at(X_i,T), cmd_move(X_i,X_j,T),
                move(X_i,X_j,T).
```

The integrity constraint part restricts inferences, including those related to the agent's movements. For example, the agent can move from a location only if it is at that location, and the agent cannot move to locations occupied by an obstacle. (An underscore, below, is an anonymous variable that can take on any value.)

```
:- move(X_i,_,T), at(X_j,T), X_i != X_j.
:- move(_,X,T), obstacle(_,X,T).
:- move(_,X,_), obstacle(_,X).
```

Spatial relationships other than at are also accommodated in the representation, though falling short of generality. (Commonsense reasoning and qualitative spatial reasoning pose well-known and unresolved challenges.) Our account is necessarily brief and incomplete, for reasons of space, but the description should give the flavor. The important point is what the encoding can produce, summaries of the execution of a given navigation plan in the form of predicates.

- traversed(path(x_0,x_g)), at(x_g,T): The path was followed until the agent reached its goal, with the last action taken at time T.
- blocked(X_j,P), at(X_i,T): The path was blocked by some obstacle at location X_j; at time T the agent was left at X_i.
- stopped(X,P), at(X,T): The path was followed until the agent stopped at X; this predicate lets us distinguish quiescence from being prevented from moving.

These mechanics provide for the construction of a set of ground terms (i.e. predicates containing no variables) that form the "logic" of an explanation for a navigation problem instance. The representation is limited in its discretization of space and time, but it can manage simple changes over time. For example, obstacles may be permanent or temporary, as might be presented by a person or object moving across a path at a specific time steps: obst(o_2,x_4,3), obst(o_2,x_4,4). A strong limitation is that unpredicted errors and deviations in behavior cannot be explained by our approach.

Image schemas have been formalized in other mathematical and logical formalisms (Frank & Raubal,1999;Kuhn,2007;Walton & Worboys,2009). One subtle issue is the intended application of the formalization. Most such work aims at description of entitities and their changing relationships. We have to make an additional commitment to interpreting image schemas in logical and even causal terms, because that is what is necessary for explanations. A move action, for example, does more than describe what happens; an agent carries out the action *in order to* change its location. This is a commonplace assumption in planning but important to make explicit because logical/causal interpretation is central to explanation.

At the end of the next section we show how this formalism works for a real navigation problem.

A Navigation Scenario

This section lays out an example navigation scenario, one that we can use as a target for explanation.

System

The hardware for this work is a K-Bot platform, from University of Pennsylvania. The robot sensor package includes a microstrain 3DM-GX2 IMU, two Point Grey Firewire Grasshopper cameras, a Point Grey GigE Blackfly camera, a Point Grey Bumblebee2 stereo camera, and an ASUS Xtion pro RGB-D camera, as well as a two Hokuyo UTM-30LX-EW (Ethernet) Scanning Laser Rangefinder and a Velodyne HDL-32E LiDAR. Software is built on ROS (Robot Operating System) Indigo by the Open Source Robotics Foundation, plus ROS software drivers to read sensor data streams.

ROS's primary navigation stack is move_base, which implements a number of essential capabilities. Cost maps provide a two-dimensional representation, in the form of a grid, of the cost to traverse a space. Some cells in the map may have nominal cost (a constant FREE_SPACE), indicating that the robot may move freely through the location corresponding to the cell. Physical objects or obstacles also occupy cells in the grid, which means that the cost recorded in those cells is the maximum possible (a constant named LETHAL_OBSTACLE). The region of each object is "inflated," meaning that the cells surrounding the region occupied by the object have an intermediate cost, to indicate a location with a risk of collision.

Planners supply the robot with a path based on the values contained in the cost map, given a starting location and a goal location. The global planner creates a general path to follow, a discrete sequence of locations, between the start and the goal. The local planner is responsible for attempting to follow the global path by generating movement commands for the robot; this path includes orientation information for the robot. When we refer to the path planner in the remainder of this paper, we mean the global planner.

Task Environment and Execution

The robot is tasked with moving across a warehouse floor. For the purposes of this scenario, a region in the center of the room is traversable, but it is also considered vulnerable and should be avoided. In a military scenario such a region might correspond to an area visible to a hypothetical observer, which is undesirable for movement under concealment; in an urban search and rescue scenario, the region might be where the ceiling above has been weakened and may fall. This task was chosen for explanation because an alternative, shorter path to the goal location on the other side of the room is obvious: a straight line. If the contextual information is absent, a questioner might reasonably ask the robot to explain.

To implement the navigation task, a custom ROS node was written to add such regions to the navigation stack via a separate layer of the cost map, as virtual obstacles, with corresponding lethal obstacle cell costs. A separate layer was used because the robot updates the cost map as it moves through the environment. Virtual objects or regions are not detected by the system's raytracing algorithm, and the costs associated with such objects would be overwritten. A separate layer also limited the need to modify existing ROS `move_base` software.

April tags (Olson,2011) were used to represent the centroid of a region of vulnerability, mounted on a physical cone for easy detection. The radius of the region was set programmatically. This was an alternative to assessing vulnerability directly; it allows comparable virtual information to be integrated into the physical environment. In our discussion of the scenario below, we will treat the April tags and the cone as being invisible to the robot, which would be the case if the vulnerable region were directly assessed.

In a sample execution of the task, the robot starts at a location in the lower right of Figure 1. The goal location is in the upper left of the figure. The robot begins with an empty cost map; cells are assigned costs based on sensor information about obstacles, in this experiment both real and virtual. The path planner searches for and returns a path from the start to the goal, a sequence of a few hundred locations, each with associated bookkeeping information. The vulnerable region in this example has been given a `LETHAL_OBSTACLE` cost, but in a different variation it could be given some lower `VULNERABLE` cost. The robot ends at the goal location and its path is shown as a dashed red line.

The visual representation shows the cost map that the path planner accesses to construct the path that the robot follows. Different colors correspond to different cost values, in particular yellow for lethal obstacle cost, light gray for free space cost, and other colors for intermediate values produced by the "inflation" mentioned above. The outermost color is for regions with unknown cost. The physical borders of the room are clearly visible on the map; the upper area shows work tables and equipment. The circle on the map, with a sphere floating above, is the visual representation of the vulnerable region, derived from the April tag located at the center of circle. A physical obstacle would look similar, except that the region opposite the robot would not be visible to the its sensors and would thus have unknown cost values. In this scenario, the robot begins in a location from which all relevant cost values for path planning can be determined directly, which means that it can follow the path produced by the planner without the need for replanning.

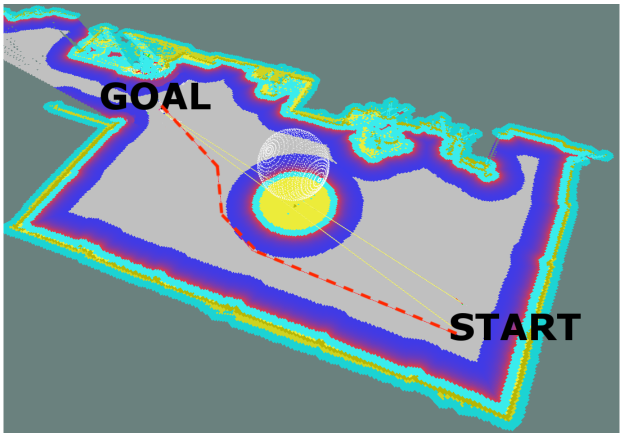


Figure 1: Movement around a region of vulnerability

Generating a representation

Figure 1 might be seen as an explanation: it indicates an obstacle in the center of the room, which the robot will avoid. By comparing a map with the physical environment it becomes clear that the object is virtual, representing a vulnerable region; the robot would presumably plan around it. But the phenomenon to be explained and the explanation are only implicit in the visualization. Even the basic vocabulary of navigation concepts is missing. The robot has no internal representation of these concepts, to impose an explanation on the visualization. At best we might say that the image helps viewers explain the robot's behavior to themselves.

What we want instead is the generation of an explicit abstract representation that captures the robot's behavior.

- *Locations*: All world locations are translated into 2D cells on the cost map, and those cells generate unique symbols. Not all locations or cells are used; relevant ones are identified in translating other kinds of objects.

- *Paths*: Three different types of paths are generated. A planned path is constructed from the set of waypoints produced by the ROS path planner. An external path is generated by an external process directly on the cost map (e.g., a hypothetical straight line path between two locations). An executed path is constructed from the sequence of the robot's sensed locations as it moves through the world. In the last case locations are sampled whenever a `cmd_vel` message is issued. After the conversion to cost map cells, path locations are filtered to remove jitter.

- *Commands*: For planned and external paths, locations are walked to create a set of `cmd_move` predicates, with programmatically generated timestamps. For executed paths, `cmd_vel` messages are converted to `cmd_move` predicates.

- *Landmarks, regions, and obstacles*: These are treated similarly, and in our navigation scenario, the vulnerable region can be any one of them, depending on the path and cost values. Generation is from an object's external specification. A directional relationship is computed to the object from each location on the path. These relationships are determined by a validated cognitive model of spatial projective terms, AVS (Regier & Carlson,2001), which we have used in previous robotics projects (Ward, St. Amant, & Fields,2017). The process that walks an executed path records the robot's pose (i.e., its location and the direction it is facing); for other paths, the direction is determined by projecting through the locations of future steps. The robot's pose plus the relative bounds of a landmark are sufficent for the model to estimate the acceptability rating of a description—in front of, behind, left, or right—for a spatial relationship. The highest-rated description is used.

The conversion is straightforward, but we present this level of detail to highlight the judgment calls necessary to make the problem of generating an explanation tractable. Locations are the main issue: the only locations generated for a problem instance are those relevant to the following of a given path, producing a few hundred locations in contrast to more than 5 million distinct cost map cells. Similarly, the entire region occupied by a landmark or an obstacle is not represented explicitly but only through relationships to path locations.

Explanations

Finally we reach explanations. Explanations can be divided into two types (Leddo & Abelson,1986). A *constructive* explanation is a direct application of the definition of scientific explanation, identifying the causal or logical factors that give rise to some result. *Contrastive* explanations make comparisons with one or more alternatives that may not be given explicitly. Generation of alternatives in general is challenging (Leddo & Abelson,1986) and we will assume that any alternatives are provided as input to the ASP inference process (as explained in more detail below).

A problem instance is generated as described for the path above. The ASP inference process fills in other predicates:

path(x_{4107},x_{4232}),

landmark(vul,x_{5446}),

at(x_{4107},1),move(x_{4107},x_{4108},1),at(x_{4108},2), ...

right(vul,58), ...

at(x_{4231},125),move(x_{4231},x_{4232},125),at(x_{4232},126),

finished(arrived(x_{4232},126))

Because the vulnerable region does not intersect the path, it is translated into a landmark; from the start until `T=58` the region is `in front of` the robot; it is to the `right` until `T=90` and thereafter `behind`. The necessary information for an explanation is provided, in general laws for the domain (e.g., requirements for and constraints on movement) and antecedents as specified in the problem instance. In words, "Why did the robot end up at(X_{4232},126)?" "Because it was at(X_{4231},125) [antecedent], there was a `step` [antecedent], it executed a `cmd_move` [antecedent], and the result was a `move` to that location [general law]." A set of such statements, working backward to the antecedent of the robot at the start of the path, constitutes a complete explanation representation.

For contrast, consider accounting for a hypothetical alternative path. In navigation, a straight line is a natural default for a human navigator (Korpan & Epstein,2018). Bresenham's algorithm is used to generate a straight-line sequence of cells on the cost map between source and goal locations. In the navigation scenario, with the vulnerable region given a `LETHAL_OBSTACLE` cost, the generation process puts obstacles on path locations inside the region. The inference process then generates the following:

path(x_{8037},x_{8292}),obst(vul,x_{8147}),obst(vul,x_{8148}),

at(x_{8037},1),move(x_{8037},x_{8038},1),at(x_{8038},2), ...

at(x_{8145},109),move(x_{8145},x_{8146},109),at(x_{8146},110),

finished(blockage(x_{8147},path(x_{8037},x_{8292})))

The inferred predicates describe the robot moving along the path until reaching a location just outside the vulnerable region; actions from that point `wait` until the time limit is reached. As with a constructive explanation, there exists a chain of domain laws and antecedents that account for this result. Said differently, it cannot be inferred that the robot reaches the goal location; it *can* be inferred that the robot will be at a different location at the time limit. The relevant domain law is the constraint that the agent cannot move to a location where there is an obstacle, which results in the inference of a blockage.

As a final example, we can change the cost of the region to `VULNERABLE` and use the same straight path. The explanation changes. In this case the robot can enter the region and that the path can be completed. As with the relationship to landmarks, the information that the robot was inside the vulnerable region is inferred, and this information forms the basis for comparison with other paths.

Discussion

We have presented a system for producing explanations of a robot's path planning and path following behavior. Part of our work is analytical. We adopted two well-known criteria for explanations: that they be understandable and that they provide a logical or causal account of a system's behavior. Image schemas satisfy the first criterion, by assumption; we have also presented evidence from the psychology literature that this assumption is plausible. Satisfying the second criterion involved showing that combinations of image schemas could be interpreted in logical or causal terms. The ASP formulation is a good match for the semantics of image schemas.

We are interested in the specific domain of navigation, though we expect to move next to consider the larger context of robot planning and acting. ASP supports inferences related to commonsense reasoning (e.g., that a landmark remains to my right from one step to the next as I move past).

Tradeoffs and limitions apply to our work. We call our explanations "lightweight" because the explanation is observational, not coupled with the robot's control processes. The main advantage of this approach is pragmatic: We can make many fewer assumptions about whether the robot is being controlled by a script, a state machine, an AI planner, or some other possibility. The disadvantage is that "why" inferences must be based on observations and domain knowledge without limited control information. For example, we can imagine a robot being programmed with a preference to pass on the left, with obstacles being observable on the right, but if this preference is not made public, the spatial relationship to an obstacle is descriptive rather than part of an explanation for why the robot chose its path.

Among the obvious limitations is the scope of the navigation task. It is reasonable to ask whether the representation and processing described in previous sections are necessary to explain such a simple activity. Activities with a cognitive component (in this case navigation tasks in general but also our use of image schemas for representation) do often turn out to be subtle underneath, but it will require human-robot interaction studies to evaluate the need for and the adequacy of explanations in this domain.

This limitation suggests another: representations are difficult to evaluate in the abstract. Mueller et al. (2019), in an extensive literature review of explainable AI, identify properties of good explanations and empirical techniques for evaluation. Our work produces representations but not the surface form of explanations, and we have not yet subjected them to evaluation. Once textual (or multimedia) explanations can be generated, human studies will be needed.

References

Balduccini, M. (2009). How flexible is answer set programming? An experiment in formalizing commonsense in ASP. In *Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning* (pp. 4–16).

Cervel, M. S. P. (1999). Subsidiarity relationships between image-schemas: an approach to the force schema. *Journal of English Studies*(1), 187–208.

Chakraborti, T., Sreedharan, S., Zhang, Y., & Kambhampati, S. (2017). Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. *arXiv preprint arXiv:1701.08317*.

Croft, W., & Cruse, D. A. (2004). *Cognitive linguistics*. Cambridge University Press.

Fox, M., Long, D., & Magazzeni, D. (2017). Explainable planning. *arXiv preprint arXiv:1709.10256*.

Frank, A. U., & Raubal, M. (1999). Formal specification of image schemata—a step towards interoperability in geographic information systems. *Spatial Cognition and Computation*, *1*(1), 67–101.

Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., & Schneider, M. (2011). Potassco: The Potsdam answer set solving collection. *AI Communications*, *24*(2), 107–124.

Hempel, C. G., & Oppenheim, P. (1948). Studies in the logic of explanation. *Philosophy of Science*, *15*(2), 135–175.

Johnson, M. (1987). *The body in the mind: The bodily basis of meaning, imagination, and reason*. University of Chicago Press.

Korpan, R., & Epstein, S. L. (2018). Toward natural explanations for a robot's navigation plans. In M. de Graaf, B. Malle, A. Dragan, & T. Ziemke (Eds.), *Notes from the Explainable Robotic Systems Worshop, Human-Robot Interaction 2018.*

Kuhn, W. (2007). An image-schematic account of spatial categories. In *Proceedings of the International Conference on Spatial Information Theory* (pp. 152–168).

Lakoff, G. (1987). *Women, fire, and dangerous things*. University of Chicago Press.

Leddo, J., & Abelson, R. P. (1986). The nature of explanations. In *Knowledge structures* (pp. 103–122). LEA.

Li, J. J. (2012). Qualitative spatial and temporal reasoning with answer set programming. In *Proceedings of ICTAI* (Vol. 1, pp. 603–609). IEEE.

Lombrozo, T., & Carey, S. (2006). Functional explanation and the function of explanation. *Cognition*, *99*(2), 167–204.

Mandler, J. M., & Pagán Cánovas, C. (2014). On defining image schemas. *Language and Cognition*, *6*(4), 510–532.

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. *arXiv preprint arXiv:1902.01876*.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *Proceedings of ICRA* (p. 3400-3407). IEEE.

Regier, T., & Carlson, L. A. (2001). Grounding spatial language in perception: an empirical and computational investigation. *Journal of experimental psychology: General*, *130*(2), 273.

Rosenthal, S., Selvaraj, S. P., & Veloso, M. M. (2016). Verbalization: Narration of autonomous robot experience. In *Proceedings of IJCAI* (pp. 862–868).

St. Amant, R., Morrison, C. T., Chang, Y.-H., Mu, W., Cohen, P. R., & Beal, C. (2006). An image schema language. In *Proceedings of ICCM* (pp. 292–297).

Timpf, S. (2002, Mar 01). Ontologies of wayfinding: a traveler's perspective. *Networks and Spatial Economics*, *2*(1), 9–33.

Walton, L., & Worboys, M. (2009). An algebraic approach to image schemas for geographic space. In *Proceedings of the International Conference on Spatial Information Theory* (pp. 357–370). Springer.

Ward, J., St. Amant, R., & Fields, M. (2017). Spatial relationships and fuzzy methods: Experimentation and modeling. In *Proceedings of ICCM.*

Wiener, J. M., Büchner, S. J., & Hölscher, C. (2009). Taxonomy of human wayfinding tasks: A knowledge-based approach. *Spatial Cognition & Computation*, *9*(2), 152–165.

Cognitive Metrics Profiling of a Complex Task: Toward Convergent Validity with Behavioral and EEG Workload Indicators

Christopher A. Stevens (christopher.stevens.28@us.af.mil)
Air Force Research Laboratory
WPAFB, OH 45433 USA

Megan B. Morris (megan.morris.1.ctr@us.af.mil)
Ball Aerospace
Fairborn, OH 45324 USA

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil) **Christopher Myers (christopher.myers.29@us.af.mil)**
Air Force Research Laboratory
WPAFB, OH 45433 USA

Abstract

Workload assessment remains a challenging, multi-dimensional problem. A variety of metrics are available (behavioral, physiological, subjective), but their relationships to each other and the underlying cognitive processes producing workload are not well understood. In the present paper, we extend an approach known as Cognitive Metrics Profiling to an unmanned vehicle control simulation. We show how the model predicts behavioral performance and physiological indicators of global workload in the task and produces insights about sources of workload.

Keywords: cognitive workload; ACT-R; EEG

Introduction

Cognitive workload has been the focus of empirical and theoretical investigation for many decades (Cain, 2004). Limitations in workload have captured the interest of cognitive scientists, in part, because its effects pervade a wide range of tasks, and it has direct practical implications. As an abstract concept, workload is challenging to measure directly. Several indirect measures have been used in the literature, including subjective, behavioral, and physiological measures, each with advantages and disadvantages.

Subjective or self-reported measures, are straightforward to implement, but can be obtrusive if administered during a task, and are vulnerable to biases when assessed retrospectively (Matthews et al., 2015). Behavioral measures include primary task or secondary task performance (e.g., accuracy,reaction time). Primary task performance is unobtrusive, but doesn't indicate how much capacity may remain for addiitonal tasks. Secondary tasks can indicate spare capacity, but are more obtrusive than primary tasks (Miller, 2001). Physiological measures such as electroencephalogram (EEG) and electrocardiography (ECG) have the advantage of being unobtrusive and capturing changes in workload across time; however, they are often contaminated with reactions to other factors such as the environment (Miller, 2001).

The difficulty in defining and measuring workload poses challenges for theoretical progress (Cain, 2004). Cognitive architectures have the potential to provide a much needed theoretical framework for informing existing workload metrics. Cognitive architectures have two advantages: (1) as formal models, they provide precise descriptions of cognitive mechanisms and processes that underlie workload, and (2) as integrative theories, cognitive architectures are applicable to an array of both simple and complex tasks across many cognitive domains.

Cognitive Metrics Profiling (CMP) is one of the first efforts to define and measure workload within a cognitive architecture (Gray, Schoelles, & Sims, 2005; Gray, Schoelles, & Myers, 2005). CMP uses the Adaptive Control of Thought Rational (ACT-R; Anderson et al., 2004) cognitive architecture to characterize the evolving cognitive demands of a task and link those demands to performance predictions. In CMP, workload is defined as a weighted sum of activity across multiple information processing modules (e.g., vision, motor, and declarative memory).

Initial validation of CMP has been promising. Past work has indicated a correlation between CMP and behavioral and subjective indicators of workload (Jo et al., 2012). However, CMP currently has two major limitations: 1) It has been evaluated only in small-scale tasks of short duration (e.g. paired associates) (Gray, Schoelles, & Sims, 2005; Jo et al., 2012) and 2) the relationship between CMP estimates and physiological indicators of workload has not been investigated.

In the present paper, we seek to further validate and extend CMP in two ways: first, we investigate the relationship between CMP and established EEG metrics of workload to further establish convergent validity, and second, we use CMP in an unmanned vehicle operator task to test its scaleability to more complex tasks.

Cognitive Metrics Profiling

Theory and Rationale

CMP uses the ACT-R cognitive architecture to quantify the degree to which cognitive resources (e.g., memory or vision) are taxed during a given task. A profile detailing cognitive resource usage can be analyzed to understand how task demands affect cognition. For example, if the declarative memory module is in use for 80% of the task then the memory demand of the task would be very high, making it difficult to take on additional tasks also heavy in memory demand. Alternatively, CMP can be used to measure global workload, which is defined as a weighted sum of activity across modules. Jo et al. (2012) found that global workload derived from CMP is correlated with subjective workload judgments.

Present work

In the present work, participants completed an UV task and we compared EEG and ECG metrics to workload profiles generated from CMP. We used a UV task because it induces a wide range of workload levels and taxes ACT-R modules to varying degrees. Together, these factors provide a wide range of workload conditions with which to validate CMP against physiological indicators of workload.

Predictions

Figure 3 lists the predicted relationships between physiological workload indicators and model-based workload generated from CMP. Several studies have shown EEG band frequencies correspond to manipulations of cognitive workload (Borghini et al., 2014; Lean & Shan, 2012). Alpha and theta have been shown to decrease and increase with cognitive workload, respectively. Research has also suggested that different frequencies within the alpha band capture different aspects of workload, with lower alpha (8-10 Hz) reflecting alertness and upper alpha (10-13 Hz) reflecting information-processing (Klimesch, 1999). Ratios of band frequencies have also shown some promise, with the Task Load Index (TLI), a ratio of theta and alpha, and the Engagement Index (EI), a ratio of beta to alpha and theta, increasing with increased workload and task engagement, respectively (e.g., Kamzanova et al., 2014; Freeman et al., 1999). In addition to EEG metrics, heart rate variability metrics have been shown to decrease with increased cognitive workload (Lean & Shan, 2012).

Method

Participants

Ten volunteer employees (M_{age} = 29.30; SD_{age} = 6.99; R_{age} = 19-41; $Proportion_{male}$ = 50.00%) from Wright-Patterson Air Force Base (WPAFB) who were unfamiliar with the task completed an informed consent document and participated in the study. Participants reported normal or corrected-to-normal vision, normal color vision, and normal hearing. This study was approved by the Air Force Research Laboratory (AFRL) Institutional Review Board (IRB).

Task Description

Participants completed two 60-minute missions of varying difficulty in IMPACT (Intelligent Multi-UxV Planner with Adaptive Collaborative/Control Technologies; Draper et al., 2017; Rowe et al., 2015)—a high fidelity UV simulator. We manipulated the task density of the missions—the number and difficulty of the tasks—to induce low vs. high levels of work load. The IMPACT environment consists of an array of monitors and two modes of communication: a microphone and headset for auditory communication, and a communication window for text communication and alerts. Alerts and instructions to complete tasks are presented primarily through the communication window and, to a lesser extent, via the headset. The primary monitor features a map of the base, a menu system for selecting and managing UVs, and a communication window. Three secondary monitors provide alternative views of the base map, UV sensor and status information, and a reference manual instructing participants how to dispatch UVs for various tasks.

During a mission, the participant must complete a variety of tasks that require scheduling and planning, resource management, multi-stage decision making, communication, and information search and acquisition. Many tasks require the participant to dispatch UVs either at predetermined times for routine surveillance or in response to sporadic security events. In order to correctly dispatch a UV, the participant must select a UV with attributes required by the task, such as the destination, UV type (e.g., aerial vs ground), optional automatic termination, and maneuver (e.g., inspection or blockade). Participants can terminate UV tasks either through scheduled automation or manually upon receiving instruction via the communication system. Other types of tasks occur periodically throughout the mission. For example, participants must reallocate resources in response to environmental or mechanical problems, and answer information queries, requiring information to be found within the interface and relayed via the communication window.

Protocol

Data collection for each participant was performed separately over the course of a single day. First, participants were trained to use the IMPACT system and to perform base defense actions. Training included participants performing a capstone mission where the experimenter revealed any behavioral errors and allowed the participant to ask questions. After training, participants were fitted with EEG and heart rate physiological sensors and participant eye gaze was calibrated to an eye tracking system. Participants then completed a low and a high task density condition. Both conditions were 60 minutes in length and were counterbalanced across participants.

Unmanned Vehicle Model

We developed a model of the UV task within the ACT-R cognitive architecture (Anderson et al., 2004). A cognitive architecture is a formal, computational framework for simulating and testing comprehensive theories of cognition (Newell, 1990). The ACT-R cognitive architecture consists of specialized information processing modules, spanning procedural and declarative memory, visual and auditory perception, speech production, and motor execution. Cognition unfolds over a series of production cycles which coordinates the flow of information among the modules. Importantly, module activity within the architecture forms the basis for workload measurement within CMP.

In the interest of brevity, we will focus on the high level strategy employed by the model, such as how it searches the interface for tasks and how it resolves conflicts between competing goals. The model's strategy is illustrated as a flow chart in Figure 1. The strategy is composed of three primary phases: an active search phase, a passive monitoring/waiting phase, and a task execution phase. During the search phase, the model inspects three locations within the interface for new tasks: (1) a message window where information queries and new task alerts appear, (2) a base map where certain problem events appear, and (3) a list of Random Anti-Terror Measures (RAMs) with target execution times and deadlines.

When a task is found, the model performs the task and rechecks the interface for new tasks that might have become available during the intervening time. When no task is found, the model proceeds to the next location. At the third location, the model compares the mission clock to the target times for the RAMs. The model will perform a RAM if the mission time is within a parameter we term leading time—a period of time preceding the deadline, during which the model will attempt to complete the task. If no RAMs can be completed, the model enters a passive monitoring phase, in which it waits for the next RAM and responds to events that pop up in the interface. Periodically, during the monitoring phase, the model will ”re-calibrate” its internal clock to the mission clock in an effort to mitigate growing temporal estimation error.

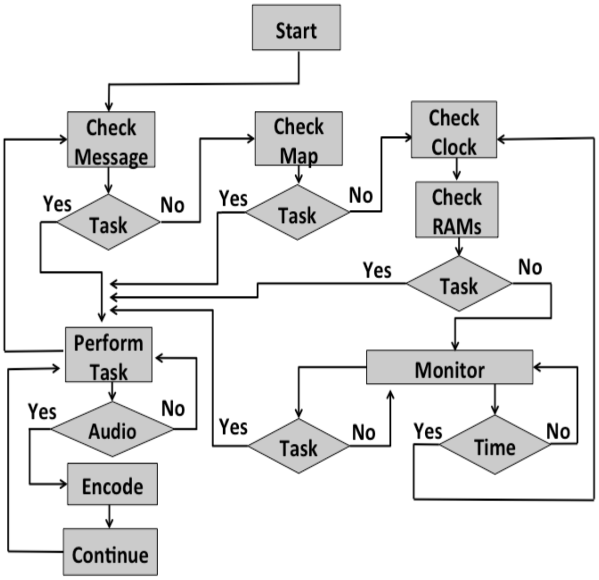


Figure 1: A flow chart of the model’s high-level strategy for the UV task. Boxes represent processes and diamonds represent decision points.

The model primarily uses a first come, first served policy to manage competing task demands. This is why the model reinspects the interface for new tasks upon the completion of a current task. One minor exception to the rule occurs when an audio message is presented during an ongoing task. In this case, the model briefly suspends the ongoing task to encode the message, resumes the suspended task, and later attempts to complete the task associated with the auditory message.

Workload and Performance Measures

Physiological Workload Metrics We collected EEG, heart rate variability, and eye tracking data for our physiological workload metrics. Throughout the recordings, the eye tracking system had difficulty locating participant’s eyes due to the experiment environment, producing several missing values. As a result, the ocular methodology and data are not reported. A list of workload metrics and respective calculations can be found in Table 1. EEG data was collected with a sampling rate of 500 Hz from a dry electrode Quick-20 Cognionics headset (Cognionics, CA, USA). Electrode locations followed the 10-20 system with 19 active channels (Fp1, Fp2, Fz, F3, F4, F7, F8, Cz, C3, C4, T3, T4, Pz, P3, P4, P7, P8, O1, and O2), two grounds located adjacent to Fp1 and Fp2, and a linked ear reference. Electrode contacts consisted of silver/silver chloride matrixed for conductivity. EEG data was cleaned and processed offline with an in-house script utilizing the MATLAB (The MathWorks, Inc., MA, USA) toolbox EEGLAB (Delorme & Makeig, 2004). EEG data was re-referenced to the linked ear reference and filtered using a Parks-McClellan optimal equiripple finite impulse response (FIR) band-pass (high-pass cutoff 1 Hz and low-pass cutoff 95 Hz) and notch filter (60 Hz). DC offset was removed and a recursive least squares adaptive filter was used to remove eye artifacts. To calculate EEG metrics, average band power was extracted from 10 second epochs with no overlap using a modified periodogram spectral estimator with a Hanning window. Specific metrics were then calculated from these band power values and log transformed.

Inter-beat (RR interval) data was collected with a sampling rate of 18 Hz from a Zephyr Bioharness 3.0 (Zephyr Technology Corp., MD, USA). RR interval data was cleaned offline with an in-house script by identifying outliers with a percent change strategy based on data epochs (e.g., Kemper et al., 2007; Persson et al., 2005). Outliers were removed and linear interpolation was utilized to extract values to replace the outliers (e.g., Peltola, 2012).

Performance Evaluation Due to heterogeneity of the tasks, we evaluated performance according to criteria that depended on task-specific requirements. A score of 1 was recorded if a participant satisfied a criterion and 0 otherwise. For example, events that required the deployment of a UV typically included a 3 minute deadline, correct destination, correct UV attributes (e.g., correct sensor), correct operation (e.g., aerial inspection) and a category for miscellaneous situation-dependent constraints (e.g., scheduled termination of a task). Information queries were evaluated according to a 3 minute deadline and the correctness of the response.

Model Fitting We varied two parameters that exert broad, cascading effects on the task dynamics and resource engagement: latency factor, which affects overall memory retrieval times by scaling memory activation, and leading time, which specifies how far in advance a RAM is completed relative to its deadline. In order to find the best-fitting parame-

Table 1: Physiological metrics and respective calculations.

Metric	Calculation
Alpha	Band power in range of 8 - 13 Hz located at Pz site.
Lower Alpha	Band power in range of 8 - 10 Hz located at Pz site.
Upper Alpha	Band power in range of 10 - 13 Hz located at Pz site.
Theta	Band power in range of 4 - 8 Hz located at Fz site.
Frontal Theta	Calculated as the average of theta at F3 and F4 sites (e.g., Kamzanova et al., 2014).
TLI	Calculated as theta (Fz)/alpha (Pz) (Gevins & Smith, 2003; Kamzanova et al., 2014).
EI	Calculated as beta/(alpha + theta) from averages of sites Cz, P3, Pz, and P4 (Kamzanova et al., 2014; Freeman et al., 1999).
Mean HRV	Calculated as the mean of RR intervals.
Median HRV	Calculated as the median of RR intervals.

ters, we performed a grid search in which latency factor $\in \{.5, 1.0, 1.5\}$ and leading time $\in \{2, 4, 8\}$ were varied independently. We simulated the model 20 times for each parameter set [1].

The model predicted the number of participants that satisfied each criterion for each task. The fit of the model was evaluated according to a normalized root mean squared error (NRMSE) measure based on the standard deviation of the binomial distribution. Some advantages of this approach include: (1) ease of interpretation, (2) it is more stringent at the boundaries (e.g., 90% correct) where data are less variable, and (3) it requires no pooling across heterogeneous data.

NRMSE was computed as:

$$NRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{I}\sum_{c=1}^{C_i}\left(\frac{\hat{x}_{i,c}-x_{i,c}}{\sigma_{i,c}}\right)^2}$$

where $i = [1, 2..., I]$ is an event index, $c = [1, 2..., C_i]$ is a criterion index for each event, $N = \sum_{i=1}^{I} C_i$ is the total number of criteria across all events, $x_{i,c}$ is the number of participants who satisfied criterion c for event i and $\hat{x}_{i,c}$ is the corresponding prediction. The standard deviation is computed as $\sigma_{i,c} = \sqrt{S \cdot p_{i,c} \cdot (1 - p_{i,c})}$, where S is the number of participants and $p_{i,c}$ is the proportion of participants who satisfied criterion c for event i. In cases where $p_{i,c} = 1$, we adjusted the value downward to the next possible value of $\frac{9}{10}$ to prevent division by zero.

Results

Behavioral Performance

After excluding 8 complex tasks that were difficult to model, there were 27 and 43 tasks remaining in the low and high task density conditions, respectively. To assess our workload manipulation, we averaged across all tasks and their criteria to yield two overall accuracy scores per subject—one for low task density and one for high task density. A paired *t*-test ($t(9) = 5.93$, $p = .00$, $d = 1.87$) revealed an effect of task density on accuracy, high ($M = .84$) vs low ($M = .94$). As predicted, mean subjective workload, as measured by the NASA-TLX, was higher in the high ($M = 53.33$) vs low ($M = 20.58$) task density condition, (paired *t*-test, $t(9) = 8.75, p = .00, d = 2.77$).

Model Results

The best-fitting parameters were latency factor = 1 and leading time = 8 (NRMSE = 2.03), suggesting that subjects were proactive in setting up RAMs in advance of their deadlines. The predicted accuracy was .98 and .90 for low and high task density, respectively. Although the model tended to overestimate accuracy, it was able to capture the qualitative drop in performance.

Workload was computed according to the formulas described in (Jo et al., 2012) using consecutive time windows of 10 seconds (see Figure 2). Across the entire mission, mean workload was higher under high task density (2.30) compared to low task density (1.59), mirroring the behavioral performance results and subjective workload assessments.

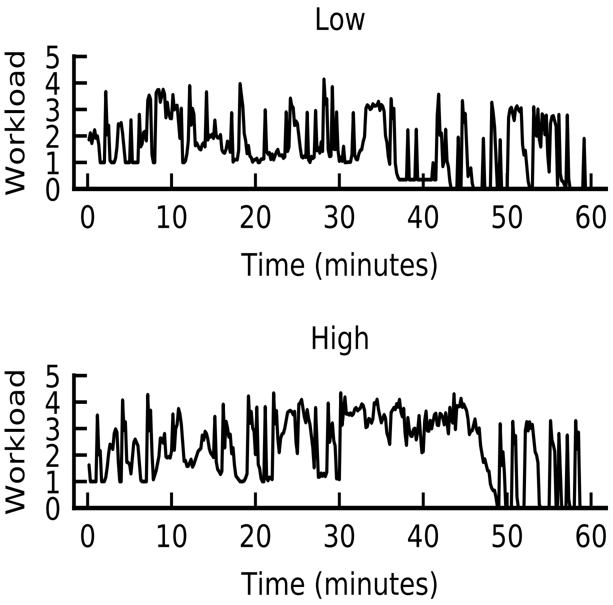


Figure 2: A comparison of global workload profiles generated by the model in the low workload and high workload conditions.

[1] The grid search was small due to simulation times and the fact that fit was only moderately sensitive to changes in parameter values.

Workload Regression

We examined the association of the physiological workload metrics with workload derived from the cognitive model (values were rounded to the nearest whole number to create five workload levels, 0 - 4) and task density condition (low vs high) using linear mixed effects modeling (LMM). We performed robust linear mixed effects modeling (RLMM) from the *robustlmm* package (Koller, 2016) in *R* (R Core Team, 2017) due to violations of residual normality and homoscedasticity assumptions. Baseline models included the metric of interest and a random intercept for subjects. Augmented models included the workload predictors of interest (1. Level, 2. Level and Condition, 3. Level, Condition, and Level x Condition) and a random intercept for subjects. Robustified estimating equations from RLMM do not correspond to likelihood statistics. As a result, we could not compare the models with an ANOVA or obtain *p* values for the fixed effects. However, Wald confidence intervals can be used to examine the significance of the fixed effects.

RLMM analyses indicate that only EI had significant workload level and task density condition effects, suggesting increased task engagement as model workload level increased and as task density condition increased. The other EEG and heart rate metrics suggested marginal and trending effects in the expected directions, except for theta and TLI metrics in terms of task density condition (see Figure 3).

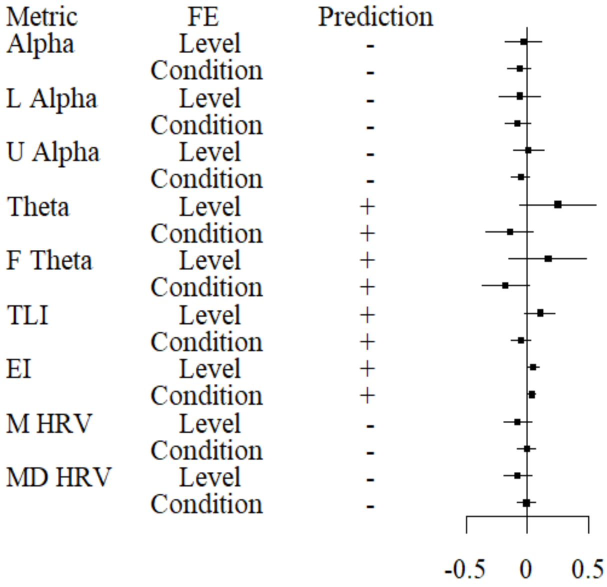


Figure 3: Predicted direction of relationship and regression coefficients between physiological workload and model-based workload. Main effects of level and condition are shown. Dots represent mean coefficient estimates and horizontal lines represent 95% confidence intervals. HRV coefficients were re-scaled by .01 for ease of presentation.

Discussion

CMP is a promising technique for characterizing workload, but it remains untested in complex environments and its convergent validity with other workload indicators, especially physiological indicators, has not been fully established. In the present study, we applied CMP to a multiple-vehicle control task that takes place over an extended time (60 minutes). Further, we examined the relationship between workload estimates generated by CMP and physiological indicators commonly associated with cognitive load or global cognitive activity. We found preliminary evidence for a relationship with one such indicator, EI, suggesting that the activity of an ACT-R model could be a valid way to characterize the cognitive resources utilized by a task. This is a potentially useful technique for predicting overall workload levels and the specific cognitive capacities affected by high workload moments.

We have provided a proof-of-concept here that CMP can discriminate between low and high workload conditions even in tasks that involve many complex interrelated subtasks over a long period of time. The CMP model predicts both an increase in subjective workload and a decrease in performance across subtasks in high complexity conditions, as was observed here. Future studies should look at CMP predictions across a wider range of task difficulties to confirm that it adequately captures the shape of the relationship between task difficulty and predicted workload.

This study adds to previous work relating ACT-R models to neural activity. It has been shown previously that buffer activity in ACT-R can be related to the BOLD signal in fMRI, suggesting that buffers may be meaningfully associated with activation of certain populations of neurons in the brain (Borst & Anderson, 2015; Qin et al., 2003). Moreover, it has been demonstrated that activity predicted by ACT-R can be recovered using Hidden Semi-Markov modeling of EEG data (Anderson et al., 2016). The present study adds to that growing body of research by suggesting that ACT-R activity may also be associated with neural indicators of cognitive activity.

We believe the tentative relationship between CMP and EI makes sense given that EI is thought to reflect multiple cognitive processes and resources. However, this is a relationship that warrants further exploration and clarification. The relationship observed here between model activity and EI is still very tentative due to noise in the indicator itself and the absence of a specific physiological model relating the two quantities. However, we propose these results justify confirmatory studies to test the hypothesis that CMP and EI may both characterize similar cognitive processes associated with workload. If this relationship is further explored, CMP may offer a potential integrative framework for behavioral, physiological, and subjective workload metrics, improving our ability to understand how they relate to each other and to the cognitive operations that they measure.

Acknowledgments

The opinions expressed herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries, or employees. This research was supported by a grant from the Air Force Research Laboratory Airman Systems Directorate Chief Scientist Venture Fund. The authors wish to thank Allen Dukes and Justin Estepp for their expertise with physiological data collection and analysis. The authors further thank Sarah Spriggs and Gloria Calhoun for overseeing data collection and for their comments and suggestions on the current work. The third author was supported by the Postgraduate Research Participation Program at the Air Force Research Laboratory through the Oak Ridge Institute for Science and Education and Department of Energy.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Anderson, J. R., Zhang, Q., Borst, J. P., & Walsh, M. M. (2016). The discovery of processing stages: Extension of sternbergs method. *Psychological review*, *123*(5), 481.

Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D., & Babiloni, F. (2014). Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, *44*, 58–75.

Borst, J. P., & Anderson, J. R. (2015). Using the act-r cognitive architecture in combination with fmri data. In *An introduction to model-based cognitive neuroscience* (pp. 339–352). Springer.

Cain, B. (2004). *A Review of the Mental Workload Literature* (Tech. Rep.). Defence Research and Development.

Delorme, A., & Makeig, S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of neuroscience methods*, *134*(1), 9–21.

Draper, M., Calhoun, G., Hansen, M., Douglass, S., Spriggs, S., Patzek, M., ... others (2017). Intelligent multi-unmanned vehicle planner with adaptive collaborative/control technologies (impact). In *19th international symposium on aviation psychology* (p. 226).

Freeman, F. G., Mikulka, P. J., Prinzel, L. J., & Scerbo, M. W. (1999). Evaluation of an adaptive automation system using three eeg indices with a visual tracking task. *Biological psychology*, *50*(1), 61–76.

Gevins, A., & Smith, M. E. (2003). Neurophysiological measures of cognitive workload during human-computer interaction. *Theoretical Issues in Ergonomics Science*, *4*(1-2), 113–131.

Gray, W. D., Schoelles, M. J., & Myers, C. W. (2005). Profile before optimizing: A cognitive metrics approach to workload analysis. In *Chi 2005* (pp. 1411–1414). Portland, Oregon.

Gray, W. D., Schoelles, M. J., & Sims, C. (2005). Cognitive Metrics Profiling. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *49*(12), 1144–1148.

Jo, S., Myung, R., & Yoon, D. (2012). Quantitative prediction of mental workload with the act-r cognitive architecture. *International Journal of Industrial Ergonomics*, *42*(4), 359–370.

Kamzanova, A. T., Kustubayeva, A. M., & Matthews, G. (2014). Use of eeg workload indices for diagnostic monitoring of vigilance decrement. *Human factors*, *56*(6), 1136–1149.

Kemper, K. J., Hamilton, C., & Atkinson, M. (2007). Heart rate variability: impact of differences in outlier identification and management strategies on common measures in three clinical populations. *Pediatric research*, *62*(3), 337.

Klimesch, W. (1999). Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain research reviews*, *29*(2-3), 169–195.

Koller, M. (2016). robustlmm: An R package for robust estimation of linear mixed-effects models. *Journal of Statistical Software*, *75*(6), 1–24. doi: 10.18637/jss.v075.i06

Lean, Y., & Shan, F. (2012). Brief review on physiological and biochemical evaluations of human mental workload. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *22*(3), 177–187.

Matthews, G., Reinerman-Jones, E., Barber, D. J., & Abich, J. I. (2015). The psychometrics of mental workload: Multiple measures are sensitive but divergent. *Human Factors*, *57*, 125-143.

Miller, S. (2001). Workload measures. *National Advanced Driving Simulator. Iowa City, United States*.

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.

Peltola, M. (2012). Role of editing of rr intervals in the analysis of heart rate variability. *Frontiers in physiology*, *3*, 148.

Persson, H., Kumlien, E., Ericson, M., & Tomson, T. (2005). Preoperative heart rate variability in relation to surgery outcome in refractory epilepsy. *Neurology*, *65*(7), 1021–1025.

Qin, Y., Sohn, M.-H., Anderson, J. R., Stenger, V. A., Fissell, K., Goode, A., & Carter, C. S. (2003). Predicting the practice effects on the blood oxygenation level-dependent (bold) function of fmri in a symbolic manipulation task. *Proceedings of the National Academy of Sciences*, *100*(8), 4951–4956.

R Core Team. (2017). *R: A language and environment fr statistical computing*. Vienna, Austria. (URL https://www.R-project.org/.)

Rowe, A. J., Spriggs, S. E., & Hooper, D. J. (2015). Fusion: a framework for human interaction with flexible-adaptive automation across multiple unmanned systems.

A Spiking Neural Architecture that Learns Tasks

Niels Taatgen (n.a.taatgen@rug.nl)
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
Cognigron Center for Cognitive Systems and Materials
University of Groningen, Nijenborgh 9, 9747 AG Groningen, Netherlands

Abstract

Cognitive architectures based on neural networks typically use the Basal Ganglia to model sequential behavior. A challenge for such models is to explain how the Basal Ganglia can learn to do new tasks relatively quickly. Here we present a model in which task-specific procedural knowledge is stored in a separate memory, and is executed by general procedures in the Basal Ganglia. In other words, learning happens elsewhere. The implementation discussed here is implemented in the Nengo cognitive architecture, but based on the principles of the PRIMs architecture. As a demonstration we model data from a mind-wandering experiment.

Keywords: Spiking neural networks; Mind Wandering; Basal Ganglia; PRIMS; Nengo; Skill Acquisition

Model code: https://github.com/ntaatgen/NengoPRIMs

Introduction

Symbolic cognitive architectures are very powerful in producing flexible task performance. Part of task performance is the ability to carry out steps in sequence. Although a production system, the typical symbolic solution to sequential behavior, is a straight-forward solution, it is less clear how it is carried out by the brain. The brain structures that are typically implicated in sequential behavior are the Basal Ganglia and the Thalamus[1]. For example, numerous ACT-R studies map model activity onto brain areas, of which procedural memory is mapped onto the Basal Ganglia (Anderson et al., 2004). Several neural network architectures that include sequential behavior have forwarded proposals for possible Basal Ganglia implementations (Stocco, Lebiere, & Anderson, 2010; O'Reilly & Frank, 2006; Eliasmith et al., 2012). However, these implementations impose quite some constraints on production rules. In the Stocco et al. implementation, the amount of information that can be transferred between modules is limited to a single item of information. The Eliasmith et al. solution does allow for the transfer of multiple items, but has no clear way in which the procedural knowledge is learned. In addition, one may wonder whether all human procedural knowledge, which is often quite task-specific, can be stored in a structure as small as the Basal Ganglia.

The work presented here is not a completely new proposal for sequential behavior, but builds on the Eliasmith et al. (2012) solution in Nengo, ACT-R (Anderson, 2007) and the PRIMs theory (Taatgen, 2013). A common idea among these theories is that procedural knowledge involves controlling the flow of information between different cognitive modules. For example, in order to perform an Aural-Vocal task in which a number has to be spoken based on the pitch of a tone (i.e., when you hear a low tone you have to say ”One”, when you hear a middle tone you have to say ”Two”, etc.), an Aural module determines the pitch, a Declarative memory module determines the mapping from pitch to number, and a Vocal module speaks the number. The role of procedural knowledge is to take the result of the Aural module and feed this into the Declarative module, and once the Declarative module successfully produces a result, move that result to the Vocal module.

If we assume that the knowledge to carry out a procedural task such as the aural-vocal task is encoded in the Basal Ganglia, we have a problem. Tasks such as the aural-vocal task, and also more complicated tasks that are typically part of psychological experiments, can typically be carried out by subjects after a short instruction and very little practice, even though they have never done these tasks before. It is therefore not very likely that they train their Basal Ganglia in that short period for this specific purpose. We therefore have to look for a solution that uses existing representations in the Basal Ganglia to do new tasks. To develop such a solution, it is useful to look at the PRIMs architecture (Taatgen, 2013). In PRIMs, procedural knowledge is decomposed into a fixed set of primitive operations. Each of these operations either makes a single comparison, or performs a single action by transferring one knowledge element from one module to another. Because the set of PRIMs is finite, we can imagine a Basal Ganglia model that is capable of carrying out any of the PRIMs, and is therefore in principle capable of performing any sequential task that can be defined in terms of PRIMs.

In this paper, I will first describe the overall architecture of the Nengo/PRIMs model. It resembles the Spaun model, a Nengo model that is capable of carrying out a range of tasks Eliasmith et al. (2012). The main difference between the two is that Spaun's procedural knowledge is hardcoded in the Basal Ganglia, whereas the Nengo/PRIMs model only encodes PRIMs in the Basal Ganglia, and uses a memory system to trigger the correct PRIM at the right moment. I will then use it to model an experiment by Smallwood et al. (2011).

[1] To save space and improve readability, I will refer to the Basal Ganglia/Thalamus combination as just the Basal Ganglia for the rest of the paper.

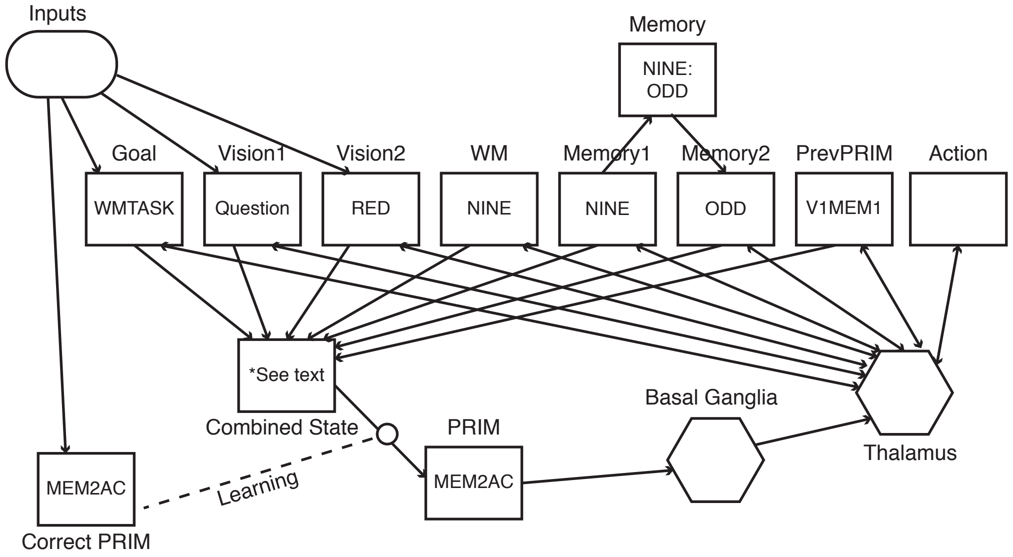


Figure 1: Overview of the Nengo/PRIMs model. Rectangles represent slots that can hold a single semantic pointer. Hexagons are more complex neural structures. The rounded rectangle provides the inputs to the network at scheduled times.

Overview of the System

Nengo basics: Semantic Pointers

Nengo is a neural network architecture based on spiking neurons. Clusters of neurons are used to represent vectors of numbers, and mappings between these clusters can calculate functions. For example, we can define a cluster of 100 spiking neurons to represent the vector $\begin{pmatrix} x \\ y \end{pmatrix}$, and connect this to another cluster of spiking neurons that will calculate and represent $\begin{pmatrix} x^2 \\ y^2 \end{pmatrix}$.

The next level of abstraction is to let these vectors represent symbols. For example, a particular vector of numbers can represent the color RED (we use 128 dimensional vectors in the model here). A symbol, represented by a vector of numbers, is called a *semantic pointer* in Nengo. Semantic pointers can represent simple symbols, but can also be convolved to create more complex representations. For example, we can represent a red ball by the following vector:

$$\text{REDBALL} = \text{COLOR} \odot \text{RED} + \text{SHAPE} \odot \text{ROUND}$$

Structure of the Model

With semantic pointers Nengo is capable of representing quite powerful knowledge structures, which can be manipulated with the appropriate mappings between clusters of neurons. The structure we will use is depicted in Figure 1. Each of the rectangles in the Figure represents a cluster of neurons that holds a single semantic pointer (we will call them ”slots” in this paper). The horizontal row of rectangles represents a set of slots that hold information related to particular cognitive modules, similar to buffer slots in ACT-R. For illustration purposes, some values have been put into the boxes. They are related to the experimental task to be discussed later. The *Goal* slot represents the current task. It, together with the visual input, is set by a separate process represented by the rounded rectangle. This process sets the values in these slots to particular values at particular times in the task. In the example, the goal is set to WMTASK, and the visual input is set to a red question mark.

The *WM* (working memory) slot can hold a single item of information. Contrary to the other buffer slots, where information decays away if not fed by another process, the WM slot maintains its value until replaced. The three *Memory* slots represent a limited long-term declarative memory. An item can be placed in Memory1, after which an associative memory (Memory) finds the associate memory that is then placed in Memory2. In the example in the Figure, memory is used to determine that NINE is ODD. The *Action* slot is used to set the model’s action. In the Figure it is not connected to anything, but it should be connected to an appropriate motor system, comparable to what has been done in Spaun (Eliasmith et al., 2012). Finally, *PrevPRIM* refers to the previous step the system has executed, because this will be part of the input for determining the next step.

The model takes cognitive steps by transferring information between the slots. These steps are represented by cognitive operations that are basically quite simple: a symbol (semantic pointer) that represents the source and destination slots. For example, V1MEM1 means: copy the contents of Vision1 to Memory1. MEM2AC means: copy the contents of Memory2 to Action. The desired action is placed in the *PRIM* slot, after which the Basal Ganglia carries out that action. The Basal Ganglia follows the standard Nengo implementation,

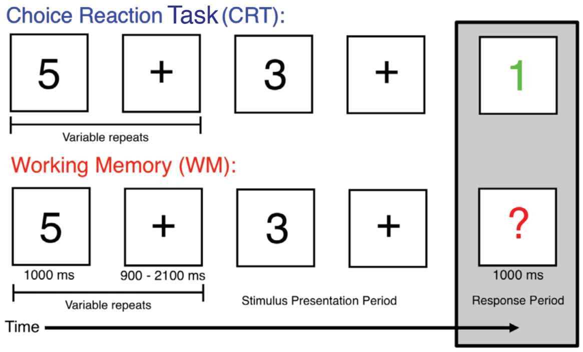


Figure 2: Design of the Smallwood et al. (2011) experiment.

and has a rule for each of the possible PRIMs.

Although the PRIMs architecture also has primitive operations to test conditions, the Nengo/PRIMs model will achieve this in a different way. The role of conditions is to determine, given the state of the system, what actions need to be carried out. Here we achieve this goal in a slightly different way: by learning a mapping between the contents of all the slots and the PRIM slot. We do this by combining all slots in a single semantic pointer:

$$\text{Combined} = \text{G} \odot \text{WMTASK} + \text{V1} \odot \text{Question} + \text{V2} \odot \text{RED} + \ldots$$

This combined semantic pointer is then mapped onto a PRIM semantic pointer.

Learning

The advantage of changing conditions into a more abstract mapping is that they can be learned instead of programmed. The current model uses supervised learning, which is why there is a *Correct PRIM* slot that is set by the input process. Whenever the model produces a PRIM on the basis of the combined state (initially random), that PRIM is compared to the Correct PRIM, after which the weights that map the combined state onto the PRIM slot are adjusted based on the error using prescribed error sensitivity (PES) learning (MacNeil & Eliasmith, 2011).

The design presented here is in principle task-general, although some parts need to be expanded for a fuller functionality (e.g., a more faithful Declarative Memory).

A Model of Mind Wandering

As an illustration of the model explained above, I will present a model of a task by Smallwood et al. (2011). In the experiment, subjects had to do two different tasks. In the Choice Reaction Task (CRT), subjects were presented with a sequence of digits that were interleaved with fixation crosses. Digits were presented for 1000ms, and the fixation cross for a variable duration between 900 and 2100ms. As long as the digits were black, no response was needed. After 2–5 black digits, a colored digit would appear, to which a response had to be made depending on whether the digit was odd or even. In the Working Memory task (WM), subjects were also presented with a sequence of 2–5 black digits, except that a colored question mark would appear instead of a colored digit. At that point subjects had to respond whether the last digit they saw was odd or even. Because subjects do not know when the question mark would appear, they had to remember the black digits. Occasionally, instead of the colored digit or question mark, subjects would be presented with a so-callled thought probe, to which they had to respond whether or not they were attending the task, or were thinking about something else. Smallwood et al. found that in the CRT, subjects were thinking about something else 68% of the time, whereas in the WM task they did so in 51% of the cases.

Models of the CRT and WM Task

In order to be able to do the tasks, the Basal Ganglia had to be prewired to carry out primitive actions. Primitive actions consisted of a source slot and a destination slot. For example, V1MEM1 would transfer the contents of the Vision1 slot to the Memory1 slot, and WMMEM1 would transfer the contents of the working memory slot to the Memory1 slot. For efficiency reasons, not all possible combinations were implemented, but a modest superset of the operations needed to do both tasks: V1MEM1, MEM2AC, V1WM, WMMEM1, MEM2WM, WMAC. A second function of the Basal Ganglia is related to learning, and was only active during learning: whenever a primitive action had completed its action, the learning signal would be suppressed. The reason is that we wanted to associate the operation with the state before the operation had been carried out, and did not want an association with the state after the operation (otherwise it would learn to repeat the operation).

A second piece of knowledge the network needs is which numbers are odd and which are even. An winner-takes-all associate memory was implemented in the Memory part of the model. Therefore, if a Semantic Pointer representing a number is placed in Memory1, ODD or EVEN would appear in Memory2.

The input node in the network feeds the input into the Vision slots of the network, and, during the training period, the correct PRIM into the Correct PRIM slot. The timing of the model is not yet completely consistent with the real experiment, but compressed in time, and restricted to just two black digits before the colored digit or question mark. Table 1 shows the schedule for what is presented by the input node to both Vision slots, and the correct PRIM operator that needs to be carried out at that point, which is send to the Correct PRIM slot to be used in the learning process.

The timing of the experiment is not consistent with human experiment, because many of the processes in Nengo are a lot faster in simulated time, but a lot slower in real time. Neither visual perception nor actions do take any time in this model, and memory retrieval is extremely fast. On the other hand, simulating a large model like this takes quite some real time, which means that for simulation purposes this is a reasonable

Table 1: Timing of the Inputs node. The V1 and V2 columns are fed into Vision1 and Vision2, respectively, and the CRT prim or WM prim is placed in the ”Correct PRIM” slot when learning is switched on.

t_{start}	t_{end}	V1	V2	CRT prim	WM prim
0.0	0.3	Digit	Black	FOCUS	V1WM
0.3	0.5	Blank	Blank	FOCUS	FOCUS
0.5	0.8	Digit	Black	FOCUS	V1WM
0.8	1.0	Blank	Blank	FOCUS	FOCUS
1.0	1.3	Digit or question	Red	V1MEM1	WMMEM1
1.3	1.6	Digit or question	Red	MEM2AC	MEM2AC
1.6	1.7	Blank	Blank	FOCUS	FOCUS

compromise.

The general idea in the CRT is that the model does not need to do anything until it sees a red digit. It then should execute V1MEM1 to determine whether the digit it sees is odd or even. After the memory has retrieved ODD or EVEN, it should execute MEM2AC to make the retrieved parity into an action. We are assuming here we have an action system that can interpret this as an action.

The WM model needs to do a bit more work: every time a black digit is presented, it should store that digit in working memory with a V1WM action. Once it sees a colored question mark, it should transfer the item from working memory into the memory retrieval system: WMMEM1. Similarly to the CRT, the result of that retrieval should be transferred to the action slot, MEM2AC. Whenever the model does not need to do anything, the table shows FOCUS. This is placed in the PRIM slot, but there is no rule in the Basal Ganglia to carry it out (because it doesn't do anything). However, the Basal Ganglia are not just waiting, but carries out a ”default” action, which will be the basis for Mind Wandering.

Modeling Mind Wandering itself

As has become clear in the previous section, the Basal Ganglia are not always engaged in task-related actions. To model mind wandering, we added a default action to the Basal Ganglia that initiates and perpetuates mind wandering as long as it does not receive a specific instruction from the PRIM slot. This option is more or less standard in the Nengo Basal Ganglia model, because you have to specify a default action for it to do if no other action is sufficiently supported.

The idea is, following some existing symbolic models (Taatgen et al., submitted), that Mind Wandering consists of a train of thought simulated by a sequence of declarative retrievals. To mimic this in a simple way, we added a number of extra associations to the memory that also produces the mapping between numbers and parity. More specifically, we added that EPISODE maps onto CRY, CRY maps onto REDEEM, and REDEEM maps onto LAUGH. The default Basal Ganglia action is to feed EPISODE to Memory1, and also copy the contents of Memory2 into Memory1. This means that if there are no active PRIMs (either because it is set to FOCUS, or when there is no specific PRIM active), EPISODE is placed in Memory1, which will in turn lead to the retrieval of CRY, which is fed back into Memory1 leading to the retrieval of REDEEM, etc.

Training

Training consisted of 40 learning blocks, each with a CRT trial and a WM trial. A trial lasted 1.7 simulated seconds following Table 1. After 40 trials the training input was blocked, after which and additional 20 blocks were simulated and used to determine the results.

Results

The critical mapping that the model needs to learn is between the combined state of the system and the PRIM to be executed. Figure 3 shows the input to the Basal Ganglia, which represents the strength of each of the PRIMs in the PRIM slot. The graphs shows the average of the 20 performance trials after learning. On the left side of the graph the WM task is shown, where the V1WM prim becomes active whenever there is a black digit. During the short periods between the digits, there is no PRIM that is active enough to exceed the 0.3 threshold, which means that the model will initiate Mind Wandering during this (very brief) period. When the red question mark is presented, the WMMEM1 PRIM is activated, transferring the contents of working memory to a memory retrieval. When the answer has been retrieved from memory, the MEM2AC PRIM is activated to transfer the retrieval to the action slot. The interesting aspect of last action is that the PRIM becomes active earlier than during training (approximately at 1.2 seconds instead of 1.3 seconds), which indicates that the learning has made sure that the PRIM has been keyed to a successful retrieval.

For the CRT we can see that the model does nothing when black digits are presented, even though the V1WM PRIM becomes active, but at a subthreshold level (indicating some transfer from the WM task). When the red digit comes up, the V1MEM1 PRIM becomes active, initiating the memory retrieval and subsequently the MEM2AC PRIM. It is clear that in the CRT the model has much more opportunity to mind wander. This can be seen slightly more clearly in the Thalamus output graph (Figure 4), where a winner-takes-all competition has produced a winning action in each of the stages.

To get an impression of how much Mind Wandering these decisions produce, we need to look at the activity in Memory. Figure 5 shows the activity of various memory items in a sample trial, measured in the Memory2 slot. We can see mind wandering by the activation of the CRY, REDEEM and LAUGH semantic pointers, while task-related activity consists of activation of ODD and EVEN. Obviously, there is a lot more Mind Wandering going on than the Basal Ganglia results suggest. The reason is that after the Basal Ganglia initiates Mind Wandering, it can dominate the activity in the

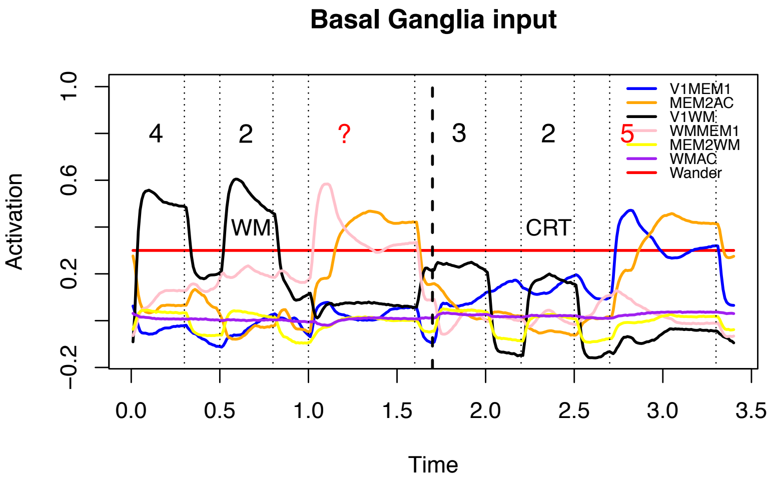


Figure 3: Input to the Basal Ganglia, showing the activation of each of the PRIMs. The WM task is between time 0 and 1.7, the CRT between 1.7 and 2.4. Representative stimuli that are presented to the model are displayed at the top of the Figure. The red horizontal line is the activation of the Wander action: this is not a real activation, but a default action if none of the PRIMs exceeds the 0.3 threshold.

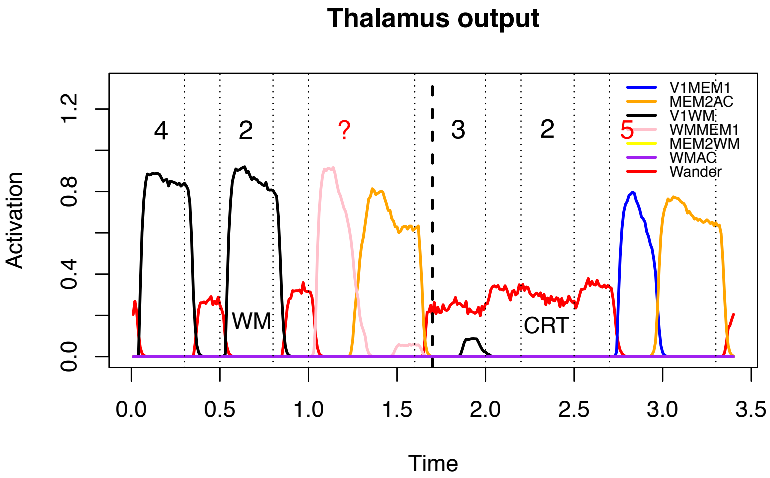


Figure 4: In the output of the Thalamus we can see which action is selected, which is the highest value of the input.

memory system for a while as long as it is not needed by the task (following the threaded cognition multitasking theory, Salvucci & Taatgen, 2008). Nevertheless, in the CRT Mind Wandering is supported by the Basal Ganglia for a much longer period, which is reflected in more memory activity.

If we calculate the proportion of Mind Wandering over all the model output (after training), we see that the Memory output matches the data most closely (Figure 6). We have to take these results with a grain of salt, though, because the timing of the experiment does not match the real experiment.

Discussion

The main purpose of this work was to demonstrate that sequential tasks can be learned by a spiking neural network following principles derived from symbolic architectures. In this model it is no longer necessary to store all procedural knowledge in the Basal Ganglia, but is stored in an associative memory that can be located elsewhere, probably in the prefrontal cortex (Cole, Bagic, Kass, & Schneider, 2010). A key difference with regular production models (and also Spaun), is that it does no test conditions explicitly, but instead learns a mapping between the cognitive system's state and the action to be performed. This has two advantages: sequential matching of production rules in a neural network is cumbersome. In order to do this in parallel, production rules already need to be hard-wired in such models, which makes flexibility a greater challenge. The second advantage is that it is much easier to learn new productions.

Still, there is a lot of work to be done. The actions this model can make are elementary PRIMs. However, in the full PRIM theory, elementary PRIMs cluster together into general purpose operators. The most probable place for this kind of

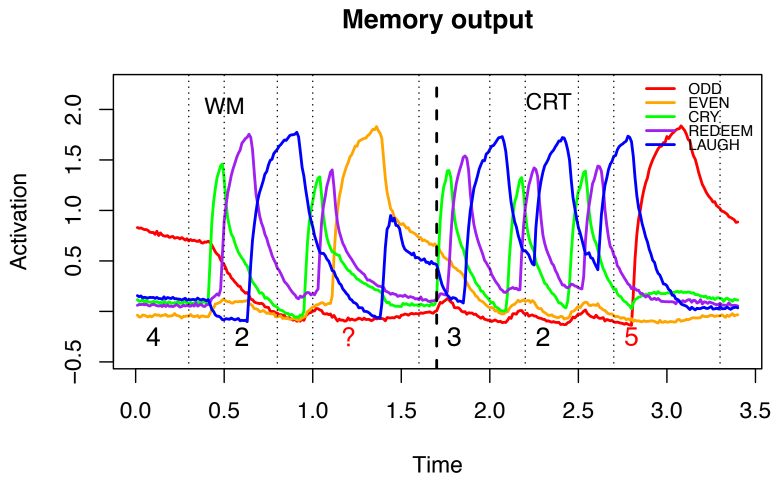


Figure 5: The contents of Memory2 during the experiment, showing which of the facts in Memory is most active.

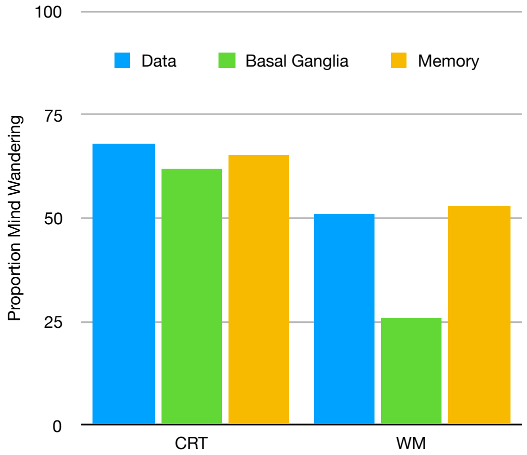


Figure 6: Proportion Mind Wandering in the data, in the model's Basal Ganglia action output, and in the model's Memory output.

learning are the Basal Ganglia. Moreover, we used supervised learning in this model. It is unclear where such a learning input would come from, and therefore a form of reinforcement learning is a better alternative.

The model's mind wandering is a nice demonstration (also showing the model can fit some data), but the Mind Wandering itself is now modeled as a "default strategy". Instead, it should also be modeled using primitive operations that compete with task-related operators.

Acknowledgments

I wish to thank Terry Stewart, Sean Aubin, Alexander Serb, and Sverrir Thorgeirsson for their help and good discussions for this project.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford university press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036–1060.

Cole, M. W., Bagic, A., Kass, R., & Schneider, W. (2010). Prefrontal Dynamics Underlying Rapid Instructed Task Learning Reverse with Practice. *Journal of Neuroscience*, *30*(42), 14245–14254.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., ... Rasmussen, D. (2012, nov). A large-scale model of the functioning brain. *Science*, *338*(6111), 1202–5. doi: 10.1126/science.1225266

MacNeil, D., & Eliasmith, C. (2011). Fine-tuning and the stability of recurrent neural networks. *PloS one*, *6*(9), e22885.

O'Reilly, R. C., & Frank, M. J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, *18*(2), 283–328.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, *115*(1), 101–130.

Smallwood, J., Schooler, J. W., Turk, D. J., Cunningham, S. J., Burns, P., & Macrae, C. N. (2011, dec). Self-reflection and the temporal focus of the wandering mind. *Consciousness and Cognition*, *20*(4), 1120–1126. doi: 10.1016/j.concog.2010.12.017

Stocco, A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological Review*, *117*(2), 541–574.

Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, *120*, 439–471.

Taatgen, N. A., van Vugt, M. K., Daamen, J., Katidioti, I., Huijser, S., & Borst, J. P. (submitted). The resource-availability theory of distraction and mind-wandering.

Extending JSegMan to Interact with a Biased Coin Task and a Spreadsheet Task

Farnaz Tehranchi (farnaz.tehranchi@psu.edu)
Department of Computer Science and Engineering
Frank E. Ritter (frank.ritter@psu.edu)
College of Information Sciences and Technology
Penn State, University Park, PA 16802 USA

Keywords: Cognitive model; Cognitive architecture; Human-computer interaction; Simulated eyes and hands.

Introduction

To act more like a human, current cognitive models require a way to see and operate in the world. This methodology and access to a world as a requirement were discussed before, and several ways have been created. For instance, cognitive model interface management systems (CMIMS) based on user interface management systems (UIMS) were introduced to provide cognitive models with eyes and hands to interact with the same interface that users see (Ritter, Baxter, Jones, & Young, 2001).

We report on developments extending JSegMan, which supports architectures to interact with uninstrumented environments (Tehranchi & Ritter, 2017). JSegMan was improved, and new features for the visual module were added. JSegMan creates a way to interact with all interfaces using an extended Java library (Robot package) to input motor commands (keystrokes, mouse moves, and mouse clicks), and uses an open source library to help with image processing (Sikuli) based on *OpenCV*. JSegMan introduced visual patterns that are small images that represent the visual objects in cognitive architectures—visual chunks in ACT-R. JSegMan parses the screen and uses the *Template Matching* method to find the target, the visual pattern, and area. Template Matching is a pattern-matching algorithm that compares a template (small image) against the overlapped image regions (the computer screen) pixel by pixel; the area that has the maximum matching score is the target area. JSegMan can identify pre-defined patterns. Visual patterns are defined for the cognitive model similar to memory chunks (Tehranchi & Ritter, 2018).

JSegMan has been used with previous models. In all applications, JSegMan provides ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004) cognitive models with eyes and hands interaction. After describing it, we report here two new interfaces it uses.

JSegMan and the Dismal Task

JSegMan (Tehranchi & Ritter, 2018) has been used along with the Dismal spreadsheet task in Emacs using an existing large ACT-R model (Paik, Kim, Ritter, & Reitter, 2015). It has 29 rules and 1,152 declarative memory task elements. JSegMan illustrates the missing knowledge in the original model because with JSegMan the effect of actions on the interface are visible and are trackable. For instance, we found one missing click in the original model. JSegMan allows the model not just to model the task performance but actually to perform the task. It conducts a large, 20 min., non-iterated task with 14 subtasks. Also, most of the key press requests to the motor module required a hand/finger adjustment. These differences between the task requirements and the model's performance were visible because the results in the interface did not match the expected output. Matching behavior in the target interface can be an essential way to validate models in the future. We adjusted 162 declarative chunks in the original Dismal ACT-R model by adding a new slot for visual objects. Additionally, to model eye movements, we added 52 new visual objects and visual locations. The use of JSegMan also provided a better fit to the human data. The model with JSegMan predicted the response times more accurately while, importantly, using the same, unmodified interface that the human subjects used. The correlation improvement is not a reliable increase, but the difference in MSE was reduced by 47% (Tehranchi & Ritter, 2018).

JSegMan and the Biased Coin

The biased coin model is based on learning in a probability choice experiment in the ACT-R tutorial (Bothell, 2017). Figure 1 shows the experiment window implemented in ASP.Net. After seeing "Ready," the model either clicks on the "Head" button for heads or the "Tail" button for tails. The feedback indicating the correct answer, either "Head" or "Tail" is displayed; "Match" is displayed in green, and for incorrect answers "Wrong" in red is displayed. Figure 2a. illustrates a summary of the model. In this experiment, heads are the correct choice on 70% of the trials without considering the previous user choices. The ACT-R model begins with a 50% chance of heads and tails. Figure 2b shows its first choice was a tail. The model adjusts the head and tail probability based on what the model, with JSegMan eyes, can see on screen through utility learning. Finally, after 100 trials the model average responding for heads approaches 70%.

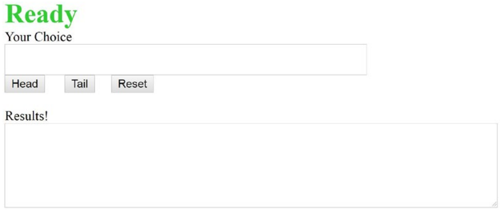


Figure 1. The biased coin interface used by JSegMan.

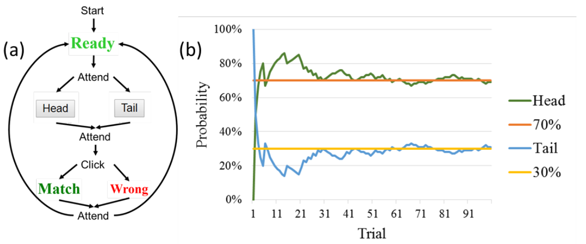


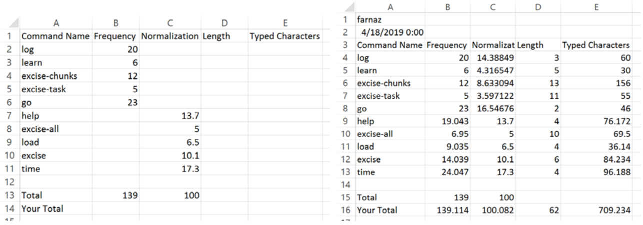
Figure 2. (a) The simplified flowchart used by the ACT-R model and patterns used by JSegMan. (b) The probability of choosing the head, green, and the tail, blue, over 100 trials.

We learned that due to the bidirectional communication what the model can see can affect the learning procedure.

JSegMan and the Excel Spreadsheet Task

We have started to extend JSegMan to interact with an Excel spreadsheet, shown in Figure 3, to perform the Dismal task, shown in Table 1 (similar to Paik et al., 2015). When compared to the existing dismal model of the task in an Emacs spreadsheet, this model will help explore the effect of the interface on task and requires JSegMan to be extended. We have (a) added a marker showing the area of attention's location (shown in Figure 4), (b) provided bidirectional communication between JSegMan and ACT-R model, and (c) reduced further limitations in cognitive modeling (e.g., models can directly interact with the same environment as a user and made it more realistic).

It remains to gather data on this version, collect more eye movement data to simulate attention shifts more accurately, model more of the task, and thus better predict human performance. Also, with JSegMan we can show whether the eyes follow the hands successfully.



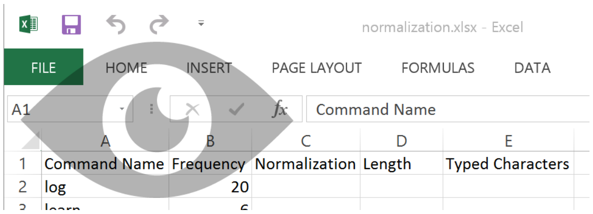
	A	B	C	D	E
1	Command Name	Frequency	Normalization	Length	Typed Characters
2	log	20			
3	learn	6			
4	excise-chunks	12			
5	excise-task	5			
6	go	23			
7	help		13.7		
8	excise-all		5		
9	load		6.5		
10	excise		10.1		
11	time		17.3		
12					
13	Total	139	100		
14	Your Total				

	A	B	C	D	E
1	farnaz				
2	4/18/2019 0:00				
3	Command Name	Frequency	Normalizat	Length	Typed Characters
4	log	20	14.38849	3	60
5	learn	6	4.316547	5	30
6	excise-chunks	12	8.633094	13	156
7	excise-task	5	3.597122	11	55
8	go	23	16.54676	2	46
9	help	19.043	13.7	4	76.172
10	excise-all	6.95	5	10	69.5
11	load	9.035	6.5	4	36.14
12	excise	14.039	10.1	6	84.234
13	time	24.047	17.3	4	96.188
14					
15	Total	139	100		
16	Your Total	139.114	100.082	62	709.234

Figure 3. The dismal task starts, on the left side, and on the right side, the final task state is in Excel.

Table 1. The dismal task subtasks.

Tasks
(1) Open File
(2) Save As
(3) Calculate Frequency (B7 To B11)
(4) Calculate Total Frequency (B14)
(5) Calculate Normalization (C2 To C6)
(6) Calculate Total Normalization (C14)
(7) Calculate Length (D2 To D11)
(8) Calculate Total Length (D14)
(9) Calculate Typed Characters (E2 To E11)
(10) Calculate Total Typed Char. (E14)
(11) Insert Two Rows
(12) Type In Name (A1)
(13) Insert Current Date (A2)
(14) Save As …



Command Name

	A	B	C	D	E
1	Command Name	Frequency	Normalization	Length	Typed Characters
2	log	20			
3	learn	6			

Figure 4. JSegMan's area of attention (the pupil) on an Excel spreadsheet.

Conclusion and Further Research

These interactive models and agents are essential for cognitive science and also important for agent-based modeling and computational organizational theories because they provide social agents that can interact with the world.

JSegMan is about adding the capabilities to model the details of visual, and motor modules for cognitive architectures, and what they can teach us about how the human mind works, how an analysis of vision and motor as they support and implement cognition might further our understanding of the human mind.

Acknowledgments

This work was partially funded by ONR (N00014-15-1-2275). We thank Dan Bothell for his assistance with the ACT-R tutorial.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Bothell, D. (2017). ACT-R 7 Reference Manual [Electronic Version] from act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr7/reference-manual.pdf.

Paik, J., Kim, J. W., Ritter, F. E., & Reitter, D. (2015). Predicting user performance and learning in human-computer interaction with the Herbal compiler. *ACM Transactions on Computer-Human Interaction, 22*(5), 25.

Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2001). User interface evaluation: How cognitive models can help. In J. Carroll (Ed.), *Human-Computer Interaction in the New Millenium* (pp. 125-147). Reading, MA: Addison-Wesley.

Tehranchi, F., & Ritter, F. E. (2017). An eyes and hands model for cognitive architectures to interact with user interfaces. In *MAICS, The 28th Modern Artificial Intelligence and Cognitive Science Conference*, 15-20. Fort Wayne, IN: Purdue University.

Tehranchi, F., & Ritter, F. E. (2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. In *Proceedings of ICCM - 2018-16th International Conference on Cognitive Modeling*, 162-167. University of Wisconsin, Madison, WI.

Combining Mental Models and Probabilities: A new Computational Cognitive Approach for Conditional Reasoning

Sara Todorovikj (sara.todorovik@gmail.com)
Paulina Friemann (friemanp@cs.uni-freiburg.de)
Marco Ragni (ragni@cs.uni-freiburg.de)
Cognitive Computation Lab, Technical Faculty, University of Freiburg, 79110 Freiburg, Germany

Abstract

Recent psychological experiments on conditional reasoning indicate the relevance of content, background knowledge and form on the sort of an individual's inference. Based on two of the most prominent theories, probabilistic and mental model based approaches, we develop a probabilistic mental model theory based on Pearl's ε-semantic. By modeling subjective belief in possible worlds, influenced by form and content of a conditional, our model is able to express numerically an individuals' degree of belief in a conditional, while providing an explainable semantics applicable to other domains.

Keywords: Probabilistic Cognitive Models; Reasoning; Conditionals; Predictive Modeling

Introduction

A core goal of cognitive science is to develop a unified theory of cognition. Johnson-Laird's Mental Model Theory (MMT, Johnson-Laird, 1983) is a key theory in the area of human reasoning, and a good candidate for offering a unified theory for a broad range of domains. Its intuitive, comprehensible form allows it to be applied to many domains within reasoning, and its algorithmic and predictive nature enables a qualitative evaluation on explored as well as novel tasks. The MMT assumes the creation and transformation of mental models to describe the scenario.

Another core unified theory for human reasoning are probabilistic approaches. The core assumption is that uncertain reasoning is the basis for rationality, rather than certain reasoning (e.g., Oaksford, Chater, & Larkin, 2000), which leads to the development of probabilistic reasoning models. These have been applied to a plenitude of reasoning domains (e.g., Elqayam & Over, 2013).

The advantages of probabilistic approaches to modeling human reasoning are, among others, the inherent ability to handle uncertain knowledge and the possible incorporation of a subjective degree of belief (e.g., Elqayam & Over, 2013).

In this paper, we propose a combination of the two approaches. We take the Mental Model Theory and extend it with a probabilistic account of the mental models using Pearl's ε-semantics (Pearl, 1991). We interpret the mental models in a conditional task, reported by Singmann, Klauer, and Beller (2016), as possible worlds in the sense of modal logic, and calculate their relative probabilities. One of the factors which define human reasoning is the type of content and the presentation form. More specifically, the individual's background knowledge, and the presentation form of the premises that participants are presented with, can heavily influence the drawn conclusions (e.g., Singmann et al., 2016). To account for this, we extend our model to give predictions for the different contents and presentation forms used in Singmann et al. (2016). The paper is structured as follows: First, we give a short introduction to conditionals, followed by two probabilistic approaches to conditional reasoning. Afterwards we introduce the ε-semantics and our cognitive model for the conditional reasoning task. Finally, we analyze the model's performance on the empirical data and compare it to the model in Singmann et al. (2016).

Reasoning with conditionals

Conditionals are statements of the form 'If p then q' (also written as $p \rightarrow q$), where p is called the antecedent, and q, the consequent. Given a conditional rule, i.e., 'if p then q', (also called a major premise) and a minor premise that describes the current situation, for example 'p is true' (given as p), individuals are asked to infer a conclusion. If an individual is given a conditional '$p \rightarrow q$' and a minor premise p, and they infer q, they followed the modus ponens inference form. If instead they were given the minor premise '$\neg q$' ('q is false'), and they conclude '$\neg p$', they followed the modus tollens inference form. There are four inference forms: *modus ponens* (MP), *modus tollens* (MT), *affirming the consequent* (AC), and *denying the antecedent* (DA), as shown in Table 1.

Table 1: The four inference forms.

MP	AC	DA	MT
$p \rightarrow q$	$p \rightarrow q$	$p \rightarrow q$	$p \rightarrow q$
$\underline{p\quad}$	$\underline{q\quad}$	$\underline{\neg p\quad}$	$\underline{\neg q\quad}$
q	p	$\neg q$	$\neg p$

When interpreting conditionals as causal relationships, in the real world we encounter so-called *disablers* and *alternatives*. Disablers are events that prevent q from happening, even if p has occurred, and alternatives describe events that enable q to happen, even if p has not, e.g.:

> If a balloon is pricked with a needle then it will pop.
> Disabler: The balloon was not inflated at all.
> Alternative: The balloon was pricked with a pen.

Data

The experiment we modeled in this paper, experiment 1 from Singmann et al. (2016), tested participant's endorsements for each of the four inference forms, depending on relative amount of disablers and alternatives, and the form of presentation. In the experiment, participants were asked to give an estimate of the probability, between 0% and 100%, for the different types of problems. The disablers and alternatives are expected to influence the estimates given by the participants. The second independent variable was the form: participants were given either no major premise (*reduced inference*), the major premise in form of a conditional, or the major premise as a *biconditional* (i.e., 'if and only if p, then q.'). In all three cases they were given a minor premise and a conclusion whose probability they were supposed to rate. The tasks used in the experiment are presented in Table 3[1].

Dual-Source model

Oaksford et al. (2000) proposed a probabilistic interpretation of conditional rules by using the probabilities of the antecedent ($a = P(p)$), consequent ($b = P(q)$), and exception ($\varepsilon = P(\neg q|p)$). Singmann et al. (2016) extended this model by disentangling the logical form and content of a conditional, by contrasting individuals' responses to regular conditional inferences, and, reduced inferences (which omit the conditional, and present only a minor premise). They are using three types of parameters: $\xi(C,x)$ (knowledge-based component, depending on the content C and inference x), $\tau(x)$ (form-based component reflecting the subjective degree of belief in the inference x), and λ (a weight given to the form-based knowledge). Endorsement of the reduced inference x with content C is expressed through the knowledge-based component, as shown in Eq. 1, and endorsement of the full inference x with content C is shown in Eq. 2.

$$E_r(C,x) = \xi(C,x) \tag{1}$$

$$E_f(C,x) = \lambda\{\tau(x) + (1-\tau(x)) \cdot \xi(C,x)\} + (1-\lambda)\xi(C,x) \tag{2}$$

ε-semantics

As described by Pearl (1991), ε-semantics is a 'formal framework for belief revision', where belief statements are statements of high probability, and belief revision conditions current beliefs based on new evidence. Simply put, we have a probability function P, which is defined over a set of possible world states, W. A probability $P(w)$ is assigned to each world state w as a polynomial function of some small, positive parameter ε. ε-semantics distinguishes between sentences that describe truths and general tendencies (e.g. 'Birds fly.'), and sentences that describe findings or observations in a specific situation (e.g. 'All blocks on this table are green.'). This is reflected in natural language when using the word 'If' (Pearl, 1991). A statement like 'If it's a bird, it flies' is reasonable, while 'If this block were on this table it would be green.' is not. In order to lay basis for our reasoning model, we will take into consideration the following definition according to Pearl (1991, p. 5):

> Let L be a language of propositional formulas, and let a truth-valuation for L be a function t, such that t maps the sentences in L to the set $\{0, 1\}$ (0 - 'false', 1 - 'true'). A probability assignment $P(w)$ is defined over the sentences in L, where each truth valuation t is regarded as a world w, and $\sum_w P(w) = 1$. This way a probability measure is assigned to each sentence l of L.

Model

In our model we aim to define worlds described by conditional rules following the definition given above. Given a conditional 'If p then q', we take into consideration all the possible worlds, i.e. all the combinations of truth-values for p and q, as shown in Table 2. As stated in the definition, we have a probability distribution P defined over all worlds, assigning a probability value p_i to each one of them.

Table 2: The possible worlds described by 'If p then q', the probability distribution P and probability values p_i, $1 \leq i \leq 4$.

p	q	P
0	0	p_1
0	1	p_2
1	0	p_3
1	1	p_4

For example, in the case of the conditional "If it is a bird, then it flies", the probability value assigned to the world where it is a bird and it is not flying ($p = 1, q = 0$) is p_3.

$$P(\beta|\alpha) = \frac{P(\alpha \wedge \beta)}{P(\alpha)} \tag{3}$$

As mentioned earlier, individuals are asked questions of the form 'Given p, how likely is it that q?', which is actually conditional probability, in this case noted as $P(q|p)$. Following the standard definition of conditional probability (Eq. 3), we obtain the four equations shown below, which describe the four inference forms using the probability distribution P of the conditional's worlds (Table 2):

$$\textbf{MP: } P(q|p) = \frac{p_4}{p_3 + p_4} \qquad \textbf{DA: } P(\neg q|\neg p) = \frac{p_1}{p_1 + p_2}$$

$$\textbf{AC: } P(p|q) = \frac{p_4}{p_4 + p_2} \qquad \textbf{MT: } P(\neg p|\neg q) = \frac{p_1}{p_1 + p_3}$$

Due to individual differences between reasoners, and a divergent background knowledge, it follows that every individual would have a different probability assignment for a certain world. Using these four equations, we can model each participant individually, and determine their personal probability

[1] We would like to note that the choice for the conditional content in the 'Girl' case can be thought of as slightly controversial, which unfortunately leads to some inconsistencies and difficulties when trying to model and/or analyze the data.

Table 3: Contents used in Singmann et al. (2016) experiments.

Keyword	Content	Disablers	Alternatives
Predator	If a predator is hungry then it will search for prey.	Few	Few
Balloon	If a balloon is pricked with a needle then it will pop.	Few	Many
Girl	If a girl has sexual intercourse then she will be pregnant	Many	Few
Coke	If a person drinks a lot of coke then the person will gain weight.	Many	Many

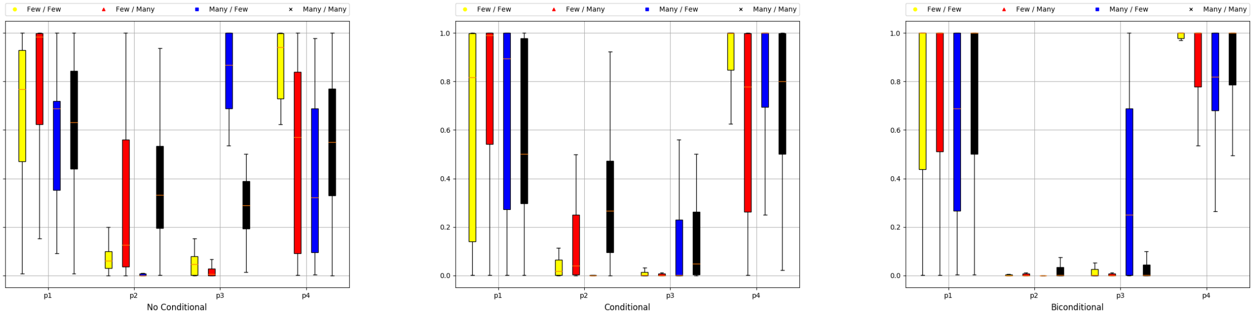


Figure 1: Box plots depicting individual world probability values p_i for every task. Probability values between 0 and 1 – no conditional (reduced inference) (left), conditional (center), and biconditional (right) case. Labels show the amount of disablers and alternatives for the task, e.g.: Few/Many → Few disablers / Many alternatives.

distribution, by taking the probability values p_1, p_2, p_3 and p_4 as our parameters and fitting them to their endorsements of the inference forms for every task. Since the parameters are bounded by the sum ($\sum_i p_i = 1$), we have only three free parameters in our model.

Parameters

The parameters in our model, p_1, p_2, p_3, p_4, describe the probability values an individual assigns to the possible worlds described by the conditional. In this section we examine our parameters in more depth and we aim to show that our model can account for the effect of disablers and alternatives on conditional reasoning, and, also, for the effect of individuals being presented with a reduced inference or a (bi)conditional.

Interpretation. Following Table 2, we will first focus on p_2 and p_3. Through these two parameters the effect of disablers and alternatives can be shown. p_2 is the probability that even though p happened, q did not, which is interpreted as the outcome of a disabler preventing q from happening. p_3, on the other hand, is the probability of the world when even though p did not happen, q did, which is interpreted as the outcome of an alternative enabling q to happen. The effect of the different presentation forms can be shown through p_4 and p_1. p_4 is the probability that both p and q happened, can be interpreted as an individual's degree of belief in the conditional rule. p_1 is the probability of the world where neither p, nor q happens, which can show an individual's belief in a rule as a biconditional (if and only if).

Hypotheses. We have two types of hypotheses about changes in the parameter values: First, tasks with few alternatives in contrast to tasks with many alternatives (and the same amount of disablers), may result in an increase in p_2. Similarly, we expect p_3 to be higher for tasks with many disablers in contrast to tasks with few disablers (for the same amount of alternatives). So, both p_2 and p_3 increase when comparing a task with few disablers and alternatives with a task with many disablers and alternatives. Second, the belief in the task's rule increases (and so does p_4) when individuals receive a (bi)conditional in contrast to a reduced inference. We expect an increase of p_1 in the biconditional case. In the reduced inference case, a belief that 'if p then q' is not present, which may lead to a high p_1 in some cases, as an expression of the lack of belief in the influence of disablers/alternatives. In the conditional case, the conditional still might be interpreted as a biconditional (e.g. Cummins, Lubart, Alksnis, & Rist, 1991). So, the change for p_1 depends on the type of conditional.

Fitting. The fitting of our model to the data was done using Python's `scipy.optimize.minimize` function, by minimizing the RMSE with the L-BFGS-B bound-constrained minimization method[2].

Figure 1 shows the individual parameter fit for every task in all three conditions. The first observation that we can make, when looking at the plots, is that the values of p_2 and p_3 are generally higher in the reduced inference case, compared to the conditional and biconditional case, where p_4 and p_1 are higher, as we assumed in our hypothesis. This also leads us to believe that individuals' reasoning is more 'logical' when they are presented with strict conditional rules, compared

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`

Table 4: Mean percentages of the individuals' values for p_1 and p_4 for every task in each condition (reduced inference, conditional, biconditional); Means of the differences between individuals' values for p_1 and p_4 for reduced inference - conditional, and conditional - biconditional, for every task. (D/A - Disablers/Alternatives; F - Few; M - Many; Red. Inf. - Reduced Inference; Cond. - Conditional; Bicond. - Biconditional)

Task	p_i	Red. Inf.	Cond.	Bicond.	Red. vs. Cond.		Cond. vs. Bicond.	
D/A		Mean	Mean	Mean	Mean	P-value	Mean	P-value
FF	p_1	64.60	60.60	75.20	-4.00	.869	14.60	**.039**
	p_4	79.20	87.40	87.20	8.20	.072	-.30	.715
FM	p_1	74.40	71.40	73.30	-2.90	.981	1.90	.903
	p_4	50.90	64.20	81.70	13.30	.159	17.40	.079
MF	p_1	58.60	66.50	63.70	7.90	.304	-2.80	.408
	p_4	39.20	81.50	71.30	42.30	$<$ **.001**	-10.20	.229
MM	p_1	61.80	58.20	73.50	-3.50	.688	15.30	.131
	p_4	54.50	71.50	81.70	17.00	**.004**	10.10	.082

to when they have to completely lean on their background knowledge, and be more creative when thinking about possible disablers and alternatives. Our next observation is about the change in values of p_2 and p_3 between tasks with varying amounts of disablers and alternatives. As we assumed in our hypothesis, it can be seen that the tasks with many alternatives have higher values for p_2, compared to the other tasks, and the tasks with many disablers have higher values for p_3.

Results and Discussion

Influence of disablers and alternatives. Our first hypothesis was about the influence of disablers and alternatives on p_2 and p_3. We first calculated the means of all individuals' values for p_2 and p_3, for each task, which are shown in Table 5, for the reduced inference case, since that is the case in which we can observe the application of individuals' background knowledge purely without having the influence of the (bi)conditional. We can immediately see that p_2 and p_3's values have a higher mean in the presence of many alternatives, or disablers, respectively.

Afterwards, we looked into pairs of tasks that differ in the amounts of disablers and alternatives, and how the p_2 and p_3 values change between them. Table 6 shows the means of the differences between the all p_i values of the pairs of tasks. In order to determine the statistical significance of the change of the probability values between tasks we performed the Wilcoxon signed-rank test on them, using Python's `scipy.stats.wilcoxon` method[3]. The analysis confirms our hypothesis that when increasing the amount of alternatives, the value of p_2 increases, and when increasing the amount of disablers, the value of p_3 increases.

Table 5: The mean percentages of the values for p_2 and p_3 for every task in the reduced inference case.

Task		p_i	Reduced Inference
Disablers	Alternatives		Mean
Few	Few	p_2	8.70
		p_3	6.00
Few	Many	p_2	28.30
		p_3	5.10
Many	Few	p_2	9.00
		p_3	82.00
Many	Many	p_2	37.50
		p_3	32.10

Influence of a (bi)conditional. Our second hypothesis was about the influence of giving a (bi)conditional on p_1 and p_4. We first calculated the means of all individuals' values for p_1 and p_4, for each task, and every condition, which are shown in Table 4. The p_4 values in the (bi)conditional case are larger compared to the reduced inference case, as expected. However, the changes in the p_1 values, are not uniform.

Comparison

After fitting our model to the data, we compared its fit to the Dual-Source model (DSM) on the same data. We obtained the DSM parameter values for the participants from `https://osf.io/zcdfq/`, and used them accordingly in equations 1 and 2, as described above. The DSM uses 22 parameters to fit all four tasks for all conditions[4], and our model uses 3 parameters, but is fitted respectively to all problems. Hence, we can determine changes in the p_i across tasks. To determine the goodness of fit we calculated the RMSE and R^2. The mean RMSE for our model was .020, and R^2 was .963, compared to DSM's mean RMSE of .049, and R^2 of .815. It should be noted that the DSM has certain limitations – the $\xi(C,x)$ parameter values can only be obtained when fitting the reduced

[3] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html`

[4] 16 values for $\xi(C,x)$, 2 values for λ, and 4 values for $\tau(x)$

inference case, so if the model was presented only with, e.g., the conditional case it would not be possible to fit it.

Table 6: Means of percentages of the differences Δp_i between individuals' values for all p_i for combinations of tasks in the reduced inference case, calculated as p_i(Task 1) - p_i(Task 2). (D/A - Disablers/Alternatives; F - Few; M - Many)

Task 1	Task 2	p_i	Reduced Inference	
D/A	D/A		Mean Δ	P-value
F/F	F/M	p_1	9.78	.178
		p_2	**19.60**	**.001**
		p_3	**-9.00**	**.016**
		p_4	**-28.35**	**.004**
F/F	M/F	p_1	-5.97	.347
		p_2	**-7.70**	< **.001**
		p_3	**76.00**	< **.001**
		p_4	**-40.04**	<**.001**
F/M	M/M	p_1	**-12.57**	**.036**
		p_2	9.20	.063
		p_3	**27.00**	< **.001**
		p_4	3.65	.739
M/F	M/M	p_1	3.17	.769
		p_2	**36.60**	< **.001**
		p_3	**-49.80**	< **.001**
		p_4	**15.34**	**.011**
F/F	M/M	p_1	-2.80	.750
		p_2	**28.80**	< **.001**
		p_3	**26.10**	< **.001**
		p_4	**-24.70**	**.002**

Reducing the number of parameters

By fitting our 3 parameters to each task, we achieved a good fit, and our next goal will be to try to reduce the fitting to each task, while still obtaining satisfactory results, by challenging the predictive capabilities of our model. We tackle this by taking into consideration the earlier observations of how the probability distributions change between tasks and also between conditions, which leads to two different approaches. In the following, where necessary, a probability value will be denoted in the following way: $p_i^x(t)$, where t is the first letter of the task's keyword (*p, b, g, c*), x is the condition (r for reduced inference, c for conditional, and b for biconditional), and i is, as before, the probability value's index ($i \in [1,2,3,4]$). So, for example $p_3^r(g)$ denotes the p_3 value for the 'girl' task in the reduced inference case. To measure the goodness of prediction for every approach we calculated the RMSE[5].

Task probability distribution differences. The first approach focuses on how probability values change between tasks, especially based on the different number of disablers and alternatives. This can be done in two different ways which differ in the number of parameters used.

Constant differences. When we fit the model initially we calculate the differences of the probability values between tasks, for the reduced inference, conditional, and biconditional case. By taking the means of those differences we obtain constant values which describe the general change of the probabilities among participants. For example, Table 7 shows the constants for probability value changes from the 'Predator' task to the other three tasks in the conditional case.

Table 7: Constants for probability value changes Δp_i between tasks for the conditional case. Values between 0 and 1. D/A - Disablers/Alternatives; F - Few; M - Many

Task 1 (D/A)	Task 2 (D/A)	Δp_1	Δp_2	Δp_3	Δp_4
F/F	F/M	.109	.113	.010	-.232
F/F	M/F	.06	-.035	.141	-.059
F/F	M/M	-.023	.237	.113	-.159

Now, we fit the participant's endorsements for one task, for which we need only three parameters, and predict the endorsements for the other tasks by using the constants, as shown in Eq. 4, where fit is the task we have already fitted, and *pred* is the task whose endorsements we predict.

$$p_i^x(pred) = p_i^x(fit) - const(fit, pred) \tag{4}$$

E.g., if we have fitted the probability values for the 'Predator' task in the conditional case, and we want to predict the p_2 value for the 'Balloon' task, we will calculate it by:

$$p_2^c(b) = p_2^c(p) - 0.113 \tag{5}$$

The RMSE values for this approach can be found in Table 8. Using the mean of the differences encourages an assumption that no matter how different individuals are, and how diverse their background knowledge is, there are still some similarities in their reasoning.

Individual differences. We are once again focusing on the differences in the probability values between tasks among participants, but now we are taking into consideration the individual differences. Here we are given the probability values for all tasks in the reduced inference/conditional case, and the probability values for one task in the conditional/biconditional case. Using that information, we aim to predict the endorsements for the other tasks in the conditional/biconditional case. In this case we need 2×3 (3 parameters for 2 tasks, reduced inference/conditional) + 3 (3 parameters for 1 task, conditional/biconditional) → 9 parameters. Eq. 6 shows how the calculations of the probability values for the conditional case are done, and Eq. 7 shows the same for the biconditional case.

$$p_i^c(t2) = p_i^c(t1) - (p_i^r(t1) - p_i^r(t2)) \tag{6}$$

$$p_i^b(t2) = p_i^b(t1) - (p_i^c(t1) - p_i^c(t2)) \tag{7}$$

[5] In the case of prediction we do not take into consideration the R^2 measure, because, as shown in (Alexander, Tropsha, & Winkler, 2015), RMSE provides a better prediction quality measure.

Table 8: Prediction results when using constants to obtain probability values. 'Task' is the task to which we fit the model and use to predict the other three tasks. (D/A - Disablers/Alternatives; F - Few; M - Many; Red. Inf. - Reduced Inference)

	Red. Inf.	Conditional	Biconditional
Task (D/A)	RMSE	RMSE	RMSE
F/F	.245	.125	.231
F/M	.260	.118	.191
M/F	.226	.136	.200
M/M	.210	.099	.181

E.g., if we have fitted all the parameters for the reduced inference, and the 'Girl' task in the conditional case, and we want to predict the probabilities for the 'Coke' task in the conditional case, we would follow Eq. 8.

$$p_i^c(c) = p_i^c(g) - (p_i^r(g) - p_i^r(c)) \tag{8}$$

Table 9: Prediction results when using individual task differences to calculate probability values. 'Conditional' and 'Biconditional' denote predicting for that condition. (D/A - Disablers/Alternatives; F - Few; M - Many)

	Conditional	Biconditional
Fitted task (D/A)	RMSE	RMSE
F/F	.336	.255
F/M	.327	.238
M/F	.222	.336
M/M	.263	.203

The RMSE for this approach can be found in Table 9. This approach makes the assumption that there are similarities in the individual differences of probability values between tasks for all conditions.

Condition probability distribution differences

In this approach we focus on how the probability values change between the reduced inference and the conditional case and between the conditional and biconditional case. We are aiming to predict a task in the conditional/biconditional case, by fitting another task in both, the reduced inference and conditional/conditional and biconditional case, and the to-be predicted task in the reduced inference/conditional case, which totals to 9 parameters (2×3 + 3). We will only take into consideration individual probability differences. Eq. 9 and 10 show how the calculations of the probability values are done.

$$p_i^c(t2) = p_i^r(t2) - (p_i^r(t1) - p_i^c(t1)) \tag{9}$$

$$p_i^b(t2) = p_i^c(t2) - (p_i^c(t1) - p_i^b(t1) \tag{10}$$

E.g., if we have fitted the probability values for the 'Balloon' task in the conditional and biconditional case, and the 'Coke' task in the conditional case, we can calculate the probability values for the 'Coke' task in the biconditional by 11.

$$p_i^b(c) = p_i^c(c) - (p_i^c(b) - p_i^b(b)) \tag{11}$$

Table 10: Prediction results when using individual condition differences to calculate probability values. 'Fitted task' is the task that is fitted in both conditions, whose parameter differences are used to predict other tasks. (D/A - Disablers/Alternatives; F - Few; M - Many; Red. - Reduced Inference; Cond. - Conditional; Bicond. - Biconditional)

	Red. to Cond.	Cond. to Bicond.
Fitted task (D/A)	RMSE	RMSE
F/F	.203	.152
F/M	.283	.261
M/F	.444	.167
M/M	.297	.322

The RMSE values for this approach can be found in Table 10. This approach makes the assumption that the individual differences of probability values when changing the type of rule are similar among different tasks.

Future work

In this paper we presented a combination of the Mental Model Theory and Pearl's ε-semantics. It is able to account for the influence of disablers and alternatives and the type of conditional. Using three parameters per task, we achieved a good fit. It is a starting point that will need more exploration to bring different cognitive computation theories closer together.

Acknowledgements

This work was supported by Heisenberg grants RA1934/3-1, RA1934/4-1, and RA1934/9-1 to MR.

References

Alexander, D., Tropsha, A., & Winkler, D. A. (2015). Beware of R^2: simple, unambiguous assessment of the prediction accuracy of QSAR and QSPR models. *Journal of Chemical Information and Modeling*, *55*(7), 1316–1322.

Cummins, D. D., Lubart, T., Alksnis, O., & Rist, R. (1991). Conditional reasoning and causation. *Memory & Cognition*, *19(3)*, 274–282.

Elqayam, S., & Over, D. E. (2013). New paradigm psychology of reasoning: An introduction to the special issue edited by Elqayam, Bonnefon, and Over. *Thinking & Reasoning*, *19*(3-4), 249–265.

Johnson-Laird, P. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness.* Harvard University Press.

Oaksford, M., Chater, N., & Larkin, J. (2000). Probabilities and polarity biases in conditional inference. *Journal of Experimental Psychology: LMC*, *26(4)*, 883.

Pearl, J. (1991). *Epsilon-semantics* (Tech. Rep.). Computer Science Department, University of California.

Singmann, H., Klauer, K. C., & Beller, S. (2016). Probabilistic conditional reasoning: Disentangling form and content with the dual-source model. *Cognitive Psychology*, *88*, 61-87.

A Process Model of Magnitude Estimation

J. Gregory Trafton (greg.trafton@nrl.navy.mil)
Naval Research Laboratory
Washington, DC 20375 USA

Abstract

We present a cognitively plausible model of non-verbal counting and magnitude estimation. Unlike existing models, the current model does not use a perfect representation of magnitude, time, or memory. Instead, it calculates a magnitude based on an imperfect rate of counting and determines when to stop counting based on an internal timer. Empirical data at both the individual and average level is matched to show a range of performance.

Keywords: magnitude-estimation; nonverbal counting; cognitive modeling

Introduction

Numerosity, magnitude, estimation, and counting are fundamental aspects to human life. Some researchers have suggested that numerosity is one of people's core concepts (Carey, 2009). Other researchers have shown that animals can count, even without an explicit (verbal) counting mechanism (Platt & Johnson, 1971).

In fact, magnitude has been explored intensively using a variety of methods including counting (Whalen, Gallistel, & Gelman, 1999), size (Moyer & Landauer, 1967), math (C. Gallistel & Leon, 1991; C. R. Gallistel & Gelman, 2000), and perception of number (Wynn, 1992). Counting, while being one of the purest measures of magnitude, has probably been studied the least, at least in humans. Additionally, there are relatively few cognitive models of counting, though there are other models of size, perception and math.

Here, we are concerned with *non-verbal counting*, where a person performs an action (e.g., lever-pressing) a specified number of times without explicit enumeration (Whalen et al., 1999). Mathematicians and philosophers have argued that non-verbal counting is the basis of higher-order math (Bell, 1937). Non-verbal counting is also used in many everyday situations, from determining how much trash is in a garage to the number of people in a queue to the number of steps on an escalator.

Previous researchers have suggested that non-verbal counting occurs through an internal, noisy accumulator (Meck & Williams, 1997; Gibbon, Church, & Meck, 1984; Meck & Church, 1983). In these accounts, magnitudes have scalar variability, varying in proportion to the mean of the magnitude (C. R. Gallistel & Gelman, 2000). Because magnitudes have scalar variability, the discriminability of the values obeys Weber's law because the degree of overlap between representations remains constant as the ratio of the means is held constant. Current accounts make these assumptions:

- An accumulator is incremented based on count (Cordes, Gelman, Gallistel, & Whalen, 2001; Whalen et al., 1999) or time (Dormal, Seron, & Pesenti, 2006; Meck & Williams, 1997).
- The accumulator value has a perfect representation, but when checked internally is noisy; the bigger the value of the accumulator, the bigger the noise (C. R. Gallistel & Gelman, 2000; Meck & Church, 1983; Meck, Church, & Gibbon, 1985).

There are several major concerns with these assumptions, however.

Over-reliance on perfect accumulators or perfect memory: First, most accounts that assume that the accumulator is based on an actual count assume that the counter is perfect (Cordes et al., 2001; Whalen et al., 1999), which is cognitively implausible. In these approaches, the counter is represented perfectly, but is retrieved with noise. For accounts that assume that the accumulator is based on an internal timer, the assumption is that the timer is perfect (Meck et al., 1985). We know from many studies of time sense that people do not have perfect representations of time (Zakay & Block, 1997; Matell & Meck, 2000) and that people are able to estimate time more accurately at shorter intervals than longer intervals. At least some of these approaches also assume perfect memory (Gibbon et al., 1984). These assumptions allowed early progress to be made on the initial models and theorizing, which clearly advanced the field. Unfortunately, these assumptions have continued on through many of the current models of counting and may lead to an incorrect understanding of how people perform non-verbal counting.

Sampling problem: If a human counter samples magnitude from a Gaussian distribution and periodically checks that magnitude against a target goal, a trace of the magnitude across a counting scenario will show it to sometimes become negative or go backwards (a standard assumption of most accumulator models and inherent in consecutive random sampling). If a further constraint is added so that the magnitude must be positive and always increase, the magnitude will consistently *under* represent the actual count. This under counting will become greater the bigger the target is because there is more opportunity for skipping a number.

Our goal here is to remove these problems and present a process model of how people perform these implicit counting tasks. We assume that people do not have a perfect sense of memory, time, or magnitude when counting non-verbally. We describe our model in the context of a classic counting experiment by Whalen et al. (1999).

Method (Whalen et al., 1999)

A complete description of the experiment can be found in Whalen et al. (1999).

Participants

Seven volunteers participated in the experiment over 8 1-hour sessions (which included other related tasks as well).

Setup and Procedure

A trial began with a ”Ready?” message in the center of the screen. When the participant pushed a button, the ”Ready?” message was replaced with an odd number from 7 - 25 (inclusive). Participants were instructed to push a key the specified number of times, as fast as they could. Participants completed a trial by pushing a different key. Participants performed 40 trials for each odd number from 7 - 25. No feedback was given regarding their accuracy.

Participants were specifically instructed **not** to verbally count the number of presses made, but to arrive at their target-goal “by feel.”

Measures

The target-goal and the number of actual keypresses was recorded and averaged for each participant. The standard deviation and coefficient of variation was also measured for each participant.

Results and Discussion

Participants were reasonably accurate for most target-goals. The average number of presses increased linearly with the target value. For all participants, the standard deviation of the number of key presses varied in direct proportion to the target magnitude.

The most surprising finding, however, concerned the coefficient of variation (the ratio of standard deviation and mean). Specifically, the coefficient of variation was *constant* across target size. Figure 1 shows the averaged data across the seven participants (digitally extracted from the original article).

Participants were presumably not performing overt or covert verbal counting because the rate that they were able to push the key (∼120ms/item), is much faster than subvocal counting can occur (∼240ms; Klahr, 1973). In fact, when participants were instructed to explicitly subvocalize, their RT was significantly and consistently longer than when they performed the non-verbal counting task. The difference between subvocalizing and non-verbal counting was much bigger when the numbers had more syllables (e.g., ”nine” vs. ”seventeen”).

Architecture and Model Description

ACT-R is a hybrid symbolic/sub-symbolic production-based system (Anderson et al., 2004) ACT-R consists of a number of modules, buffers, and a central pattern matcher. Modules in ACT-R contain a relatively specific cognitive faculty usually associated with a specific region of the brain. For each module, there are one or more buffers that communicate directly

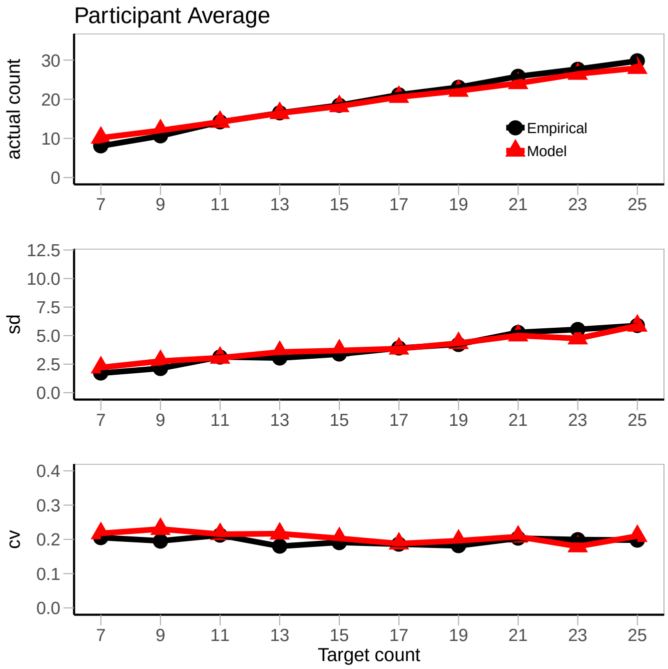


Figure 1: Average performance of the seven individuals in the Whalen et al. (1999) study. The x axis in all three graphs is the target count (the goal that the participants were given). The top panel shows the (remarkably accurate) accuracy on counting. The middle panel shows the increasing standard deviation the higher the target goal becomes. The bottom panel shows the flat coefficient of variation. The darker circles show the data digitally extracted from the original article while the lighter triangles show the model fit.

with that module as an interface to the rest of ACT-R. At any point in time, there may be at most one item in any individual buffer; thus, the module's job is to decide what and when to put a symbolic object into a buffer. The pattern matcher uses the contents of the buffer to match specific productions.

ACT-R uses if-then rules (productions) that will fire when their preconditions are met by matching the contents of the buffers. If there is more than one production that can fire, the one with the highest utility (production strength) will fire. Each production can change either internal state (e.g., buffer contents) or perform an action (e.g., click on a button).

ACT-R interfaces with the outside world through the visual module, the aural module, the motor module, and the vocal module. The architecture supports other faculties through intentional, imaginal, temporal and declarative modules.

Because most researchers believe that numerosity is a core concept (Carey, 2009) and many animals can actually count non-verbally, we have created a new ACT-R module, called the magnitude module.

The Magnitude Module

The magnitude module provides a mechanism for performing non-verbal counting until a specific target-goal is reached.

Instead of relying on a perfect counter or a perfect sense of time, the magnitude module only has imperfect representations of time and counting. Note that the magnitude module is not used for exact, verbal counting, but rather for non-verbal numeric estimation (exact verbal counting can be performed easily by traditional ACT-R).

A key component to non-verbal counting is deciding when to stop. We propose here that the internal temporal module (Taatgen, Van Rijn, & Anderson, 2007) is used. The temporal module tracks time intervals and is quite accurate at short timer scales, becoming progressively less accurate and noisier at longer time scales. The temporal module simply keeps track of how long it has taken since counting began. A rate of counting is calculated based on the (noisy) timer and an updated previous magnitude. Finally, a target amount of time can be determined based on the rate and the target number.

High level description of the magnitude module

There are three components to each model: start, count, finish.

Start The model prepares to begin counting by setting a target-goal (e.g., 17) and preparing to count (e.g., by putting their finger on the counting key). The rate is undefined at this point.

Count The model counts by making a call to the magnitude module for every count it makes. Every count initiates a physical keypress as well. Every count, several quantities are updated.

Rate A current rate is calculated based on the amount of time that has passed since counting began and the successor of the last magnitude.

Magnitude The current magnitude is calculated based on current time and the current rate. Note that because magnitude is based on the model's imperfect sense of time and an imperfect rate, it never has a perfect representation of count. Because the timer is more accurate at short time intervals, it is frequently (but not always) correct at smaller counts. Subitizing is not explicitly modeled and in fact previous researchers have suggested that subitizing is not needed during non-verbal counting (Cordes et al., 2001).

Time-to-stop Time to stop is based on the rate × target-goal. Because people have different levels of accuracy for non-verbal counting, a mean-scalar (*m*) and a standard-deviation-scalar (*sd*) are included in this calculation.

Notice that magnitude ends up having scalar variability. In this account, scalar variability arises because of the imperfect time sense that people have.

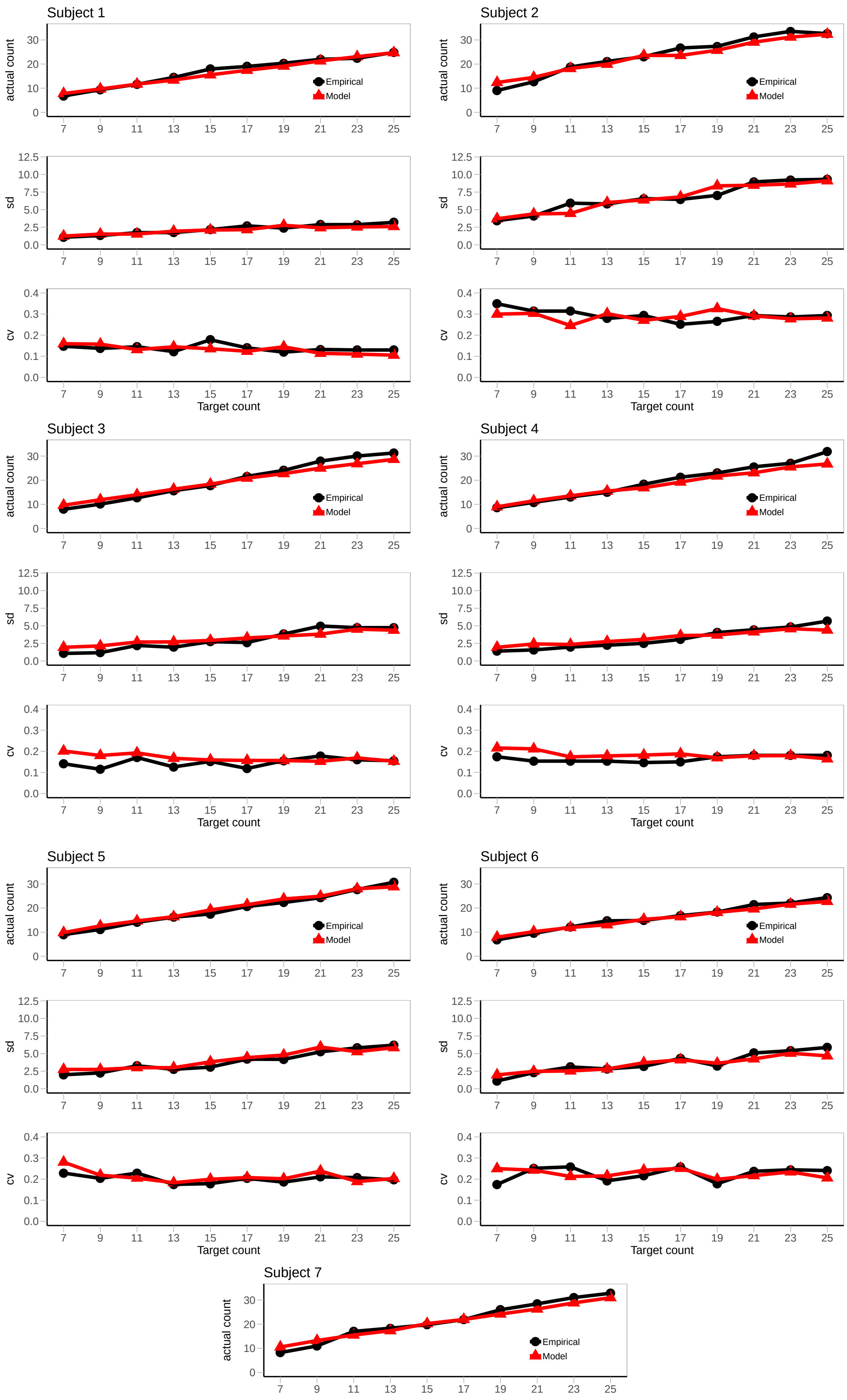
Finish The model finishes counting when the current time is greater than or equal to the computed time-to-stop.

These three components occur in the natural order: Start begins a trial, while Count performs the counting itself, and then Finish ends the trial.

Model Fit

The data was presented in the original Whalen et al. (1999) study as a series of graphs of the seven individuals. A single graph of average performance by participant was not presented (presumably to show that the coefficient of variation was constant across every single participant). The individual data was digitally extracted and averaged into the graph shown in Figure 1. A model was fit to every single participant as well, show in in Figure 2.

Model fits were created by running the model 250 times for both the overall average and each individual. 250 was selected because it provided stability across the entire range of participants and variables. All standard ACT-R parameters were left at their defaults. Two magnitude parameters (*m* and *sd*) were fit for each participant and for the average performance. Both parameters stayed within a narrow range (.1 - .7 for *m* and .3-.5 for *sd*); changes to these parameters only impacted the strength of the individual fit, not the overall pattern.


Subject 1
Subject 2
Subject 3
Subject 4
Subject 5
Subject 6
Subject 7
actual count
sd
cv
Target count
Empirical
Model
0
10
20
30
0.0
2.5
5.0
7.5
10.0
12.5
0.0
0.1
0.2
0.3
0.4
7
9
11
13
15
17
19
21
23
25

Participant	R^2	RMSD
1	.99	.48
2	.99	7.1
3	.99	3.5
4	.99	2.3
5	.99	4.0
6	.99	1.2
7	.99	4.9
All Participants	.99	3.2

Table 1: Model fit table for counting accuracy (top panels).

Participant	R^2	RMSD
1	.79	.35
2	.87	.70
3	.92	.67
4	.92	.63
5	.90	.53
6	.88	.74
7	.86	.54
All Participants	.94	.42

Table 2: Model fit table for counting standard deviation (middle panels).

R^2 and RMSD fit metrics between the empirical and model data were generated for each individual participant and the average of all participants. Table 1 shows the fit metrics for the count data (how accurate the counting was; top panels). Table 2 shows the fit metrics for the standard deviation data (how the standard deviation increased across target count; middle panels). Table 3 shows the fit metrics for the coefficient of variation data (the relatively constant values across target count; bottom panels). For count data and standard deviation, a high R^2 and a low RMSD shows a good fit. For the coefficient of variation fit metrics, R^2 should be close to 0 because it is a constant, while RMSD should be low.

As can be seen in the fit tables and the graphs, the model fits the data quite well on all three primary variables: count, standard deviation, and coefficient of variation.

Participant	R^2	RMSD
1	.02	.02
2	.04	.04
3	.02	.04
4	.04	.03
5	.04	.02
6	.00	.04
7	.09	.02
All Participants	.06	.02

Table 3: Model fit table for counting coefficient of variation (bottom panels). Note that the R^2 should be close to 0.

General Discussion

We described a process model for non-verbal counting. Our model has several advantages over existing models. First, current models typically rely on an internal representation that is perfect – of magnitude, time, or memory. Consistent with most of people's representations, we believe that none of these are represented perfectly.

The current model does not have a perfect model of time. Previous models use 'clock time' to calculate rates and therefore magnitude. However, there is a great deal of evidence that people's sense of time is quite good for short intervals and becomes worse at longer intervals (Matell & Meck, 2000; Taatgen et al., 2007). Thus, this model uses a cognitively plausible measure of time intervals (Taatgen et al., 2007).

The current model does not have a perfect model of magnitude. Magnitude is represented as a scalar value that increases over time and in the non-verbal counting task we have modeled here it is created directly from the rate of counting. The model suggests that magnitude estimation is inherently imperfect because people do not have a perfect representation of time.

The current model does not have a perfect representation of memory, though it inherits that memory imperfection from ACT-R (Altmann & Trafton, 2002). In the current model, memory is not explicitly used, but certainly if the model needed to store, remember, and retrieve a magnitude the machinery exists to do so.

The current model also solves the sampling problem discussed earlier. Because this model determines when to stop based on time, this model never has a negative or backwards-going magnitude. Nor does this model consistently undercount because of a greater chance of skipping numbers.

The current model can presumably explain non-verbal counting in animals as well. Animals seem to represent magnitudes in the same way that people represent non-verbal magnitudes (Church, 1984; Gibbon et al., 1984; Meck & Church, 1983), and this model would capture the same features (e.g., scalar variability) of animal counting that have been described in the literature (Platt & Johnson, 1971).

It is interesting to note that both magnitude and time sense have similar representations: they both have scalar variability, more accurate at smaller numbers and less accurate at bigger numbers. This remarkable similarity suggests that both time and magnitude are intimately connected. In our model, we connect them directly: people's sense of time is critical to how magnitude estimations occur. Without a sense of time (or if time-sense is being used for something else), the model suggests that magnitude estimation is exceedingly difficult – perhaps so difficult that another strategy would need to be used.

ACT-R is well known for modeling average behavior, and equally well known for not being able to model variability very well. A typical model fit, for example, shows empirical means and model means overlapping. However, these models very rarely adequately model the variability inherent in the empirical data. This model, however, models not only the mean data, but also the variability. This emphasis on modeling the full distribution of behavior is a core strength of our approach here.

We should emphasize that the current model is for non-verbal counting only. Other researchers have studied other forms of numerosity – estimating the number of objects on a screen; explicit counting; approximate counting, and others. Exactly how this model will scale to those other tasks is for future work. Certainly a similar model could presumably capture the observed empirical patterns: examining density and then extrapolating based on how long it took to determine density may be a method to estimate the number of objects on a screen.

In summary, the current model emphasizes non-verbal counting using cognitively plausible – and imperfect – core mechanisms. We modeled one of the best known empirical examples of non-verbal counting (Whalen et al., 1999) and it is the only existing model we know of that captures the full range of non-verbal counting through a high-fidelity process model.

Acknowledgments

This work was supported by the Office of Naval Research to JGT. The views and conclusions contained in this document should not be interpreted as necessarily representing the official policies of the U. S. Navy. We thank Sunny Khemlani, Anthony Harrison, Hillary Harner, and Gordon Briggs for their advice and comments on a previous draft.

References

Altmann, E. M., & Trafton, J. G. (2002). An activation-based model of memory for goals. *Cognitive Science*, 39-83.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Bell, T. (1937). Complete independence. *Men of mathematics*.

Carey, S. (2009). *The origin of concepts*. Oxford University Press.

Church, R. M. (1984). Properties of the internal clock a. *Annals of the New York Academy of Sciences*, *423*(1), 566–582.

Cordes, S., Gelman, R., Gallistel, C. R., & Whalen, J. (2001). Variability signatures distinguish verbal from nonverbal counting for both large and small numbers. *Psychonomic bulletin & review*, *8*(4), 698–707.

Dormal, V., Seron, X., & Pesenti, M. (2006). Numerosity-duration interference: A stroop experiment. *Acta psychologica*, *121*(2), 109–124.

Gallistel, C., & Leon, M. (1991). Measuring the subjective magnitude of brain stimulation reward by titration with rate of reward. *Behavioral neuroscience*, *105*(6), 913.

Gallistel, C. R., & Gelman, R. (2000). Non-verbal numerical cognition: From reals to integers. *Trends in cognitive sciences*, *4*(2), 59–65.

Gibbon, J., Church, R. M., & Meck, W. H. (1984). Scalar timing in memory. *Annals of the New York Academy of sciences*, *423*(1), 52–77.

Klahr, D. (1973). A production system for counting, subitizing and adding. In *Visual information processing* (pp. 527–546). Elsevier.

Matell, M. S., & Meck, W. H. (2000). Neuropsychological mechanisms of interval timing behavior. *Bioessays*, *22*(1), 94–103.

Meck, W. H., & Church, R. M. (1983). A mode control model of counting and timing processes. *Journal of Experimental Psychology: Animal Behavior Processes*, *9*(3), 320.

Meck, W. H., Church, R. M., & Gibbon, J. (1985). Temporal integration in duration and number discrimination. *Journal of Experimental psychology: animal behavior processes*, *11*(4), 591.

Meck, W. H., & Williams, C. L. (1997). Characterization of the facilitative effects of perinatal choline supplementation on timing and temporal memory. *Neuroreport*, *8*(13), 2831–2835.

Moyer, R. S., & Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature*, *215*(5109), 1519.

Platt, J. R., & Johnson, D. M. (1971). Localization of position within a homogeneous behavior chain: Effects of error contingencies. *Learning and Motivation*, *2*(4), 386–414.

Taatgen, N. A., Van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review*, *114*(3), 577.

Whalen, J., Gallistel, C. R., & Gelman, R. (1999). Nonverbal counting in humans: The psychophysics of number representation. *Psychological Science*, *10*(2), 130–137.

Wynn, K. (1992). Addition and subtraction by human infants. *Nature*, *358*(6389), 749.

Zakay, D., & Block, R. A. (1997). Temporal cognition. *Current directions in psychological science*, *6*(1), 12–16.

Cognitive Modeling with Symbolic Deep Learning

Vladislav D. Veksler (vdv718@gmail.com)
DCS Corp, U.S. Army Research Laboratory

Norbou Buchler
U.S. Army Research Laboratory

Keywords: symbolic modeling, deep learning, machine learning, cognitive architectures

Automatic generation of user-models based on user-task interactions is the holly grail of Cognitive Modeling and Human-Computer Interaction fields. Such automatic model generation would be of great use for behavioral predictions, better understanding of cognition, and better understanding of the task-environment. Mapping which environment features cause which actions seems like a classification problem, perfectly suited for machine-learning techniques like Deep Learning (DL). There are, however, some drawbacks in current-form Deep Learning approaches that make it less than ideal for automatic model generations based on limited user-task interactions. In this paper we bring examples of DL-like symbolic cognitive framework approaches that have the potential to overcome such drawbacks.

Deep Learning

Deep Learning is a multi-layer neural network approach that has received much recent adoration for unprecedented success in input and situation recognition and classification (Rusk, 2015). Unfortunately, DL suffers from a few drawbacks that limit its applicability across domains.

First, DL does not create an observable model. That is, what deep networks learn cannot be investigated beyond a general input-output mapping. DL could still be useful in the domain of user-modeling for predicting user actions, but not for understanding the cognitive state responsible for that state-action mapping. This problem falls under the domain of explainable AI (XAI), and bares additional significance for accepting/trusting any recommendations derived via DL methods.

Second, DL is susceptible to catastrophic interference – where new training examples can break a previously stable classifier. This issue arises specifically in dynamic domains, where there is no immutable training set, and the classifier needs to be constantly updated.

Finally, DL is more suitable to making predictions from billions of examples than from a few dozens or even hundreds of observations. This is the greatest limitation of the deep learning approach, making it unsuitable for small-data domains. This makes DL especially difficult to apply for generating predictions from experts in narrow domains, where little data can be obtained from subject-matter experts (e.g. cybersecurity). Additionally, this makes it difficult to employ DL for learning from individual users, since single-user behavior usually would not generate enough data for DL classifiers.

However, the multi-layer hierarchical approach to classification is not exclusive to the big-data AI domains. Many symbolic cognitive frameworks are based on hierarchical memory that is very similar to subsymbolic deep neural network approaches, without the aforementioned limitations.

Symbolic Deep Learning

Symbolic Deep Learning (SDL) is promising in that this method is capable of building classifiers from a small number of examples, rather than the millions required for more traditional ML/DL methods (d'Avila Garcez, Dutra, & Alonso, 2018; Dutra, Garcez, & D'Avila Garcez, 2017; Zhang & Sornette, 2017). In this way, SDL learning efficiency is much closer to that of humans than that of DL. Moreover, SDL memory is incremental (i.e., does not require a pre-specified size of the network), and is thus robust against catastrophic interference. Finally, symbolic memory lends itself to human interpretation, thus addressing the issues relating to XAI. Essentially, SDL addresses all of the traditional DL limitations, and is a promising avenue for automatic model generation.

Symbolic hierarchical representations have a long history in Psychological literature. Some of these were integrated as models of memory without action-selection (e.g. Feigenbaum & Simon, 1984; Gobet & Lane, 2005). Such purely declarative models are more useful for predicting recognition than state-action mapping.

Integrated cognitive architectures that include both state recognition and action selection often include hierarchical memory systems, as well. For example, declarative memory chunks in ACT-R are symbolic memory elements that are, in fact, sets of links to lower-level chunks (Anderson, 1993; Anderson & Lebiere, 1998). The ACT-R theory is incomplete in its description of how chunks are created (beyond those created upon goal-completion). An integration of cognitive architectures like ACT-R with learning/memory model like EPAM/CHREST may ultimately be the solution to automatic model generation.

The most promising models of hierarchical learning/memory systems for the purposes of SDL system development and automatic model generation may be found in categorization research domain. Models in the categorization literature were specifically developed with the purposes of mapping multi-feature inputs onto participant decisions (e.g. Gluck & Bower, 1988; Nosofsky, Gluck, Palmeri, McKinley, & Glauthier, 1994).

The greatest problem facing such hierarchical symbolic memory systems seem to be those of computational limitations. For example, the configural-cue model of memory (Gluck & Bower, 1988) creates a configural node (i.e. chunk) for every unique set of potential inputs, thus creating a maximum of $(k + 1)^n - 1$ memory chunks, where n is the number of input dimensions and k is the number of possible input values along each input dimension[1]. Although this exponential memory growth is concerning for large-input domains (e.g. image recognition), it should not cause much issue in the domain of automatic model generation for most non-graphical tasks.

For example, let us assume a specific user interface such as an Intrusion-Detection System (IDS). When cyber-security professionals employ such a system, each of their observations constitutes a network alert record, and each observation is followed by a decision whether to elevate the alert, or not to (this task-environment is fairly representative of much non-graphical software UI across domains). Such a record will comprise 5-10 fields, consisting of a time-stamp and a few other mostly nominal values such as a port-number, operating system, alert-type, etc. There are only a few port-ranges that are ever observed, only a few types of alerts, etc. Assuming five input fields with $\approx$10 potential values in each field, the configural-cue memory system would grow to $\approx$160 thousand nodes. Of course there will less than 10 potential values for some fields and more than 10 potential values for others, but it is reasonable to presume that even a low-end PC can handle this load, much less a modern server using GPU acceleration. Even with ten input fields (a maximum of $\approx$26 billion nodes) we can expect computational power to no-longer be the limitation that it was decades ago when this model was first proposed.

Perhaps more important than the raw computational power available today, there is efficiency to be gained in SDL by creating memory chunks only when they prove necessary. For example, Veksler, Gluck, Myers, Harris, and Mielke (2014) propose to a conservative-rational incremental memory system that reduces memory size, especially in noisy environments. Such memory reduction is exponential, improving efficiency by several factors of magnitude, and greatly reducing the concern over computational limitations for SDL.

Summary

Both, symbolic and subsymbolic deep learning methods date back a half century, and both were shelved for decades due to a lack of computational resources needed to run these algorithms. The modern era of parallel processing and GPU computing, along with some algorithmic efficiency has revived Deep Learning as a field. The same technological advances, including SDL-specific algorithmic efficiency improvements are ripe to revive the SDL field, as well.

SDL promises to overcome many of the limitations of subsymbolic DL, enabling applicability in small-data domains, incremental memory processes that are robust to catastrophic interference, and observability and explainability of the learned state-action mapping (XAI). Given this potential, SDL seems like the right technique for automatically generating models of user behavior, especially for modeling expert or individual behavior.

References

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

d'Avila Garcez, A., Dutra, A. R. R., & Alonso, E. (2018, apr). Towards Symbolic Reinforcement Learning with Common Sense. *CoRR*, *abs/1804.0*. Retrieved from `http://arxiv.org/abs/1804.08597`

Dutra, A. R. A., Garcez, A., & D'Avila Garcez, A. S. (2017). A Comparison between deep Q-networks and deep symbolic reinforcement learning. In *Ceur workshop proceedings.* Retrieved from `http://ceur-ws.org/Vol-2003/NeSy17_paper6.pdf`

Feigenbaum, E., & Simon, H. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, *8*(4), 305–336.

Gluck, M. A., & Bower, G. H. (1988). From Conditioning to Category Learning - an Adaptive Network Model. *Journal of Experimental Psychology-General*, *117*(3), 227–247.

Gobet, F., & Lane, P. C. R. (2005). The CHREST architecture of cognition: Listening to empirical data. *Visions of mind: Architectures for cognition and affect*, 204–224.

Nosofsky, R. M., Gluck, M. A., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: a replication and extension of Shepard, Hovland, and Jenkins (1961). *Mem Cognit*, *22*(3), 352–369.

Rusk, N. (2015, dec). *Deep learning* (Vol. 13) (No. 1). Nature Publishing Group. doi: 10.1038/nmeth.3707

Veksler, V. D., Gluck, K. A., Myers, C. W., Harris, J., & Mielke, T. (2014, oct). Alleviating the curse of dimensionality âĂŞ A psychologically-inspired approach. *Biologically Inspired Cognitive Architectures*, *10*, 51–60. doi: 10.1016/j.bica.2014.11.007

Zhang, Q., & Sornette, D. (2017, jul). Learning like humans with Deep Symbolic Networks. *arxiv.org*. Retrieved from `http://arxiv.org/abs/1707.03377`

[1] Given n features (e.g. *large*, *square*, *white*), we can create a chunk for every combination of feature presence and absence ({*large*}, {*square*}, {*white*}, {*large*, *square*}, {*large*, *white*}, {*square*, *white*}, and {*large*, *square*, *white*}). If we represent feature presence as a 1 and feature absence as a 0, we can represent each chunk as a binary number, and the total number of possible chunks is the total number of possible binary numbers, minus the blank chunk, which is $2^n - 1$. When each feature dimension can have two potential values, the total number of possible chunks is $3^n - 1$. With k possible values on n feature dimensions, we can have at most $(k + 1)^n - 1$ possible chunks to represent all potential feature combinations.

Kickstarting Adaptive Fact Learning Using Hierarchical Bayesian Modelling

Maarten van der Velde[1] **(m.a.van.der.velde@rug.nl)**
Florian Sense[1] **(f.sense@rug.nl)**
Jelmer Borst[2] **(j.p.borst@rug.nl)**
Hedderik van Rijn[1] **(d.h.van.rijn@rug.nl)**
[1]Dept. of Experimental Psychology & Behavioural and Cognitive Neuroscience, University of Groningen, The Netherlands
[2]Bernoulli Institute, dept. of Artificial Intelligence, University of Groningen, The Netherlands

Learning facts is an inescapable part of education, whether it be memorising French words or studying US topography. Our lab has developed a digital learning environment that uses a cognitive model of human memory to determine when and how often a fact should be rehearsed during a learning session. The system tracks how difficult each fact is for a given student, continually refines this assessment on the basis of the student's responses, and adjusts the scheduling of items so that difficult facts are repeated sooner and more frequently than easy facts. This adaptive fact learning system has been successfully applied in various contexts (van Rijn, van Maanen, & van Woudenberg, 2009; Sense, van der Velde, & van Rijn, 2018).

Currently, all facts are initially assumed to be equally difficult for all learners. As observations are made, the difficulty estimate is tuned to the right level for each learner-fact pair. This means that knowledge about a learner's general ability or about a fact's typical difficulty is not used at all. In this study, we propose several methods for using data from prior learning sessions to inform the initial difficulty estimates of the model. Using such learning history is expected to make the learning process more efficient, as the model would be better able to quickly hone in on the appropriate difficulty estimate for each fact. We use hierarchical Bayesian modelling to make individualised predictions on the basis of previous learning sessions and test these predictions in a new session.

Adaptive fact learning model

The scheduling of items within a learning session is determined by an adaptive model that builds on earlier work by Pavlik and Anderson (2005). This model is described in more detail in Sense, Behrens, Meijer, and van Rijn (2016).

The model represents each fact by its own memory chunk, with an activation (a measure of the strength of the memory trace) that is boosted by each repetition and decays over time. At time t, and given n previous repetitions at $t_1,...,t_n$ seconds ago, the activation A of chunk i is expressed by Equation 1. The d parameter in this equation controls how quickly a fact's activation decays after a repetition, and therefore how frequently the fact is repeated. Differences in difficulty between facts are captured in the *rate of forgetting* parameter α, a component of d, which is estimated separately for each learner-fact pair. The more difficult a fact is, the higher its rate of forgetting will be, and the faster its activation will decay.

$$A_i(t) = ln\left(\sum_{j=1}^{n} t_j^{-d_i(t)}\right) \quad \text{with} \quad d_i(t) = 0.25 * e^{A_i(t_{n-1})} + \alpha_i \qquad (1)$$

At any given time, the system selects whichever fact has the lowest estimated activation to be rehearsed, thereby maximising the spacing between repetitions while also aiming to repeat each fact before it is forgotten. A new fact is introduced only when all activation values are above a threshold of -0.8.

The system currently starts out with the assumption that all facts have a rate of forgetting of 0.3, and it refines this estimate over the course of the learning session. It uses the difference between expected response times (based on the fact's activation at the time of presentation; the higher the activation, the faster the expected response) and observed response times, as well as response accuracy, to make step-wise adjustments to the estimate that best reflect the observed behaviour.

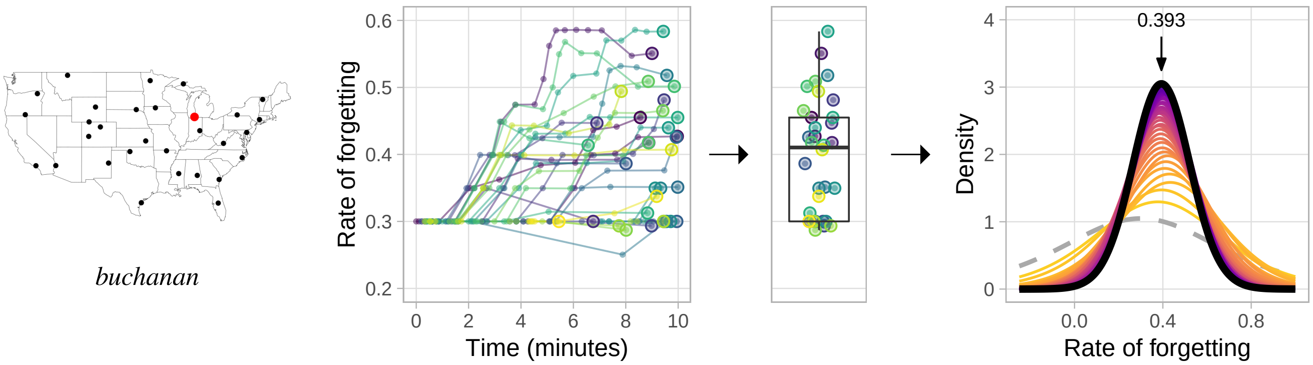
Predicting rate of forgetting

In this study, we use previous learning history to predict what the rate of forgetting of a particular fact will be for a given learner. We then take this prediction as the initial rate of forgetting estimate, rather than the default value.

We test four prediction methods and compare them to the default prediction of 0.3. Fact-level difficulty estimates for a set of topography facts (names of relatively unknown US cities; see Figure 1a for an example) were obtained from an initial experiment in which participants completed a learning session with the default system. In a follow-up experiment, learner-level estimates were derived for different participants who studied a comparable set of facts with the default system. These participants then completed another learning session in which they studied the facts from the first experiment with a system that, depending on the condition to which they were assigned, initialised new facts with a rate of forgetting that was based on one of the four prediction methods or on the default value[1].

Fact-level prediction As multiple learners study the same fact, we form an increasingly detailed picture of its difficulty through the rates of forgetting observed in all these learners. It is to be expected that a new learner studying this fact will

[1] A preregistration with a more detailed description of the protocol and the analysis plan is available at https://osf.io/vwg6u/.



(a) (b) (c) (d)

Figure 1: The process by which a fact-level prediction of the *rate of forgetting* of the fact shown in **(a)** is made. **(b)** In previous learning sessions, the rate of forgetting of the fact is estimated separately for each learner and refined over the course of the session. **(c)** The final estimates (one per learner) are used to train the Bayesian model. **(d)** The posterior predictive distribution of the Bayesian model is updated as observations are added. The prior predictive distribution is shown as a dashed grey line, the final posterior predictive distribution as a black line, with intermediate predictive distributions shown in increasingly dark colours. The model prediction, indicated by the arrow, is the mode of this final distribution.

find it similarly difficult. For this reason, we use the rates of forgetting measured in other learners to make a fact-level prediction of the rate of forgetting that can be used as an initial estimate when the fact is encountered by a new learner. Figure 1 shows how such a prediction is made. Predicted rates of forgetting come from a hierarchical Bayesian model which models the rate of forgetting using a Normal-Gamma distribution with a weakly informative prior centered on 0.3: $NG(\mu = 0.3, \kappa = 3, a = 1, b = 0.2)$.

Learner-level prediction Through the same process as in the fact-level prediction, but instead using the rates of forgetting of all facts that a given learner has encountered in the past, we can predict a learner's rate of forgetting. This value is then used as the initial estimate for all facts that the learner encounters.

Fact- and learner-level prediction We also test a method in which a distinct prediction is made for each learner-fact pair. Two posterior predictive distributions—one for the fact-level prediction and another for the learner-level prediction—are combined using logarithmic opinion pooling (Genest, Weerahandi, & Zidek, 1984) with equal weights. The mode of this combined distribution becomes the predicted rate of forgetting.

Domain-level prediction The domain-level prediction, reflecting the general difficulty of the material in a domain among a certain population, is the mean of all fact-level predictions for the set of facts. This value is used as the initial rate of forgetting in all learner-fact pairs, resulting in a domain-specific alternative to the fixed default prediction of 0.3.

Results & Discussion

Data have been collected from 159 participants for the second experiment, which tests the predictions made by the Bayesian model, while a replication in an online sample is still ongoing. Preliminary results suggest that using learning history to predict rates of forgetting does affect learning performance, as participants are more accurate while studying if the system uses one of the prediction methods (a Bayesian ANOVA shows strong evidence for an effect of condition on accuracy: $BF_{10} = 15.7$), potentially with beneficial effects on motivation. However, this does not appear to translate to higher performance on a delayed recall test (a Bayesian ANOVA shows strong evidence against an effect of condition on test score: $BF_{01} = 19.9$). We will conduct further analyses to address the other questions set out in the preregistration[1], as well as any exploratory questions that arise from this rich data set.

References

Genest, C., Weerahandi, S., & Zidek, J. V. (1984). Aggregating opinions through logarithmic pooling. *Theory and Decision*, *17*(1), 61-70.

Pavlik, P. I., & Anderson, J. R. (2005). Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect. *Cognitive Science*, *29*(4), 559-586.

Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials. *Topics in Cognitive Science*, *8*(1), 305-321.

Sense, F., van der Velde, M., & van Rijn, H. (2018). Deploying a Model-based Adaptive Fact-Learning System in University Courses. In *Proceedings of the 16th International Conference on Cognitive Modeling* (p. 136-137).

van Rijn, H., van Maanen, L., & van Woudenberg, M. (2009). Passing the test: Improving learning gains by balancing spacing and testing effects. In *Proceedings of the 9th International Conference on Cognitive Modeling* (p. 110-115).

The Role of Discourse in Italian Pronoun Interpretation: Investigating Variations in Experimental Results with Cognitive Modeling

Margreet Vogelzang (margreet.vogelzang@uni-oldenburg.de)
Institute of Dutch Studies and Cluster of Excellence "Hearing4all"
University of Oldenburg, Germany

Abstract

This paper investigates to what extent the variations in experimental results on the interpretation of Italian subject pronouns can be explained by the different discourses used in the experimental studies. A cognitive model implemented in ACT-R was used to simulate pronoun processing and interpretation in discourse, which is influenced by the various contexts used in empirical experiments. Our simulations show that the discourse contexts used in the experiments strongly influence the interpretation of Italian subject pronouns, but not to the extent that all data in different experiments can be explained by it. We therefore conclude with suggestions for further research both on the influence of discourse context and the influence of task on the interpretation of Italian pronouns and (linguistic) experiments in general.

Keywords: pronoun interpretation; Italian; cognitive modeling; null subjects; discourse context

Introduction

Referring expressions such as pronouns (*he*, *she*) occur frequently in daily life. It is therefore essential for successful communication that such expressions are understood correctly. Nevertheless, experimental studies on subject pronoun interpretation in Italian show that healthy native adults sometimes have interpretation preferences of around 50% (Tsimpli, Sorace, Heycock, & Filiaci, 2004), suggesting that they are not sure what the meaning is supposed to be. Notably, experimental findings on Italian subject pronoun interpretation show a substantial amount of variation (compare, e.g., Carminati, 2002; Tsimpli et al., 2004; Vogelzang, Foppolo, Guasti, Van Rijn, & Hendriks, 2019). For example, Carminati (2002) and Vogelzang et al. (2019) found a strong preference for null pronouns to refer back to the subject antecedent, whereas in the experiment of Tsimpli et al. (2004) participants only selected the subject antecedent for a null pronoun around half of the time.

One explanation for the varying results could be that these experiments have used different experimental stimuli and different tasks and methodologies. It is generally known that the interpretation of referring expressions can be influenced by the surrounding discourse context. Previous studies have identified several discourse factors that can influence the interpretation of referring expressions. For example, the prominence of discourse referents determines whether they are likely antecedents of a referring expression. In general, less informative referring expressions such as pronouns refer to entities that are highly prominent in the discourse, and more informative referring expressions such as full noun phrases (NPs) refer to entities that are less prominent in the discourse (cf. a.o. Ariel, 1990; Givón, 1983; Gundel, Hedberg, & Zacharski, 1993; for an overview see Arnold, 1998). Additional attention has been given in the literature to coherence relation (Kehler, Kertz, Rohde, & Elman, 2008) and pragmatic plausibility (Carminati, 2002) as influencing the interpretation of referring expressions. In the case of object pronoun interpretation in Dutch, it has been shown that a change in discourse, making the most prominent referent (i.e. the discourse topic) more clear, can eliminate interpretation difficulties that children typically show in other studies (Spenader, Smits, & Hendriks, 2009).

Thus, it is known that discourse context influences the processing and interpretation of referring expressions. It is also known that experimental findings on the interpretation of Italian subject pronouns have not been consistent. It is therefore a logical next step to investigate to what extent differences in discourse can explain variations in experimental results on Italian pronoun interpretation. In this paper, we will focus solely on the influence of the specific experimental stimuli used, putting aside any differences in task, participant sample, and so on. To this end, we will more extensively examine the discourse contexts and experimental results of Tsimpli et al. (2004) and Vogelzang et al. (2019).

The influence of these experimental discourses will be investigated using a cognitive model developed within the cognitive architecture ACT-R. The model will be used to simulate existing empirical data, which will be discussed in the next section. This way, our model simulations will examine to what extent the observed variations in experimental findings can be explained merely by the discourses used.

Experimental Findings on Italian Pronouns

Italian, like Spanish, Catalan, Romanian, Arabic and many other languages, is a language that knows null pronouns. This means that in many cases, a grammatical subject does not have to be realized but can be omitted, creating a null pronoun or null subject (e.g., *corre* 'he/she/it runs'). In addition, Italian has overt pronouns such as *lui* 'he', through which a grammatical subject is explicitly realized. Null pronouns are generally used to refer to the most prominent referent in a discourse, whereas overt pronouns are generally used to refer to a different referent (Carminati, 2002).

Importantly, the influence of discourse prominence on the processing and interpretation of null and overt pronouns in Italian indicates that differences between the findings of different experimental studies (Tsimpli et al., 2004; Vogelzang et al., 2019) may be the result of the different contexts being used. The results of Tsimpli et al. (Figure 1), obtained using a picture selection task, show a much weaker preference for subject antecedents for null pronouns than would be expected based on Carminati's (2002) classical findings.

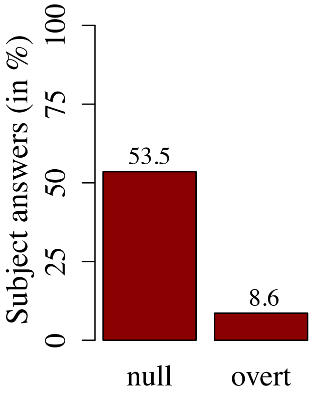


Figure 1. Experimental data from Tsimpli et al. (2004) on the interpretation of null and overt pronouns in Italian.[1]

As stated, this may in part be due to the specific discourses used in the experiment. More specifically, Tsimpli et al. (2004) used short discourse contexts with two clauses such as in (1):

(1) La mamma dà un bacio alla figlia,
mentre lei/Ø si mette il cappotto.

"*The mother kisses the daughter,*
while she/Ø puts on her coat."

In the example context in (1), the prominence of the two referents in terms of frequency is the same, although their grammatical roles differ. The grammatical subject of the first clause is only mentioned once and therefore it is conceivable that this character has not been clearly established as the most prominent referent, which is generally also the discourse topic.

Vogelzang et al. (2019), in contrast, found, using a referent selection task, a strong subject preference for null pronouns (Figure 2).

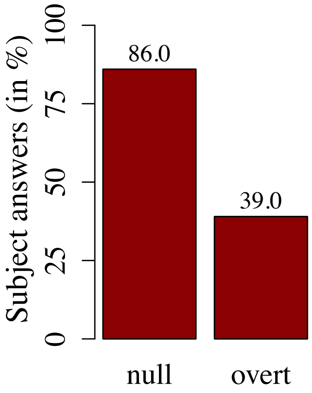


Figure 2. Experimental data from Vogelzang et al. (2019) on the interpretation of null and overt pronouns in Italian.[2]

The longer discourse contexts with three clauses used by Vogelzang and colleagues as in (2) make one referent, which occurs as the grammatical subject twice, much more prominent than the other referent:

(2) Il riccio compra della moquette per il soggiorno.
Ieri il riccio ha raccontato al topo una storia,
mentre lui/Ø si annoiava davanti alla tv.

"*The hedgehog is buying some carpet for the living room. Yesterday the hedgehog has told the mouse a story, while he/Ø was bored in front of the TV.*"

Thus, discourse prominence in terms of recency, frequency, and grammatical role differs between these two experiments, as do the interpretations of participants. In the following sections, we will examine the processing of these discourses and of the pronouns within these discourses more closely with a computational cognitive model.

[1] The experiments of Tsimpli et al. (2004) were mainly focused on language attrition, but this paper will only look at their data on monolingual adults. Additionally, Tsimpli et al. (2004) looked at both forward and backward anaphora, but only backward anaphora will be taken into account in this analysis. Finally, the picture selection task of Tsimpli et al. (2004), in which pictures with two characters and the mentioned action were shown, contained a third answer option, namely an 'other' character, not mentioned in the discourse. To allow for a better comparison between the two studies, this option was not taken into account in the current description of the data, and the percentages of answers were adjusted accordingly.

[2] Vogelzang et al. (2019) included a third subject condition in their experiment, namely a full noun phrase. To allow for a better comparison between the two studies, this option was not taken into account in the current description of the data. In the referent selection task that Vogelzang et al. (2019) used, pictures of the two mentioned characters were shown without any action.

Cognitive Model

The cognitive model we will use to examine the influence of different discourses on Italian pronoun processing is implemented in the cognitive architecture ACT-R (Anderson, 2007; Anderson et al., 2004), which pre-specifies constraints on human cognition and processing. The current model[3] builds on the pronoun interpretation models of Hendriks et al. (2007), Van Rij et al. (2010) and Vogelzang (2017), the latter of which has previously been used to fit the data of Vogelzang et al. (2019) on Italian pronoun interpretation. The most relevant and important aspects of the model will be discussed here; for a more elaborate discussion of the mechanisms used in the model we refer you to Van Rij et al. (2010) and Vogelzang (2017).

The model uses a constraint-based bi-directional approach to pronoun processing, in which a listener reasons both from his/her own perspective and from the perspective of the speaker (cf. Blutner, 2000; Hendriks & Spenader, 2006; previously implemented in a computational cognitive model in Hendriks et al., 2007; Van Rij et al., 2010). The idea that the interpretation of ambiguous referring expressions requires listeners to reason about alternative forms from the perspective of the speaker was originally proposed by Hendriks and Spenader (2006) for Dutch. They formulated their perspective-taking account within the constraint-based linguistic framework Optimality Theory (OT; Prince & Smolensky, 2004), in which hierarchically ranked constraints are used to determine the optimal meaning for an input form (in interpretation) or the optimal output form for a meaning (in production).

In order to expand this approach to Italian, the model discussed here incorporates a constraint regulating the interpretation of null pronouns, stating that null pronouns refer to the discourse topic (similarly to overt pronouns in non-null subject languages, cf. Beaver, 2004; Grosz, Joshi, & Weinstein, 1995; Hendriks, Englert, Wubs, & Hoeks, 2008; Van Rij et al., 2013). As a consequence, a listener would reason that if a speaker would have wanted to refer to the discourse topic, they would have used a null pronoun. So, if the speaker used an overt pronoun instead, they likely wanted to refer to something other than the discourse topic. The other constraints incorporated in the model are based on referential economy (Burzio, 1998), and reflect the idea that speakers prefer to be as efficient as possible and therefore prefer shorter linguistic expressions such as null pronouns over longer linguistic expressions such as overt pronouns. In interpretation, these constraints referring to referential economy will not be relevant. However, because the model additionally reasons about alternative forms from the perspective of the speaker, they will be used to reason about which form a speaker would have most likely used for reference to the topic (null pronoun) or reference to a non-topical referent (overt pronoun).

In addition to constraints, the discourse also affects pronoun processing. More specifically, the model uses a discourse processing component based on the model of Van Rij et al. (2013), in which the prominence of a referent in discourse is determined by the standard ACT-R mechanisms of activation (which is based on its frequency and recency in the discourse) as well as an additional 'boost' that represents additional activation for referents associated with the grammatical subject (set to 1.0). This will most likely make the previous grammatical subject the referent with the highest activation, which can thus be considered the discourse topic. In Van Rij et al. (2013), this grammatical subject boost is argued to represent working memory (WM) capacity, as differences in this activation boost to associated information can account for individual differences in WM capacity (Daily, Lovett, & Reder, 2001).

Results

In this section, different simulations will be described for the processing and interpretation of Italian subject pronouns in the studies of Vogelzang et al. (2019) and Tsimpli et al. (2004). Importantly, every round of simulations uses the same model to simulate the data from both studies, only varying the input (the discourses) presented to the model. Every simulation will differ slightly due to pre-defined mechanisms of the cognitive architecture, such as varying latencies when retrieving information from memory.

In line with the original experiments, the model was run on 32 discourses (items) for 40 simulations (participants) to simulate the experiment of Vogelzang et al. (2019). The model was run on 10 discourses (items) for 20 simulations (participants) to simulate the experiment of Tsimpli et al. (2004). Half of the discourses contained a null pronoun and half contained an overt pronoun.

Simulation 1

For the first round of simulations, the activation boost given to grammatical subjects, representing WM capacity, and the number of practice items presented to the model were kept the same as in Vogelzang (2017); the activation boost was set to 1.0 and the number of practice items to 2000. The results of the simulation for the discourses of Vogelzang et al. (2019) and Tsimpli et al. (2004) are presented in Figures 3 and 4, respectively.

As can be seen in Figure 3, the model data shows very similar interpretational preferences to the experiment data of Vogelzang et al. (2019). The model shows different interpretations, however, compared to the experiment data of Tsimpli et al. (2004), although the tendency of null pronouns referring to the subject and overt pronouns referring to the non-subject is present in both the model data and the experiment data; this can be seen in Figure 4. Specifically, the model shows a higher percentage of subject interpretations for both null pronouns and overt pronouns than the participants in the experiment.

[3] full model code is available at https://sites.google.com/view/margreetvogelzang/experiment-files

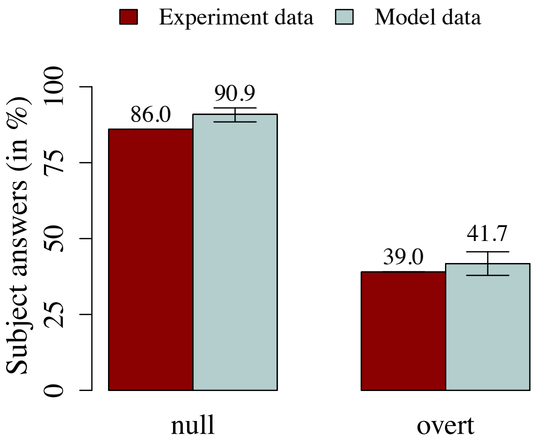


Figure 3. Experimental data from Vogelzang et al. (2019) and model output on Italian pronoun interpretation.

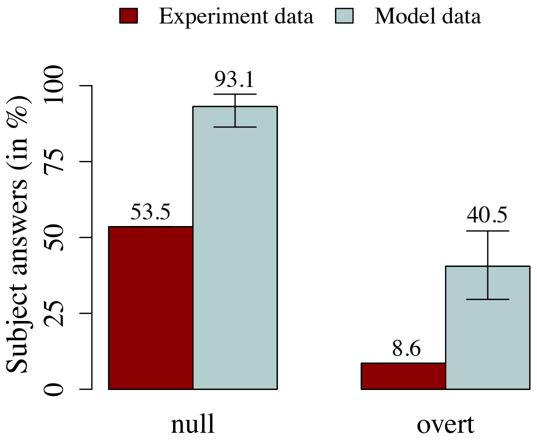


Figure 4. Experimental data from Tsimpli et al. (2004) and model output on Italian pronoun interpretation.

One possible explanation for this could be that the activation boost that is associated with the grammatical subject is constant, i.e. is equally strong no matter how many times a referent is mentioned in the grammatical subject position. Since making the topic of a discourse more clear aids pronoun interpretation (Spenader et al., 2009), however, it is more likely that this activation boost is gradually increasing with each mention rather than a consistently large boost. This possibility will be explored in the next simulation.

Simulation 2

When examining the discourses used in the experiments (see (1) and (2)) more closely, it can be seen that the same character is mentioned in the grammatical subject position once in the discourses used by Tsimpli et al. (2004) compared to twice in the discourses used by Vogelzang et al. (2019). We will now assume, following findings of Spenader et al. (2009), that the discourse topic becomes more clear the more consistent a discourse is, so the more often a certain referent occurs in the grammatical subject position. Figure 5 shows a proposed stepwise activation boost according to the following function:

$$boost = n^2/10$$

in which n is the number of consecutive occurrences of a referent in the grammatical subject position within a discourse.

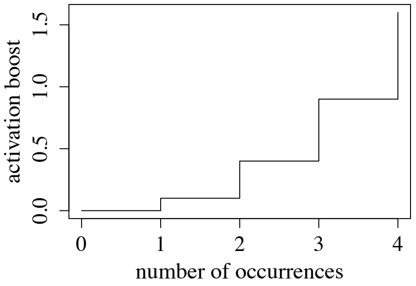


Figure 5. Effect of the number of occurrences of a referent in the grammatical subject position on the activation boost.

Using this function to calculate the boost in activation given to grammatical subjects, new simulations were run for the experiments. The results show that the model can still account for the interpretational preferences of participants in the experiment of Vogelzang et al. (2019) with longer discourses (Figure 6) and that the same model can now also account for the interpretation of null pronouns found in the experiment of Tsimpli et al. (2004) (Figure 7). Notably, the predictions of the model differ considerably based on the discourse used. However, although the interpretational preference of overt pronouns referring to the non-subject can be seen in both the model data and the experiment data, the actual interpretation of overt pronouns in the data of Tsimpli and colleagues is not reproduced by the model. Possible explanations for this are discussed in the next section.

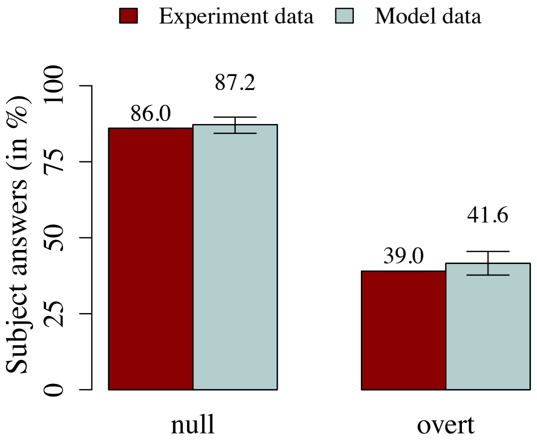


Figure 6. Experimental data from Vogelzang et al. (2019) and model output on Italian pronoun interpretation.

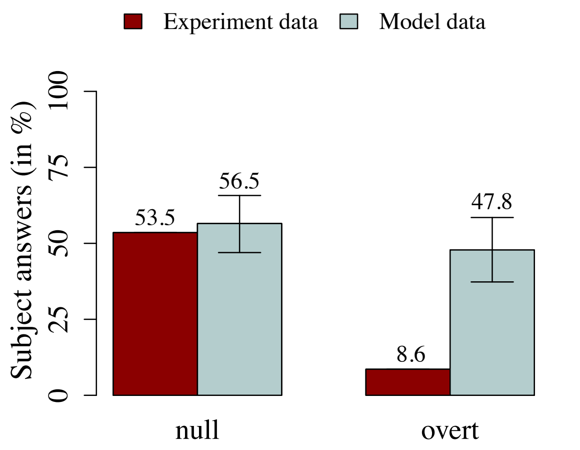


Figure 7. Experimental data from Tsimpli et al. (2004) and model output on Italian pronoun interpretation.

Discussion

In this paper, we investigated with the help of a cognitive model to what extent differences in discourse context can explain variations in experimental results on Italian pronoun interpretation. We examined the discourses used in experiments by Tsimpli et al. (2004) and Vogelzang et al. (2019), which differ in the number of clauses used and the number of times that the referents are mentioned. We simulated pronoun processing in these discourses using an ACT-R model which built on previous cognitive models of pronoun processing (Hendriks et al., 2007; Van Rij et al., 2010; Vogelzang, 2017). The model used an activation boost to keep referents associated with the grammatical subject of the previous sentences active in memory as the discourse topic.

The results from the first simulation showed that the data of Vogelzang et al. (2019) could be simulated accurately, but the data of Tsimpli et al. (2004) could not be accounted for. In the second model simulation, a function rather than a constant was used to determine the activation boost given to the referent associated with the grammatical subject of a sentence to reflect a gradual increase of certainty about the discourse topic, in line with experimental evidence from Spenader et al. (2009). The results showed that this simulation can account for the data of Vogelzang et al. (2019) and can partially account for of the data of Tsimpli et al. (2004). More specifically, null pronouns were accurately predicted to refer to the subject slightly more than half of the time. This indicates that a gradual increase in the activation of a recurring grammatical subject is a viable possibility, and something that should be seriously considered when modeling processes in which prominence in discourse plays an important role or when designing (linguistic) experiments. However, Tsimpli et al.'s (2004) findings for overt pronouns, which referred to the subject less than 10% of the time, were not replicated by the model. We will discuss three possible causes for this below.

First, Serratrice (2007) notes that Tsimpli et al.'s experimental results are not in line with the classical null-pronoun-refers-to-the-subject findings of, a.o., Carminati (2002). Additionally, the results from Tsimpli et al. (2004) show a stronger interpretation preference for overt pronouns than for null pronouns, which is also in contrast to Carminati (2002). Serratrice (2007, p. 233) suggests that this may be caused by the fact that "*In Carminati's experiment the subject and the object interpretation were presented in written form, while in this study the two alternatives were presented pictorially*". However, in Vogelzang et al.'s (2019) study the answers were also presented pictorially. Therefore, the pictorial presentation can not explain the differences in interpretation between Vogelzang et al.'s (2019) study, whose results are in line with Carminati (2002), on the one hand and the Tsimpli et al.'s (2004) study on the other hand. Nevertheless, there were some differences between the tasks that should be examined in more detail in future research, such as the type of pictures presented (pictures with actions, Tsimpli et al. vs. pictures without actions, Vogelzang et al.) and the number of answer possibilities (3 vs. 2, respectively).

A second possible explanation for the strong preference of overt pronouns to refer to the non-subject in the study of Tsimpli et al. (2004) could be related to aspects of the discourse that were not taken into account in the model. Discourse prominence was taken into account in terms of recency and frequency though standard ACT-R activation mechanisms, but it is possible that recency plays a bigger role than that, as it is known to influence the accessibility of a referent (Arnold, 1998; Givón, 1983). When examining the discourses in (1) and (2), we can see that Tsimpli et al. introduced the second referent at the end of the pre-critical clause, whereas Vogelzang et al. provided linguistic content (in (2) a direct object) in between the second referent and the end of the pre-critical clause. Thus, it may be possible that very recent referents hold a special status, which was not taken into account in the model.

A third possible explanation stems from the observation that for both null and overt pronouns participants showed fewer subject interpretations in the study of Tsimpli et al. (2004) compared to the study of Vogelzang et al. (2019). This might be related to the verbs used in the discourses, as verb bias or implicit causality (Garvey & Caramazza, 1974) can influence which referent (previous subject or non-subject) will likely be the actor in the continuation of the discourse. Similarly, an event-structure bias (Stevenson, Crawley, & Kleinman, 1994) of verbs could trigger the preference to continue the story with the end state of an action, which was the goal (non-subject) rather than the source (subject), of the verb in the pre-critical sentences. The model did not take any verb bias into account; potential effects of the verbs could be tested using a sentence completion task with the verbs used in both experiments.

Concluding, we investigated to what extent the variations in experimental results on the interpretation of Italian subject pronouns can be explained by the different discourses used in the experimental studies. Our simulations suggest that the discourse contexts used in the experiments crucially influence the interpretation of Italian subject pronouns. Thus, discourse prominence in terms of recency, frequency, and grammatical role seem to play an important role in the processing and interpretation of pronouns, which has to be taken into account when interpreting experimental results. Nevertheless, the model was not able to account for all data, and further research, both on the processing of discourse and on the influence of the specific task, is needed to investigate variations in experimental results on Italian pronoun interpretation.

References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*(4), 1036–1060. http://doi.org/10.1037/0033-295X.111.4.1036

Ariel, M. (1990). *Accessing noun-phrase antecedents*. London: Routledge.

Arnold, J. E. (1998). *Reference form and discourse patterns* (Doctoral dissertation). Stanford: Stanford University.

Beaver, D. (2004). The optimization of discourse anaphora. *Linguistics and Philosophy*, *27*(1), 3–56. http://doi.org/10.1023/B:LING.0000010796.76522.7a

Blutner, R. (2000). Some aspects of optimality in natural language interpretation. *Journal of Semantics*, *17*(3), 189–216. http://doi.org/10.1093/jos/17.3.189

Burzio, L. (1998). Is the Best Good Enough? Optimality and Competition in Syntax. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, & D. Pesetsky (Eds.), *Anaphora and Soft Constraints*. Cambridge, MA: MIT Press.

Carminati, M. N. (2002). *The processing of Italian subject pronouns* (Doctoral dissertation). Amherst, MA: University of Massachusetts.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: a source activation account. *Cognitive Science*, *25*(3), 315–353. http://doi.org/10.1016/S0364-0213(01)00039-8

Garvey, C., & Caramazza, A. (1974). Implicit causality in verbs. *Linguistic Inquiry*, *5*(3), 459–464.

Givón, T. (1983). Topic continuity in discourse: An introduction. In T. Givón (Ed.), *Topic Continuity in Discourse: A Quantitative Cross-language Study*. Amsterdam: John Benjamins Publishing Company.

Grosz, B. J., Joshi, A. K., & Weinstein, S. (1995). Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, *21*(2), 203–225.

Gundel, J. K., Hedberg, N., & Zacharski, R. (1993). Cognitive status and the form of referring expressions in discourse. *Language*, *69*(2), 274–307.

Hendriks, P., Englert, C., Wubs, E., & Hoeks, J. (2008). Age differences in adults' use of referring expressions. *Journal of Logic, Language and Information*, *17*(4), 443–466. http://doi.org/10.1007/s10849-008-9065-6

Hendriks, P., & Spenader, J. (2006). When Production Precedes Comprehension: An Optimization Approach to the Acquisition of Pronouns. *Language Acquisition*, *13*(4), 319–348. http://doi.org/10.1207/s15327817la1304_3

Hendriks, P., Van Rijn, H., & Valkenier, B. (2007). Learning to reason about speakers' alternatives in sentence comprehension: A computational account. *Lingua*, *117*(11), 1879–1896. http://doi.org/10.1016/j.lingua.2006.11.008

Kehler, A., Kertz, L., Rohde, H., & Elman, J. L. (2008). Coherence and coreference revisited. *Journal of Semantics*, *25*(1), 1–44. http://doi.org/10.1093/jos/ffm018

Prince, A., & Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Oxford, UK: Blackwell Publishing. http://doi.org/10.1002/9780470759400

Serratrice, L. (2007). Cross-linguistic influence in the interpretation of anaphoric and cataphoric pronouns in English–Italian bilingual children. *Bilingualism: Language and Cognition*, *10*(3), 225–238. http://doi.org/10.1017/S1366728907003045

Spenader, J., Smits, E.-J., & Hendriks, P. (2009). Coherent discourse solves the pronoun interpretation problem. *Journal of Child Language*, *36*(1), 23–52. http://doi.org/10.1017/S0305000908008854

Stevenson, R., Crawley, R., & Kleinman, D. (1994). Thematic roles, focusing and the representation of events. *Language and Cognitive Processes*, *9*, 519–548.

Tsimpli, I., Sorace, A., Heycock, C., & Filiaci, F. (2004). First language attrition and syntactic subjects: A study of Greek and Italian near-native speakers of English. *International Journal of Bilingualism*, *8*(3), 257–277. http://doi.org/10.1177/13670069040080030601

Van Rij, J., Van Rijn, H., & Hendriks, P. (2010). Cognitive architectures and language acquisition: A case study in pronoun comprehension. *Journal of Child Language*, *37*(3), 731–766. http://doi.org/10.1017/S0305000909990560

Van Rij, J., Van Rijn, H., & Hendriks, P. (2013). How WM load influences linguistic processing in adults: a computational model of pronoun interpretation in discourse. *Topics in Cognitive Science*, *5*(3), 564–80. http://doi.org/10.1111/tops.12029

Vogelzang, M. (2017). *Reference and Cognition: Experimental and computational cognitive modeling studies on reference processing in Dutch and Italian* (Doctoral dissertation). Groningen: University of Groningen.

Vogelzang, M., Foppolo, F., Guasti, M. T., Van Rijn, H., & Hendriks, P. (2019). Reasoning about alternative forms is costly: Comparing the processing of null and overt pronouns in Italian using pupillary responses. *Discourse Processes*. http://doi.org/10.1080/0163853X.2019.1591127

The model that knew too much: The interaction between strategy and memory as a source of voting error

Xianni Wang[1], John K. Lindstedt[1], Michael D. Byrne[1,2]
{xw48, j.k.l, byrne}@rice.edu
[1]Department of Psychological Sciences, [2]Department of Computer Science
6100 Main St., MS-25, Houston, TX 77005 USA

Abstract

This paper presents a family of models of a voting task we developed in order to investigate how errors arise from the interaction between strategy and knowledge. We crossed four task strategies with five different declarative memories and two visual strategies to yield a total of 40 different ACT-R models, and then tested the models through Monte Carlo simulations with 500 runs of each model. The findings suggest that some strategies work best when knowledge is incomplete, and that more task knowledge can lead to more errors in the recall process. These results highlight the importance of studying human error using a thorough exploration of the strategy space.

Keywords: ACT-R; error prediction; voting

Introduction

Human error is important for both theoreticians and practitioners to understand human cognition and performance. While theoreticians like to collect and classify errors, practitioners are more interested in their remediation, prevention, and even elimination. However, research on how to bridge the gap between the theoretical and practical areas is still not particularly common. In this paper, we describe an error prediction method that connects theoretical and practical work on human errors. This method accounts for human performance in routine behaviors using computational modeling and ultimately we hope can be used to predict human error before a system is implemented.

Error prediction methods are often based on traditional hierarchical task models (e.g., Annett & Duncan, 1967), which often assume that the processing system is explicitly hierarchical in structure and therefore break down complex tasks into hierarchies and sub-goals. Botvinick and Plaut (2004) suggested that hierarchical schemas and goals are not always necessary, at least in routine behavior. Instead, they presented a recurrent network model that uses recurrent connections within a network, which map from environmental inputs to action outputs, to represent an everyday task. However, Cooper and Shallice (2006) contrasted this recurrent network model with their more traditional, hierarchically structured interactive activation model. They criticized Botvinick and Plaut's recurrent network approach, describing a set of problems with the approach, such as its behavioral inflexibility, and concluded that hierarchical structures are still necessary and play a causal role in the control of behavior.

Another approach to studying human error is to create human performance models using ACT-R (Anderson, 2007). This goes one step beyond models based on a traditional hierarchical structure by using cognitive architectures. ACT-R is a computational cognitive architecture that simulates and integrates human cognition, attention, and motor behavior. This helps researchers to understand how people organize knowledge and produce behavior in different ways. There are several published ACT-R models that can make the same errors as people (e.g., Anderson, et al., 1998; Halbrügge, Quade, & Engelbrecht, 2015; Lebière, Anderson, & Reder, 1994; Trafton, Altmann, & Ratwani, 2011).

However, it is not easy to predict human error using ACT-R. First, there are many types of errors, but a human performance model usually only makes a specific kind of error. If there is a complex working system that contains several sub-tasks, it will take time and effort to create models that cover all possible errors. Second, in general, computational human performance models are fitted to and/or make predictions about average human behavior. However, predicting errors cannot simply be a question of fitting the mean, because even if the average person does not make an error, there may still be a substantial number who do.

Our domain for error modeling is voting. People usually think that filling out a ballot is a simple task, but, in reality, unintentional undervotes, overvotes, or votes for the wrong candidate are very common in almost all elections. An undervote occurs when the number of votes is less than the maximum number allowed in a race, and an overvote occurs when the number of votes is more than the maximum number allowed. One reason for all these errors lies in the poor designs of the ballots, which fail to support human perceptual and cognitive limitations. There is clear evidence that ballot design problems have affected the outcomes of multiple elections in the United States (Laskowski et al., 2004). For example, more than 2,000 votes intended for Gore were cast for Buchanan in Palm Beach County, Florida, during the 2000 elections due to the use of the infamous butterfly ballot (Wand et al., 2001).

A standard usability evaluation prior to deployment would likely detect poor designs and prevent errors. However, usability specialists are rarely asked to perform such tests prior to an election. Instead, election officials, who have little formal training or the expertise in assessing ballots, are left to the task. In addition, most elections in the U.S. are administered at the county level, and there are over 3,000 counties in the U.S. Within each county, there are often hundreds of different precincts, each with a slightly different ballot style, meaning that, for each national election, tens of thousands of ballot designs are deployed. This makes conducting a traditional usability test for every single ballot intractable due to the problem scale.

While it is impossible to perform usability testing on every ballot before every election deployment, some initial work has been done on predicting errors in voting tasks. In Greene (2010), an ACT-R model was presented that could sometimes make the same mistake that voters made in Sarasota, Florida in 2006. In this case, the first DRE screen contained one race, but there were two races presented on the second screen. This layout inconsistency led to 13.9% of votes being undervotes in the top race on the second screen. Greene modeled these first two screens to explore two voting strategies. The first strategy was to read the first screen from top to bottom before selecting a candidate, and then recall a useful location from the first screen to use to direct the visual search on the next screen. With this strategy, the model used the first screen to set expectations about where to find relevant landmarks (e.g., titles of races); it could then miss the critical top race on the second screen when the model extended those expectations from one screen to the next. The second strategy was to read both screens from top to bottom, without any recall. In contrast to the first strategy, the second strategy did not result in a critical top race undervote.

Greene's (2010) model offers a meaningful opportunity for computational human performance modeling to make a unique contribution to the voting field. However, this model does not reflect the full complexity of voting. Different voters almost certainly approach ballots differently. It is therefore critical that the models reflect not just one or two voting strategies, but the entire range of behaviors, so that specific interactions between voting strategies and ballot designs can be uncovered.

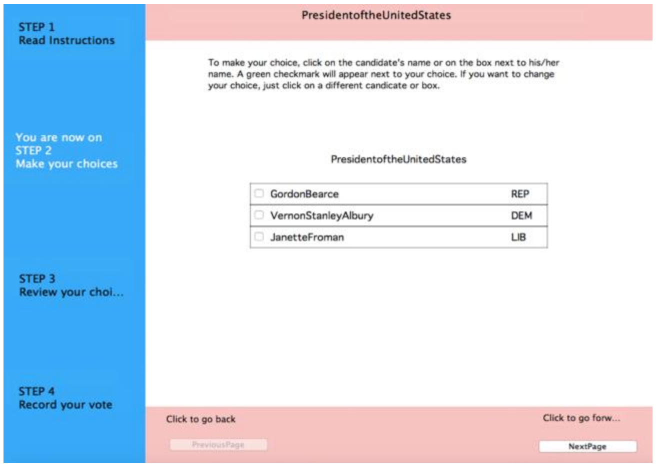
To capture more of the voting complexity, we developed a model-based approach that covers a family of voting strategies using ACT-R. For each model, the memory strategy, ballot knowledge, and visual search strategy were considered independently. Memory strategy represents how voters access their memories when they cast a vote; ballot knowledge defines voters' level of knowledge of the races and candidates; and visual search strategy indicates voters' visual directions when conducting a visual search. In total, our system consists of 40 voting models that crossed four memory strategies with five kinds of ballot knowledge and two visual strategies.

When it comes to visual search, humans have a remarkable ability to organize their perceptual inputs. The human visual system tends to group individual items in a visual image into larger structures under certain circumstances. This allows for the more efficient use of attention but sometimes leads to critical errors in executing a task. For example, the ballot used in Wisconsin in 2002 led to many unintentional votes. On this ballot, the race for governor was split across two columns, which led many voters to consider the two visual groups as two races. Many voters voted twice, once in each column. To handle situations like this, our system makes use of a visual grouping algorithm that enables more realistic visual scanning behaviors (Lindstedt & Byrne, 2018).

In short, our model-based system assessed a ballot layout with a family of voting models. Each voting model was tested multiple times, and the average across those runs was calculated. After running every ballot through each model repeatedly, all combinations of strategies and knowledge that generated high error rates were identified.

The Voting Task

Our system was implemented for an emulated voting task using the VoteBox task environment. Multiple experiments have been published in which human subjects voted using VoteBox (e.g., Everett, 2007; Everett, et al., 2008). This voting task contains 21 races that share a consistent layout (see Figure 1). The layout was designed to be easy to understand, with a relatively simple display that comprised the voting instructions, title of the race, candidates' names and party affiliations, a "previous page" button, and a "next page" button, all clearly arranged and presented across the screen.



To make your choice, click on the candidate's name or on the box next to his/her name. A green checkmark will appear next to your choice. If you want to change your choice, just click on a different candicate or box.

PresidentoftheUnitedStates

□ GordonBearce	REP
□ VernonStanleyAlbury	DEM
□ JanetteFroman	LIB

Figure 1: Mock ballot of a presidential race.

All versions of the model contain two phases. The first is a studying phase in which the model studies the display thoroughly to retain group information produced by the visual grouping algorithm. The second phase is a voting phase; after obtaining and storing group information during the first phase, the model now has expectations about where to look. It directs its gaze to the appropriate place and then makes a vote.

Modeling Strategies

A total of 40 voting models were developed. Each model includes a memory strategy, ballot knowledge, and a visual search strategy. To produce a comprehensive error prediction, multiple plausible versions were considered for each component. The details of each component are described in the sections below. We defined four memory strategies, five kinds of ballot knowledge, and two visual search strategies.

Memory Strategies

Voters have to remember their choices, and they access their memories in different ways. There are two primary memory strategies for simple form-filling tasks like voting: retrieval and recognition. Some voters can simply recall the names of those for whom they intend to vote, at least in some races. For example, many voters, when prompted, can retrieve from memory the candidate for whom they intend to vote in presidential elections. Other voters may instead scan the list of names first to try to recognize their preferred candidates. Some voters vote almost exclusively according to party affiliation but then have to remember which races, if any, have exceptions. Some voters may rely on party affiliation if they can neither recall for whom they intended to vote nor recognize any of the candidates' names on the list. While some voters may also write out a list and bring it into the voting booth, it is not clear how common this is, and it is, in fact, illegal in some jurisdictions. So, we did not consider this strategy.

Our models capture four memory strategies one could reasonably expect a voter to employ—a strictly retrieval-based strategy, a strictly recognition-based strategy, a retrieval-then-recognition contingency strategy, and a simple party-only look-up strategy (in case of exceptions to their default party). The first strategy represents the scenario in which the model first tries to retrieve the candidate's name from memory. If the model fails to recall the name, then it relies simply on a party affiliation. The second strategy considers the situations in which the model first tries to retrieve their choice, but, if the retrieval fails, it then scans the list of names and votes for the one it recognize. If recognition also fails, it votes by party affiliation. For the third strategy, the model does not even attempt to retrieve; rather, it scans the list of names to see if it can recognize any of them. If recognition fails, it votes by party. For the last strategy, the model simply votes based on party affiliation. It first retrieves the specific party affiliation for specific races, but, if the retrieval fails, default party affiliation becomes the only criterion. The last step of these four memory strategies—voting by default party affiliation—is used only when all the previous steps fail. The default party affiliation could be either the Democratic, Republican, or Libertarian Party.

Other memory strategies are certainly possible, but it is unclear how a voter could use the contents of their memory to vote in a meaningfully different manner without substantial overlap with one of the strategies listed above.

Ballot Knowledge

Voters have different levels of knowledge about the races and candidates. Some voters might have encoded all of the candidates' names, some may only know the names of candidates they intend to vote for, and some may only have parts of the intended candidates' names in their memories. In addition, ballot knowledge is not always easy to recall. Some voters may only remember their choices for the first few races because it is much more likely that voters will have more frequent exposure to top-of-the-ballot candidates. ACT-R represents situations like this using base-level activation, which reflects the recency and frequency of a specific memory.

Table 1: Ballot Knowledge

Ballot Knowledge	Candidates' Names	Activations for Intended Candidates
FULL-MEMORY	All candidates	Races 1 to 7: 0.7 Races 8 to 14: 0.6 Races 15 to 21: 0.5
ALL-ROLLOFF	Intended candidates only	Races 1 to 7: 0.7 Races 8 to 14: 0.6 Races 15 to 21: 0.5
ALL-PERFECT	Intended candidates only	All races: 0.8
MOST-ROLLOFF	70% of intended candidates	Races 1 to 3: 0.8 Races 4 to 7: 0.7 Races 8 to 11: 0.6 Races 12 to 15: 0.5 Race 16 to 21: Abstained
MOST-PERFECT	70% of intended candidates	Race 1 to 15: 0.8 Race 16 to 21: Abstained

Five ballot knowledge types were therefore created (see Table 1). First, we defined three levels of how many candidates' names were stored. The models could remember all candidates' names, only the intended candidates' names, or only the first 70% of the intended candidates' names. Then, we assigned two types of activations for intended candidates: roll-off activations and constant high-level activations. Models with roll-off activations are most familiar with the candidates for the first several races; then, as they progress down the ballot, their familiarity with candidates decreases. In the second condition—constant high-level activations—the models are highly familiar with all races to the same degree. Note that the various contents and activation levels of memory were not chosen as an exhaustive search of all possible knowledge held by voters, but rather as an illustrative sample of common voter scenarios—some voters have certainly done their homework extensively, while others have likely only decided “important” races.

Visual Search Strategies

While reading in a serial order is the most common search strategy, eye-tracking studies have demonstrated that it is not universal (Aaltonen, Hyrskykari, & Räihä, 1998; Fleetwood & Byrne, 2006). People scan displays in different ways: some readers read in a serial, item-by-item pattern, from one corner to its diagonal opposite; some people scan globally and read all the bold, large, or colored headers first; and some simply prefer to scan randomly.

Two visual search strategies were used when looking for candidates: a serial search and a random search. The serial search strategy is a serial item-by-item search with a left-to-right, top-to-bottom pattern. With the random search strategy, the models conduct a random search.

Model Evaluation

Method

The first issue to address is the number of Monte Carlo replications. We used the approach outlined in Byrne (2013) based on confidence intervals. We expected the overall error rate generated by the model to be around 5% and wanted the 95% confidence intervals for the model predictions to be no wider than 2% in either direction. The table in Byrne (2013) shows this requires 457 model runs; we ran 500 per model to be slightly more conservative.

Error Predictions

For each model run, the ballot, as completed by the model, was compared with the “intent” initialized at the beginning of the run, and any discrepancies were noted as errors. Errors occurred across the entire voting process. The model might have retrieved an unintended name, recognized an unintended name, or failed to retrieve and then recognized an unintended name. For the model that simply made votes based on party affiliation, it may have retrieved an unintended party. The model may even have failed to retrieve and/or recognize an intended name, and then have voted by default party affiliation. We used Democratic as the default party affiliation for this model evaluation; however, intended candidates’ party affiliations did not always match the default party affiliation. The model occasionally also mis-clicked on the name above or below the intended name.

Overall, our models generated an average 5% error rate across all voting models. This is somewhat higher than

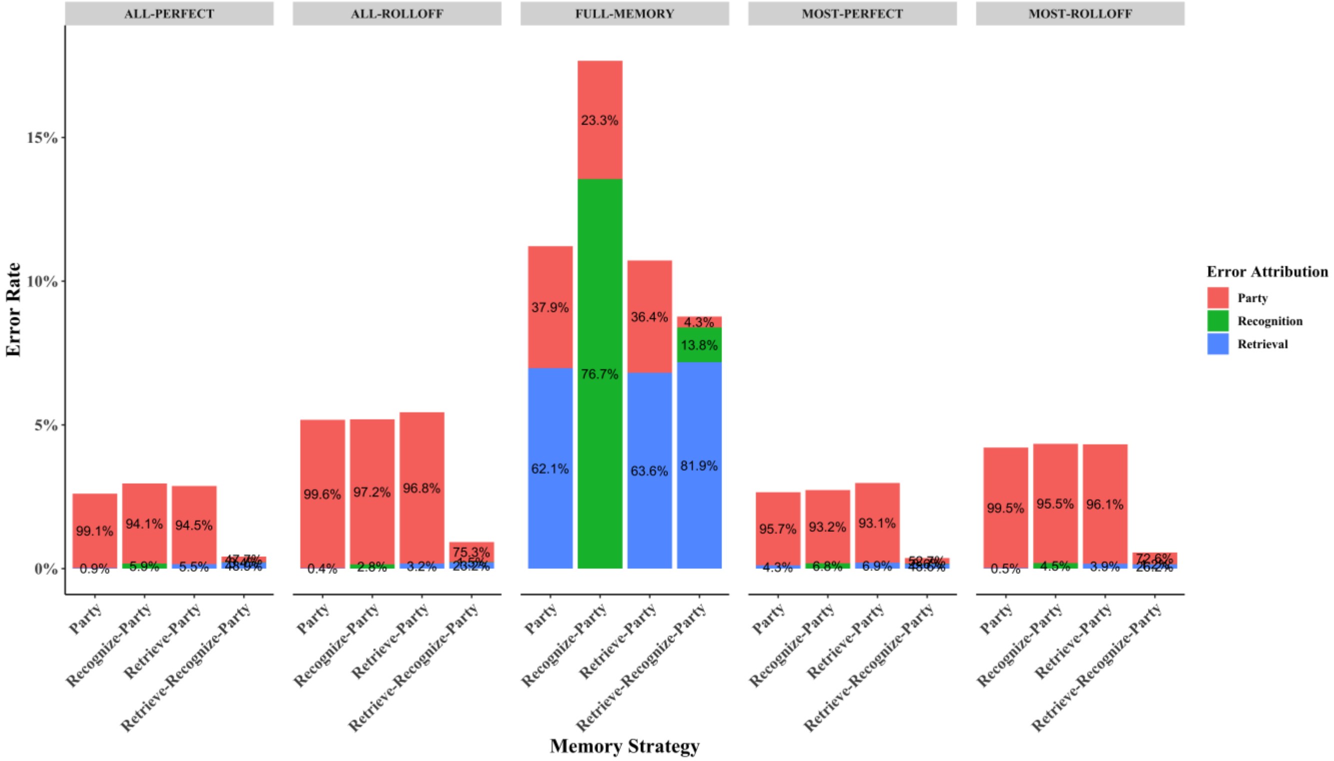


Figure 2: Interaction between memory strategy and ballot knowledge in voting error rates. The bars show voting errors. The five ballot knowledge types are written along the top; each shows the error rates for the four memory strategies. The three colors of the bars indicate the three kinds of processes the model used when it made an error. Red, green, and blue represent the party affiliation, recognition, and retrieval, respectively.

observed human error rates of around 1.5% (e.g., Everett, et al. 2008) and unfortunately it is impossible to compare our models directly to human data since we cannot know the strategies used by people in such studies. Instead, we are interested in how the model strategies interact to produce errors.

Differences in error rates with visual strategies were not found, which means that using either a serial or a random scanning pattern did not affect the voting results. The main story here is therefore about memory strategy and ballot knowledge. We observed differences in voting errors based on the interaction between voting strategy and ballot knowledge.

Figure 2 presents five groups of bars that represent the error rates of the five kinds of ballot knowledge. For each type of ballot knowledge, the percentages of the errors for the four memory strategies are displayed. For the FULL-MEMORY condition, the model generated 9% more errors than the other four ballot knowledge types. The model also generated more errors with roll-off activations for intended candidates. For the MOST-ROLLOFF and ALL-ROLLOFF conditions, the voting model was 2% more likely to make errors than with the MOST-PERFECT and ALL-PERFECT conditions. Additionally, for these four levels of ballot knowledge other than FULL-MEMORY, there were clearly fewer errors with the three-step “retrieve-recognize-party” memory strategy.

We then investigated the error attributions for each vote to determine which process the model was using when it made an error. In Figure 2, each bar is partitioned into three colors, which represent the three kinds of process the model was using when it made an error: retrieval, recognition, or party affiliation. For the FULL-MEMORY condition, most of the errors occurred in the recognition and/or retrieval processes. Within FULL-MEMORY, 7% more voting errors were generated with the “recognize-party” memory strategy. However, for the other four kinds of ballot knowledge, differences in error attributions with memory strategies were not apparent; most of the errors were generated in the last steps—voting by party affiliation.

Discussion

The error predictions indicate that extra ballot knowledge actually led to more errors, especially with the involvement of recognition. Common sense would suggest that a broader knowledge base should help to mitigate mistakes, but this is not always the case. Another example that also suggests that a strategy works best when knowledge is incomplete is the recognition heuristic. The recognition heuristic describes a situation where, if one of two objects is recognized and the other is not, the recognized object is more likely to be selected (Goldstein & Gigerenzer, 2002). This strategy requires ignorance to make a choice—if people know everything or nothing about the options, it simply does not work. For example, for the question “which city has a larger population?” most people choose Dublin over Nenagh since they can recognize Dublin only, but it is harder for people to make a selection if the choices become San Diego and San Antonio, as they are more likely to recognize both of these cities. Similarly, in the voting task in our study, the models knew everything in the FULL-MEMORY condition, including both the intended and unintended candidates’ names. Thus, compared to the other four ballot knowledge types, the memory strategies did not work well with FULL-MEMORY, and more errors occurred in the recognition processes.

Because of the more frequent recognition errors, one thing we can expect with the FULL-MEMORY condition is a greater impact of candidate name order. Voters who cannot recall their intended candidate’s name must scan the list of names and see if they can recognize any, and their choices can be biased by the order in which candidates’ names appear on the ballot (Miller & Krosnick, 1998). In our study, the model with “recognize-party” memory strategy checks each candidate, sees if it recognizes, and if so, votes for it. Since some voters use top-to-bottom visual search, an advantage for the top candidate can be predicted.

Another finding has to do with the interaction between task knowledge and recall performance. Schooler and Anderson (1997) suggested an association between the number of choices and recall performance, positing that the more choices we have, the more likely we are to make a recall error at each name. We observed the same relationship in our models. The FULL-MEMORY condition contains both intended names and unintended names, and the models could either retrieve an intended name or an unintended name for each race in that condition. It was therefore more likely to make errors in the retrieval process since incorrect answers are available. However, with the other four ballot knowledge types, there are only intended names available in memory. Wrong names were therefore less likely to be retrieved with these four levels of knowledge.

We can also conclude from the error predictions that the three-step “retrieve-recognize-party” memory strategy had a better performance than the two-step memory strategies. As can be seen in Figure 2, a large portion of the errors came from the last steps, voting by party affiliation, across five levels of knowledge. Comparing to the two-step strategies, the additional one step prevented errors that could be made in the last step, and so the least amount of errors was generated with the three-step memory strategy.

Note that the errors made here are not the result of poor ballot design. However, we believe that further interactions, those between strategy, knowledge, and ballot design, will show how the visual layout of the ballot can influence error rates. Poor layouts may not induce all voters into error, but differentially affect those who use particular strategies. Furthermore, we believe that these kinds of errors are not limited only to filling out ballots, but likely occur in other tasks that are essentially form fill-in, such as interacting with electronic health records.

Our model-based system represents the first use of ACT-R as an error prediction tool to diagnose if there are particular combinations of strategies that lead to error. The idea that one can understand the error space by modeling only one strategy or predicting mean behavior is likely to miss critical combinations of factors that produce errors. Our results demonstrate that subtle interactions between strategy and knowledge can have substantial effects on error rates. Thus, it is critical to consider multiple combinations of both when attempting to model errors, even in a task that appear as simple as voting.

Acknowledgments

This research was supported by grant #CNS-1550936 from the National Science Foundation. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of NSF, the U.S. Government, or any other organization.

References

Aaltonen, A., Hyrskykari, A., & Räihä, K. J. (1998). 101 spots, or how do users read menus? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 132-139). ACM Press/Addison-Wesley Publishing Co.

Anderson, J.R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*(4), 341-380.

Annett, J., Duncan, K.D., 1967. Task analysis and training design. *Journal of Occupational Psychology 41*, 211–221.

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111*(2), 395-429.

Byrne, M. D. (2013). How many times should a stochastic model be run? An approach based on confidence intervals. In *Proceedings of the 12th International Conference on Cognitive Modeling* (pp. 445-450). Ottawa: Carleton University.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review, 113*(4), 887-916.

Everett, S. P. (2007). *The usability of electronic voting machines and how votes can be changed without detection* (Doctoral dissertation). Rice University, Houston, TX.

Everett, S. P., Greene, K. K., Byrne, M. D., Wallach, D. S., Derr, K., Sandler, D., & Torous, T. (2008). Electronic voting machines versus traditional methods: Improved preference, similar performance. *Human Factors in Computing Systems: Proceedings of CHI 2008* (pp. 883-892). New York: ACM.

Fleetwood, M. D., & Byrne, M. D. (2006). Modeling the visual search of displays: A revised ACT-R model of icon search based on eye-tracking data. *Human-Computer Interaction*, *21*(2), 153-197.

Goldstein, D. G., & Gigerenzer, G. (2002). Models of ecological rationality: The recognition heuristic. *Psychological Review*, *109*(1), 75.

Greene, K. K. (2010). *Effects of Multiple Races and Header Highlighting on Undervotes in the 2006 Sarasota General Election: A Usability Study and Cognitive Modeling Assessment* (Doctoral dissertation). Rice University, Houston, TX.

Halbrügge, M., Quade, M., & Engelbrecht, K. P. (2015). A predictive model of human error based on user interface development models and a cognitive architecture. In *Proceedings of the 13th International Conference on Cognitive Modeling* (pp. 238-243). University of Groningen, Groningen, the Netherlands.

Laskowski, S. J., Autry, M., Cugini, J., Killam, W., & Yen, J. (2004). *Improving the usability and accessibility of voting systems and products*. NIST Special Publication 500-256. Retrieved from https://user-centereddesign.com/files/NISTHFReport.pdf.

Lebière, C., Anderson, J. R., & Reder, L. M. (1994). Error modeling in the ACT-R production system. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 555-559). Erlbaum Hillsdale, NJ.

Lindstedt, J. K., & Byrne, M. D. (2018). Simple agglomerative visual grouping for ACT-R. In I. Juvina, J. Houpt, & C. Myers (Eds.), *Proceedings of the 16th International Conference on Cognitive Modeling* (pp. 68-73). Madison, WI: University of Wisconsin.

Miller, J. M., & Krosnick, J. A. (1998). The impact of candidate name order on election outcomes. *Public Opinion Quarterly*,*62*(3), 291-330.

Schooler, L. J., & Anderson, J. R. (1997). The role of process in the rational analysis of memory. *Cognitive Psychology*, *32*(3), 219-250.

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2011). A memory for goals model of sequence errors. *Cognitive Systems Research*, *12*(2), 134-143.

Wand, J. N., Shotts, K. W., Sekhon, J. S., Mebane, W. R., Herron, M. C., & Brady, H. E. (2001). The butterfly did it: The aberrant vote for Buchanan in Palm Beach County, Florida. *American Political Science Review*, *95*(4), 793-810.

Neural Principles for Modeling Relational Reasoning: Lesson learned from Cognitive Neuroscience

Julia Wertheim (wertheim@cs.uni-freiburg.de)
Cognitive Computation Lab, Technical Faculty, Albert-Ludwigs-Universität Freiburg,
Georges-Köhler-Allee 79, 79110 Freiburg, Germany

Marco Ragni (ragni@cs.uni-freiburg.de)
Cognitive Computation Lab, Technical Faculty, Albert-Ludwigs-Universität Freiburg,
Georges-Köhler-Allee 52, 79110 Freiburg, Germany

Abstract

Cognitive models serve the purpose of implementing theories of human cognition and give the opportunity to simulate reasoning processes for comparing them to participant data. Relational reasoning is particularly relevant, because it is closely connected to spatial navigation and planning. In modeling relational reasoning, findings from neuroscience have been largely neglected. As we are showing, the connection between neuroimaging and cognitive modeling has been elementary so far. We aim at bridging the gap between the neurocognitive correlates of relational reasoning and cognitive models thereof. Computational models and architectures, as well as the recent neuroimaging literature investigating relational reasoning are reviewed. By identifying functional modules, we postulate the neuroscientific loci which a modeler aiming at simulating reasoning should consider before conceptualizing a neurocognitive model of relational reasoning.

Keywords: Cognitive modeling, Relational reasoning, Neuroimaging, Cognitive architecture

Introduction

Cognitive models enable the testing of cognitive theories and the comparison to psychological findings. In the last decade, the interest in biologically plausible cognitive modeling has been rising, not least because of transnational research projects such as the Blue Brain Project (Markram, 2006). Especially interesting is the modeling of higher cognitive processes such as relational reasoning, usually taking the form of premises like “Tom is to the right of Sally. Sally is to the right of George.”, from which the inference “Tom is to the right of George.” can be derived. For successfully solving this task, participants are asked to infer explicit knowledge about the objects’ relations to each other, which is implicitly given in the premises. Relational reasoning is closely linked to spatial navigation and hence motor function, as well as to analogy and language processing. Thereby, it is more versatile than other reasoning types and particularly promising for cognitive modeling since multiple cognitive abilities are recruited. The expected findings of modeling relational reasoning could greatly contribute to fields such as Brain-Computer-Interfaces or medical diagnostics. For example, if a patient suffers from a brain lesion, the only information available so far is a potential function loss in cognitive abilities associated with the respective region. Detailed information about the region’s function in terms of more complex cognitive abilities such as reasoning is not yet available. Biologically plausible cognitive model could provide details about the wide-ranging cognitive deficits resulting from the loss when informed by the lesion site. Hence, investigating the connection between models of cognition and neuroscience are beneficial for developing medically relevant models of neuropathology and diagnostic purposes.

But how *do* we go about investigating human cognition, specifically relational reasoning? According to Marr’s analysis, there are three levels to be considered (Marr, 1982, see Figure 1). On the computational level, the strategic aim of the cognitive effort is evaluated. This involves a formal model or theoretical framework of relational reasoning, such as mental model theory, as well as cognitive architectures in which these can be implemented. On the algorithmic level, human performance is assessed. This involves reasoning effects and models describing and explaining the processes. On an implementation level, the ‘hardware’ in which cognitive process are implemented is considered, namely neuroanatomy. Regarding relational reasoning, this results in finding the neural correlates of these processes and assessing their neuroanatomical feasibility.

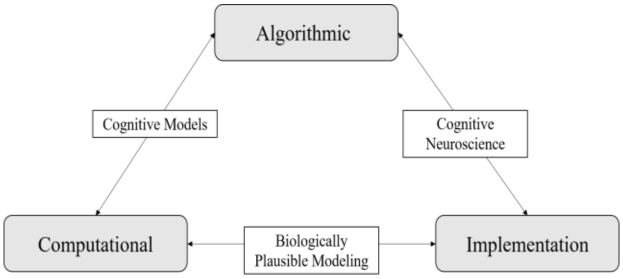


Figure 1: Representation of the analysis levels.

By modeling cognitive processes, we predominantly work on the computational level, but they are inevitable intertwined. In the case of relational reasoning, cognitive models are described to solve reasoning tasks which have previously been tested on human participants, hence the algorithmic and computational level are connected. In cognitive neuroscience, the algorithmic and implementation levels are conjoined by identifying the relevant brain areas. As we are going to show in the following, the link between the implementation and computational level is currently underdeveloped, although both levels bear important insights for the other. Our article serves as a resource for modelers aiming at the development of a cognitive model for relational reasoning which is based on neuroscientific insights.

Mutually informing each other: cognitive neuroscience and cognitive modeling

Cognitive neuroscience can greatly benefit from the insights of cognitive models, for modeling provides more sophisticated analyses of cognitive processes. On this basis, neuroscientific hypotheses can be formulated more accurately and tested on more precise levels of analysis by revealing hidden cognitive processes and fostering more accurate study designs (Forstmann et al., 2011). Conversely, cognitive modeling likewise benefits from neuroscientific findings in using them to restrain and inform the set-up and conceptualization of cognitive models, thereby making them more viable (Forstmann et al., 2011). This includes systematic reviews of cognitive models, some of which can be favored based on neuroscientific plausibility (Forstmann et al., 2011).

In this article, we aim at bridging the gap between cognitive models and insights from cognitive neuroscience about the neural underpinnings of relational reasoning. For this, we systematically review cognitive models and neuroimaging studies of relational reasoning and identify the most central brain regions. We characterize the regions in terms of their functionality to the task and summarize by establishing a neuroscience-based standard functions and brain regions required for modeling relational reasoning.

Cognitive Models of Relational Reasoning

Regarding the algorithmic level, Friemann and Ragni (2018) have recently published a review of spatial relational reasoning models. Their collection of models was adapted to review the current state of the art, Further, we conducted an online researched via Google Scholar and Pubmed. As Table 1 displays, the models vary greatly along different parameter such as the number of dimensions of processable relations and their inclusion of findings from cognitive neuroscience. The listed models are going to be categorized in terms of the aforementioned dimensionality, working memory capacity and whether findings from cognitive neuroscience are incorporated in the model.

Dimensionality

The lowest dimensionality of one, enabling the processing of relational dichotomies is offered by the models of Schlieder and Berendt (1998), Bara, Bucciarelli and Lombardo (2001), Hummel and Holyoak (2001), Morrison et al., (2004), Krumnack, Bucher, Nejasmic, Nebel and Knauff (2011) and Dietz, Hölldobler and Höps (2015). An additional dimension, allowing for, e.g., the processing of cardinal directions, is featured by the models by Ioerger (1994), Schultheis and Barkowsky (2011), Wertheim and Stewart (2018) and Kounatidou, Richter and Schöner (2018). Only the models by Johnson-Laird and Byrne (1991) and Ragni and Knauff (2013) provide a three-dimensional space in which relational reasoning operations are represented.

Working memory

Concerning working memory, most models exhibit a limited capacity inspired by human processing. In the models by Schlieder and Berendt (1998), Dietz et al. (2015) and Kounatidou et al. (2018), the capacity of the working memory is unspecified and therefore not explicitly adapted to human performance.

Table 1: Overview of cognitive models of spatial relational reasoning.

Authors	Dim.	WM	ND
Johnson-Laird & Byrne, 1991	3	Lim	✗
Ioerger, 1994	2	Lim	✗
Schlieder & Berendt, 1998	1	n/a	✗
Bara et al., 2001	1	Lim	✗
Hummel & Holyoak, 2001	1	Lim	✗
Morrison et al., 2004	1	Lim	✓
Krumnack et al., 2011	1	Lim	✗
Schultheis & Barkowsky, 2011	2	Lim	✓
Ragni & Knauff, 2013	3	Lim	✗
Dietz et al., 2015	1	n/a	✗
Wertheim & Stewart, 2018	2	Lim	✓
Kounatidou et al., 2018	2	n/a	✓

Note. Dim.: Number of dimensions; WM: Working memory; Lim: Limited capacity; ND: Inclusion of neuroscience data.

Inclusion of neuroscience data

Concerning the implementation of neuroscience data, only few models appear to be relevant. The model by Morrison et al. (2004) includes lesion patient data, whereas the model by Schultheis and Barkowsky (2011) is explicitly based on the modularity hypothesis. Apart from that, only the models by Wertheim and Stewart (2018) and Kounatidou et al. (2018) run on artificial neural networks which are (partially) based on the mechanisms of actual neurons.

We conclude that the inclusion of neuroscience data has not yet been widely used in the development of cognitive models. Nonetheless, cognitive models can theoretically be implemented in current cognitive architectures such as the Turing-complete ACT-R. Also, this has already been done by, e.g., Wertheim and Stewart (2018) in the Neural Engineering Framework (NEF). Hence, an online research was conducted via Google Scholar and Pubmed to review cognitive architecture providing a programming framework and to investigate to what extend insights from neuroscience have been used to restrain frameworks or provide predictions. A notable review has been on cognitive architectures has been published by Samsonovich (2010), but not all aim at biological plausibility.

Cognitive Architectures

Some frameworks consider brain function on either a level of restraining implementation possibilities or in modeling neurocognitive processes. These include the architectures 4CAPS (Just, Carpenter & Varma, 1999) and conceptually also the precursor 3CAPS (Just & Carpenter, 1992) by approximating the BOLD response. ACT-R (Anderson, 2007) and EPIC (Meyer & Kieras, 1999) incorporate anatomically and functionally plausible correspondents to brain regions. The NEF (Eliasmith, 2013) simulates neuronal activity and connectivity, whereas SOAR (Newell, 1992) restrains working memory to a neurobiologically plausible time span. CLARION (Sun, 2002) is based upon the modularity hypothesis and Sigma (Rosenbloom, 2013) features neural networks. In Table 2, we evaluate the cognitive architectures towards features essential to biologically plausible computation of relational reasoning.

Table 2: Overview of cognitive architectures suitable for processing relational reasoning

Model	Module	BOLD
3CAPS	✓	✗
SOAR	✓	✗
4CAPS	✓	✓
EPIC	✓	✓
CLARION	✓	✗
ACT-R	✓	✓
NEF	✓	✗
Sigma	✗	✗

Note. Module: Does it feature separate interacting modules?; BOLD: BOLD function predicted?

Modularization

Modularization of functional components has been a common practice when designing cognitive architectures. This is based upon the modularity hypothesis stemming from evolutionary psychology which claims that cognition is facilitated by function-specific brain regions serving as modules (Fodor, 1985). We have found that almost all architectures share this basic trait, except for Sigma (Rosenbloom, 2013).

BOLD prediction

A common approach in biologically plausible cognitive modeling is the prediction and approximation of the BOLD response derived from fMRI studies. So far, his has only been accomplished in the framework 4CAPS (Just et al., 1999) and ACT-R (Anderson, 2007).

Functional Brain Regions for Relational Reasoning

The meta-analysis by Wertheim and Ragni (2018) examines the neural correlates of relational reasoning and was used to identify the brain regions active during task solving. In the following, the regions are examined based on their involvement in cognitive processes with regard to their relevance to cognitive modeling.

Frontal lobe

The most wide-spread activation was found in the bilateral, but mostly left prefrontal cortex. According to O'Reilly and Munakata (2000), this region is functionally responsible for active and flexible maintenance of complex mental representations, as well as goal-directed executive control, especially regarding the monitoring of overall processing (Eriksson et al., 2015). Further, a left-sided activation in reasoning has been previously assumed and supported by lesion studies. For example, Goel et al. (2006) showed that left-, in contrast to right-sided lesions hinder participants to correctly decide whether determinate tasks are correct.

Particularly relevant to relational reasoning is the dorsolateral prefrontal cortex and middle frontal gyrus (DLPFC/MFG, BA 9, 8). It is active during executive functioning and cognitive monitoring (Prabhakaran, Smith, Desmond, Glover & Gabrieli, 1997) and in maintaining multiple relations (Waltz et al., 1999), which is proposed to translate to the entertainment of a mental model and integration of several relations (Wertheim & Ragni, 2018). For example, the MFG is used in the architecture ACT-R serving as a declarative memory module (Anderson, 2007). BA 6 (Supplementary motor area, SMA) is involved in task planning (Hanakawa et al., 2002), whereas the precentral gyrus (PreCG, BA 9, 8) facilitates attention management (Acuna, Eliassen, Donoghue & Sanes, 2002).

Parietal lobe

The posterior parietal cortex (PPC) is typically associated with the (repetitive) processing of spatial information and scenarios, such as mental rotation (O'Reilly & Munakata, 2000). Specifically, activation was found in the bilateral superior parietal lobule (SPL), inferior parietal lobule (IPL), precuneus (PCUN, BA 7, 40). It is involved in executive working memory and sustained attention (Koenigs, Barbey, Postle & Grafman, 2009) and linked to the selection of the attention focus (Awh, Vogel & Oh, 2006). From a modeling perspective, this region is specifically involved in the construction and manipulation of mental models (Ragni, Franzmeier, Maier, & Knauff, 2016). Concerning the precuneus, neuroimaging studies have found its specific involvement in abstract tasks, as well as episodic memory retrieval (Cavanna & Trimble, 2006). Henceforth, the PPC facilitates a mental space in which model representation and manipulation takes place.

Basal ganglia

The right claustrum shares extensive structural connections to the prefrontal cortex (Ullman, 2006). In a computational sense, the selection of actions is assigned to this region (O'Reilly & Munakata, 2000). This assumption is supported by further neuroimaging studies of reasoning, e.g., Jia et al. (2011) assign rule induction to the basal ganglia. From a computational perspective, O'Reilly (2006) specified its role of gating of mental representations coming from the PFC.

Table 3: Overview of brain regions central to relational reasoning found by Wertheim and Ragni (2018).

Regions	BA	Function
SPL	7	Processing spatial information, construction and manipulation of mental models
IPL	40	
SMA	6	Task planning
DLPFC	9, 8, 46	Executive functioning, cognitive monitoring, maintaining of complex information, attention management Declarative memory
Claustrum	-	Action selection, representation gating

Note. BA: Brodmann area; SPL: Superior parietal lobule, IPL: Inferior parietal lobule, SMA: Supplementary motor area, DLPFC: dorsolateral prefrontal cortex.

Occipital lobe

Although Wertheim and Ragni (2018) did not find any significant clusters in the occipital lobe, processing-wise it is interesting because it shares connections with the PPC (Culham & Kanwisher, 2001). As it is active during the processing and abstraction of visual input (O'Reilly & Munakata, 2000), it should be considered for processing visual inputs and redirecting information to the PPC.

Discussion

In this article, we reviewed the current state of the art of cognitive models, architectures and the neuroscience of relational reasoning and hence provide a guideline for programmers aiming at building biologically plausible models of relational reasoning. Although there exists a considerable selection of models explaining the cognitive mechanisms underlying reasoning on the algorithmic level, only few have been implemented. Hence, we continued by reviewing cognitive architectures and found that there have been various approaches to including neuroscience results by either restraining programming environments or deriving predictions relevant for neuroscience. Nonetheless, the interface between cognitive architectures and cognitive neuroscience is sparse and only a synthesis of different approaches could foster the current state of the art. Concerning cognitive models, this would be by assuming three dimensions on which relations can be represented, a psychologically plausible constraint on working memory, was well as the potential implementation with neural networks. Regarding cognitive architectures, starting points are anatomically and functionally specified modules, as well as the prediction of the BOLD response (for examples see, Anderson et al., 2008 and Borst & Anderson, 2015).

Cognitive modeling would benefit from conjoining preexisting approaches to integrating neuroscience. Similarly, cognitive neuroscience can benefit by informing experimental set ups from cognitive simulations. For example, O'Doherty, Hampton and Kim (2007) developed model-based neuroimaging for correlating assumed cognitive processes with actual scanning data. Concerning our review of neuroimaging studies, the most relevant regions which need to be considered in cognitive models have been identified and examined for their involvement in functions relevant to relational reasoning (see Figure 2). These include the PPC for abstracting and processing spatial information and working memory and the basal ganglia for action selection and information gating. The basal ganglia have already been implemented as an action selection system in the NEF (Senft, Stwart, Bekolay, Eliasmith & Kröger, 2016), whereas the imaginal buffer in ACT-R serves as a correspondent to the PPC (Anderson, 2007). Further identified regions are the SMA for task planning, and DLPFC for meta-cognitive functions such as cognitive monitoring and attention management. This region-function matching can inform cognitive models by a precise selection of actions and associated brain regions. Since we did not only identify the relevant regions but also their function specifically in relational reasoning, modelers can either only include the regions' respective function or decide to consider neuroanatomical details as well.

We initially claimed that there exists a gap between the implementation and computational level of investigating cognition. We have contributed to closing this gap by identifying biologically relevant features of architectures which should be merged and extended, as well the most functionally relevant brain regions from neuroimaging. By this, we have established a first example of a necessary precondition to neurocognitive modeling and proposed guidelines from which both domains can benefit. Cognitive models could be improved by this localization of activation foci by constraining the models based on the cognitive succession of cognitive demands needed to fulfill a task which is examined by data derived from neuroimaging and corresponding cognitive theories.

From a practical perspective, biologically plausible cognitive models could be used for diagnostic purposes in medical environments. From these models, we could infer more detailed cognitive impairments in higher cognitive functions. So far, it is only possible to identify basic impairments following the damage of brain tissue, such as impairments in processing language and working memory. By developing a more detailed account of the neurological, as well as cognitively functional subunits of the mind-brain, diagnoses and decisions can be improved and more elaborate restorative and preventive therapies can be developed. Potentially, this program might develop into a resource for the structure-function mapping between brain regions and their involvement in specific tasks which would foster the mutual exchange between these two vibrant fields of research, as well as increase the practical usage of neuroscientific data for cognitive modeling.

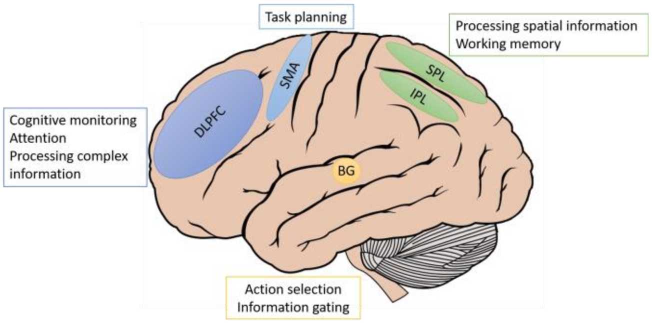


Figure 2: Representation of the brain regions active during relational reasoning and associated functions thereof.

The merging of the computational and implementation domains can be deepened and fostered by analyzing further meta-analyses on the neurocognitive correlates of reasoning tasks which can be theoretically or have been practically implemented in cognitive architectures. This would improve the specificity of determining which brain regions subserve cognitive functions, thus merging the approaches by brain mapping via neuroimaging and bypassing the difficulties of small sample sizes in neuroimaging studies (by meta-analyses) and cognitive modeling (by task specificity). Another domain of future work could be the more in-depth connection between implementation and computation by investigating the structural properties of the respective regions (e.g., arrangement of layer cells, interaction between inhibitory and excitatory mechanisms) for examining potential structure-function dependencies relevant to cognitive architectures per se and for spatial relational reasoning in particular.

Acknowledgments

This research was supported by the BrainLinks-BrainTools Cluster of Excellence funded by the German Research Foundation (DFG, grant number EXC 1086) to JW, a Heisenberg scholarship (RA 1934/3-1 and RA 1934/4-1) to MR and the DFG priority program New Frameworks of Rationality (SPP 1516).

References

Acuna, B. D., Eliassen, J. C., Donoghue, J. P., & Sanes, J. N. (2002). Frontal and parietal lobe activation during transitive inference in humans. *Cerebral Cortex*, *12*(12), 1312-1321.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?*. New York: Oxford University Press.

Anderson, J. R., Carter, C. S., Fincham, J. M., Qin, Y., Ravizza, S. M., & Rosenberg-Lee, M. (2008). Using fMRI to test models of complex cognition. *Cognitive Science*, *32*(8), 1323-1348.

Awh, E., Vogel, E. K., & Oh, S. H. (2006). Interactions between attention and working memory. *Neuroscience*, *139*(1), 201-208.

Bara, B. G., Bucciarelli, M., & Lombardo, V. (2001). Model theory of deduction: A unified computational approach. *Cognitive Science*, *25*(6), 839-901.

Borst, J. P., & Anderson, J. R. (2015). Using the ACT-R Cognitive Architecture in combination with fMRI data. In *An introduction to model-based cognitive neuroscience* (pp. 339-352). Springer, New York, NY.

Cavanna, A. E., & Trimble, M. R. (2006). The precuneus: a review of its functional anatomy and behavioural correlates. *Brain*, *129*(3), 564-583.

Charlton, R. A., Barrick, T. R., Lawes, I. N. C., Markus, H. S., & Morris, R. G. (2010). White matter pathways associated with working memory in normal aging. *Cortex*, *46*(4), 474-489.

Dietz, E.-A., Hölldobler, S., & Höps, R. (2015). A computational logic approach to human spatial reasoning. In *IEEE symposium series on computational intelligence* (pp. 1627–1634). IEEE.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford, UK: Oxford University Press.

Eriksson, J., Vogel, E. K., Lansner, A., Bergström, F., & Nyberg, L. (2015). Neurocognitive architecture of working memory. *Neuron*, *88*(1), 33-46.

Fodor, J. A. (1985). Precis of the modularity of mind. *Behavioral and Brain Sciences*, *8*(1), 1-5.

Forstmann, B. U., Wagenmakers, E. J., Eichele, T., Brown, S., & Serences, J. T. (2011). Reciprocal relations between cognitive neuroscience and formal cognitive models: opposites attract?. *Trends in Cognitive Sciences*, *15*(6), 272-279.

Friemann, P., & Ragni, M. (2018). Cognitive Computational Models of Spatial Relational Reasoning: A Review. *In* Thrash, Kelleher, & Dobnik (Eds.). *The 3rd Workshop on Models and Representations in Spatial Cognition (MRSC-3).*

Goel, V., Tierney, M., Sheesley, L., Bartolo, A., Vartanian, O., & Grafman, J. (2006). Hemispheric specialization in human prefrontal cortex for resolving certain and uncertain inferences. *Cerebral Cortex*, *17*(10), 2245-2250.

Hanakawa, T., Honda, M., Sawamoto, N., Okada, T., Yonekura, Y., Fukuyama, H., & Shibasaki, H. (2002). The role of rostral Brodmann area 6 in mental-operation tasks: an integrative neuroimaging approach. *Cerebral Cortex*, *12*(11), 1157-1170.

Hummel, J. E., & Holyoak, K. J. (1996). LISA: A computational model of analogical inference and schema induction. In G. W. Cottrell (Ed.), In *Proceedings of the 18th Annual Conference of the Cognitive Science Society* (pp. 352-357). Hillsdale, NJ: Erlbaum

Ioerger, T. R. (1994). The manipulation of images to handle indeterminacy in spatial reasoning. *Cognitive Science, 18*(4), 551-593.

Johnson-Laird, P. N. & Byrne, R. M. J. (1991). *Deduction*. Hillsdale, NJ: Erlbaum.

Jia, X., Liang, P., Lu, J., Yang, Y., Zhong, N., & Li, K. (2011). Common and dissociable neural correlates associated with component processes of inductive reasoning. *NeuroImage, 56*(4), 2292-2299.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: individual differences in working memory. *Psychological Review, 99*(1), 122-149.

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping*, *8*(2-3), 128-136.

Koenigs, M., Barbey, A. K., Postle, B. R., & Grafman, J. (2009). Superior parietal cortex is critical for the manipulation of information in working memory. *Journal of Neuroscience*, *29*(47), 14980-14986.

Kounatidou, P., Richter, M. & Schöner, G. A Neural Dynamic Architecture That Autonomously Builds Mental Models in (2018). *In* T.T. Rogers, M. Rau, X. Zhu, & C. W. Kalish (Eds.), *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 641-646). Austin, TX: Cognitive Science Socitey.

Krumnack, A., Bucher, L., Nejasmic, J., Nebel, B., & Knauff, M. (2011). A model for relational reasoning as verbal reasoning. *Cognitive Systems Research*, *12*(3-4), 377-392.

Marr, D. (1982). A computational investigation into the human representation and processing of visual information. *WH Freeman and Company, San Francisco, CA*.

Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, *7*(2), 153-160.

Meyer, D. E., & Kieras, D. E. (1999). Précis to a practical unified theory of cognition and action: Some lessons from EPIC computational models of human multiple-task performance. In D. Gopher & A. Koriat (Eds.), *Attention and performance XVII: Cognitive regulation of performance* (pp. 17-88). Cambridge, MA: MIT Press.

Morrison, R. G., Krawczyk, D. C., Holyoak, K. J., Hummel, J. E., Chow, T. W., Miller, B. L., & Knowlton, B. J. (2004). A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration. *Journal of Cognitive Neuroscience*, *16*(2), 260-271.

Nee, D. E., Brown, J. W., Askren, M. K., Berman, M. G., Demiralp, E., Krawitz, A., & Jonides, J. (2012). A meta-analysis of executive components of working memory. *Cerebral Cortex*, *23*(2), 264-282.

Newell, A. (1992). SOAR as a unified theory of cognition: Issues and explanations. *Behavioral and Brain Sciences*, *15*(3), 464-492.

O'Doherty, J. P., Hampton, A., & Kim, H. (2007). Model-based fMRI and its application to reward learning and decision making. *Annals of the New York Academy of sciences*, *1104*(1), 35-53.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. MIT press.

Prabhakaran, V., Smith, J. A., Desmond, J. E., Glover, G. H., & Gabrieli, J. D. (1997). Neural substrates of fluid reasoning: an fMRI study of neocortical activation during performance of the Raven's Progressive Matrices Test. *Cognitive Psychology*, *33*(1), 43-63.

Ragni, M., Franzmeier, I., Maier, S., & Knauff, M. (2016). Uncertain relational reasoning in the parietal cortex. *Brain and Cognition*, *104*, 72-81.

Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Review*, *120*(3), 561-588.

Rosenbloom, P. S. (2013). The Sigma cognitive architecture and system. *AISB Quarterly, 136*, 4-13.

Samsonovich, A. V. (2010). Toward a unified catalog of implemented cognitive architectures. *BICA*, *221*(2010), 195-244.

Schlieder, C., & Berendt, B. (1998). Mental model construction in spatial reasoning: A comparison of two computational theories. In U. Schmid, Krems, J. F., & F. Wysotzki (Eds.), *Mind modelling: A cognitive science approach to reasoning, learning and discovery* (pp. 133-162). Lengerich (Germany): Pabst Science Publishers

Schultheis, H., & Barkowsky, T. (2011). Casimir: an architecture for mental spatial knowledge processing. *Topics in Cognitive Science*, *3*(4), 778-795.

Senft, V., Stewart, T. C., Bekolay, T., Eliasmith, C., & Kröger, B. J. (2016). Reduction of dopamine in basal ganglia and its effects on syllable sequencing in speech: A computer simulation study. *Basal Ganglia*, *6*(1), 7-17.

Sun, R. (2001). *Duality of the mind: A bottom-up approach toward cognition*. Mahwah, NJ: Erlbaum.

Ullman, M. T. (2006). Is Broca's area part of a basal ganglia thalamocortical circuit?. *Cortex*, *42*(4), 480-485.

Waltz, J. A., Knowlton, B. J., Holyoak, K. J., Boone, K. B., Mishkin, F. S., de Menezes Santos, M., ... & Miller, B. L. (1999). A system for relational reasoning in human prefrontal cortex. *Psychological Science*, *10*(2), 119-125.

Wertheim, J., & Ragni, M. (2018). The neural correlates of relational reasoning: A meta-analysis of 47 functional magnetic resonance studies. *Journal of Cognitive Neuroscience*, *30*(11), 1734-1748.

Wertheim, J. & Stewart, T. (2018). Explaining Reasoning Effects: A Neural Cognitive Model of Spatial Reasoning. *In* T.T. Rogers, M. Rau, X. Zhu, & C. W. Kalish (Eds.), *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 2675-2680). Austin, TX: Cognitive Science Society.

Put Feeling into Cognitive Models: A Computational Theory of Feeling

Robert L. West (robert.west@carleton.ca),
Brendan Conway-Smith (brendan.conwaysmith@carleton.ca)
Institute of Cognitive Science, Carleton University, Ottawa, ON K1S5B6 Canada

Abstract

Feelings are potentially conscious experiences that inform us about brain/body states related to drives (e.g., feeling hungry), emotions (e.g., feeling angry) and knowledge states (e.g., feeling unsure). In this paper we propose a unified computational definition of feelings that can be used to add feelings to cognitive models.

Accounting for feelings in cognitive models is important since feelings have strong effects on human performance and decision-making. However, there is considerable disagreement over what feelings are and how, or if, they can be incorporated into cognitive models. We address this issue by providing a functional, computational definition of feelings.

Computational Theory of Mind (CTM) is an area of philosophy that argues that the brain is a form of computer. There are a variety of arguments in favour of this view, likely the most well known belong to Fodor (2000). Likewise, there are a variety of criticisms of this view, probably the most well known are Searle's (1984). Internally, CMT theorists argue about the right way to map computation to cognition. Mostly these discussions revolve around knowledge and language, but the question of how to relate feelings to computation has been broached, so this is one source of ideas about how to computationally implement feelings.

Another source of ideas is Cognitive Modeling itself. Cognitive modeling can be considered an empirical endeavour. The goal of Cognitive Modeling is to use computational modeling to represent cognitive theories and to test these theories through comparisons to data. The ultimate goal of Cognitive Modeling is to build a Unified Cognitive Architecture capable of simulating all or most human cognitive abilities (Newell, 1990). Cognitive architectures, such as ACT-R (Anderson & Lebiere, 1998) and SOAR (Laird, 2012) have been very successful in modeling knowledge driven behaviour but it is not clear how to add feelings to these architectures. However, by examining the structure of these architectures, the options for adding feelings can be elucidated.

What are Feelings?

CTM debates are focused around concepts such as symbolic representation, referents, semantics, propositions, qualia, and meaning. CTM is intended to describe the relationship between computation and the brain, but because most of the discussion is based around knowledge and language, it is unclear if these concepts are meant to apply beyond this domain (Rescorla, 2015). In particular, there seems to be an intuition that feelings are not the same as thought or language, and so must be computationally represented in a different way.

According to Damasio (2019) feelings are mental representations of non-symbolic bodily states, which are used for decision making. According to Alston (1969) feelings are, "spontaneously-emerging occurrent phenomenal experiences," which he refers to as "datable states of consciousness." However, Arango-Muñoz & Michaelian (2014) indicate that feelings do not involve "properly propositional content."

Overall, there seems to be agreement that a feeling is a unitary phenomena that we have potential conscious awareness of. Feelings can factor into decision making but there is an intuition that feelings are somehow different from propositional, symbolic knowledge. Finally, feelings are derived from more complex, distributed phenomena, such as emotions and drives.

Noetic Feelings

In addition to drives and emotions, feelings can also be derived from states of knowing or learning. These feelings have been referred to by terms such as, feelings of knowing or FOK (Hart, 1965), metamemory (Flavell, 1971), knowledge judgements (Schneider, 2000), cognitive emotions (Standish, 1992), and epistemic feelings (Arango-Muñoz & Michaelian, 2014). In the following paper we will subsume this lexicon under the term "noetic feelings." This follows Metcalfe's (2013) identification of "noetic" to mean cognitive phenomena in which the referent concerns an internal state or internal representation.

Research indicates that noetic feelings drive memory search as subjects take more time to search their memory if they "feel" they know it (Barnes et al., 1999). Studies also show that noetic feelings are a reliable signal of the likelihood of memory retrieval (Hart, 1965), and feelings of probable retrieval success or retrieval failure affect the strategy used to engage the problem (Conway, 2009; Singer & Tiede, 2008). Noetic feelings have also been reliably correlated with improved learning outcomes (Wang, Haertel, & Walberg, 1990). Subjects will also spend more time learning words previously considered to be difficult to remember (Nelson & Leonesio, 1988). Moreover, the "feeling of rightness" has been studied in the rapid solving of complex, real-world problems (Thompson et al., 2011).

Models associated with noetic phenomena include Reder's (1996) use of the source of activation confusion (SAC) model, Dougherty's (2001) multiple-trace memory model, Metcalfe's (1993) holographic associative model, and Sikström and Jönsson's (2005) stochastic dri model of memory strength to explain delayed judgement of learning.

Thus research indicates that noetic feelings are a guidance system integral to directing cognitive processes. Progress toward accurately describing human cognition requires integrating noetic feelings into cognitive modeling.

Reasoning from Architectures

There is a tendency in CTM papers to focus on foundational issues. In the case of emotions, for example, this manifests as a concern over establishing what emotions are before considering how they can be computationally represented. For example, emotions are defined variously as bodily states (Damasio, 2019), perception (Prinz, 2006), and natural kinds (Barrett, 2006). However, since there is no agreement on the status of emotions we have no foundational basis to reason about the nature of the feelings that are derived from emotions.

In contrast to this foundational approach, we ground our work on the function of feelings within cognitive models. That is, we take a top down functional approach as opposed to a bottom up foundational approach. Ideally, these two different approaches can inform each other, but it is important not to confound the two.

We take the ability of cognitive models to account for data as evidence that the model embodies something true about the computational functionality of the brain. One criticism that can be levelled at this approach is that there are many different cognitive models. However, our focus is not on the differences between the models, but rather on their similarities. We argue that there are significant convergences in cognitive modeling at the level of the architecture. More generally, we interpret unified cognitive architectures as a way of grounding theory in functional coherence, without engaging with foundational issues. In particular, we focus on the Common Model Architecture.

Common Model of Cognition

The Common Model of Cognition (formerly known as the Standard Model of Cognition) is a conceptual architecture put forward by Laird et al. (2017). The concept of a common model is based on Laird et al.'s claim that there has been significant convergence across cognitive architectures over time, to where we are now at the point that we can talk about a common cognitive architecture. The common model describes a conceptual architecture that is common to most, if not all, cognitive architectures capable of modelling complex human behaviour.

The basic structure of the common model is shown in Figure 1. The common model describes a production system (corresponding to procedural memory) that interacts with different modules through a buffer system that corresponds to working memory. The architecture is parallel and asynchronous, with the production system acting as a control system. There are significant divergences in terms of how components are implemented in different common model-type architectures (e.g., spiking neurons, neural networks, high dimensional vectors, semantic networks, Bayesian networks, graph theory, etc.). However, the common model describes the common functionality across different implementations.

The common model is not meant to describe all of human cognition, it is a model of cognitive control and decision making. As Newell (1990) noted, this is one possible starting point for understanding cognition. In contrast, CTM appears to have knowledge and language as its starting point. In other words, CTM is based on *knowing* while the common model is based on *doing*. Bridging the two is conceptually tricky, not least because they use the same terms in different ways. In this paper we will attempt to merge CTM work on feelings with the common model. Specifically, we argue that feelings are best modelled as non propositional representations in buffers (related to this see West & Young, 2017, for a discussion of representing amygdala states in the buffers).

Qualia

Qualia refers to the qualitative differences between our conscious experience of thoughts, senses, emotions, and drives. Explaining how different patterns of neural activity can produce these qualitatively different experiences is part of what Chalmers (1996) referred to as the hard problem of consciousness (Chalmers, 1996). There continues to be much debate on the subject and

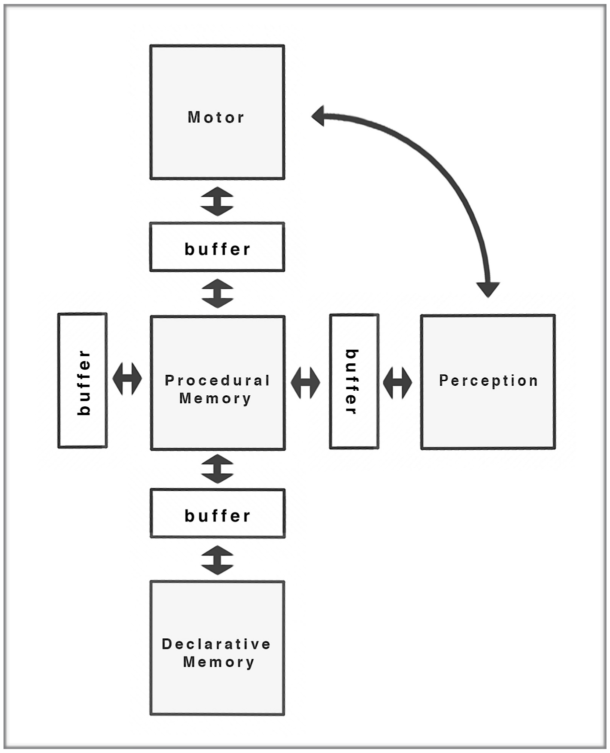


Figure 1. The Common Model of Cognition

the matter is far from settled. Given that there is no agreed upon definition of consciousness we will not make any strong claims about what parts of the common model are conscious. However, because the architecture can report the buffer contents it is clear that the buffer contents are potentially conscious, and are definitely conscious when reported. For example, if a model reports remembering a brown cat this would correspond to a conscious awareness of this cat in memory. Likewise, feelings can be consciously experienced but are not necessarily always consciously experienced (Redder & Schunn, 1996; Metcalfe & Son, 2012; Son & Kornell, 2005).

Modules and mechanisms

Fodor (1983) has greatly influenced how to think about the brain in terms of modules. However, cognitive modellers almost universally ignore Fodor's foundational requirements for modularity. So we will not follow Fodor on this. Instead we take modules in cognitive models to represent mechanisms in the brain (Betchel, 1994).

The mapping between a module in a model and a module in the brain can occur in different ways. It could be one to one, where a module in the architecture maps directly to an area of the brain, as suggested by J. Anderson (e.g.; Anderson et al., 2004). In terms of emotion, this approach is represented by theories of basic emotions, which postulate distinct neural modules for processing specific emotions. For example, (Panksepp, 1998) postulates areas for basic emotions based on comparisons of mammalian brains.

However, a computational module could also map to a network of multipurpose modules assembled to generate a higher level function, as suggested by M. Anderson (2010). In terms of emotion, this approach is represented by theories of complex emotions and emotional networks. Such networks could produce complex sets of feelings or blended feelings.

Brain wide states, such as neurotransmitter levels or hormones can also be modeled. For example, Ritter et al. (2006) describe a system for the ACT-R architecture that specifies the effects of hormones and neurotransmitter levels on modules in the architecture. Likewise, Core Affect theory (Russell et al, 1999) models brain wide chemical states in terms of a two dimensional valence/arousal model. We assume that brain wide states contribute to feelings through their impact on modules.

Propositions, Symbols and Feelings

Cognitive models either use symbolic propositional knowledge or, in the case of neural networks and spiking neuron models, they act as if they do. This makes sense for modelling knowledge driven processing but it raises a concern because consciously experienced feelings seem to possess qualia without associated propositional content. This can evoke complex philosophical questions (as represented by the thought experiment of Mary the colour blind scientist). However, we hope to avoid questions related to qualia by focusing on function. We begin by considering if the role of buffer representations is necessarily symbolic or propositional.

The buffer contents could be considered as representing propositional knowledge in the sense that the buffers are considered to contain true information. The buffer contents could also be considered to be symbolic representations in that they can correspond to things in the real world. However, what the buffers actually contain is the outputs of a module. How this relates to the state of the world is dependent on the relationship between the module and the world. If we consider the immediate function of the buffer contents for choice or decision making, they do not refer to anything except the matching code in the *if* condition of a production rule.

Whether or not the buffer contents should be considered propositional or symbolic is hard to answer because there is very little agreement on how to define these terms. Many people (but not all) would agree that the linguistic representation of the statement "there is a black cat" is both propositional and symbolic. However, if we change it to a visual representation of a black cat then some would argue that it is no longer propositional or symbolic, while others would maintain that this changes nothing.

The key is to distinguish between the status of a representation conferred by being in a buffer versus the status of a representation conferred by its syntactic or representational structure. We argue that being in a buffer does not directly imply that a representation is symbolic or propositional as the only essential requirement is that the code in a buffer can match the code in the *if* part of a production. What could potentially distinguish a buffer containing a feeling from a buffer containing knowledge is the computational structure of the representation itself.

Here it is important to note that buffer contents in the brain are represented by neural firing patterns. However, these neural patterns can be represented by a symbol in a model without implying that the pattern has a symbolic function in the brain. For example, if the average spiking rate of a group of neurons was expressed as 42, although 42 is a symbol, it is without meaning unless you know the question that it answers. Even the numerical value of 42 is meaningless without knowing the units of measurement. Using symbols in a model may be merely a convenience for the modeller, it does not necessarily imply a theoretical commitment. For example, if the feeling of anger was represented by putting the word “anger” in a buffer, this would not imply anything.

Following this we can ask — what would it mean if a buffer contained a word (or neural pattern) corresponding to a qualia, such as anger, or tired, or unsure? Functionally, because feelings can be experienced consciously, we know from experience that we use them to make decisions. Whether or not they are propositional or symbolic seems to depend on the extent of the conceptual framing of the decision. If it is simply the moment of matching to a production then it can be argued that they are neither propositional nor symbolic. If the question is “why did you hit that man?” then the function of the feelings involved could be argued to be propositional and symbolic, in terms of their role in larger decision.

Mentalese

Mentalese is a concept invented by Fodor (2000) to distinguish between language and the language of thought. However, we use the term in the broader sense outlined by Pinker (1997), in which different modalities have their own mentalese. For example, Pinker proposed that we have visual mentalese. We interpret the contents of the buffers to be mentalese and propose there are different types of mentalese. The implication of this is that the mentalese used in one buffer may not be directly translatable to the mentalese used in another.

This is an important issue for the common model. If two buffers use the same mentalese, then a single production can transfer information directly from one buffer to another without reference to the content, but if they use different mentalese there needs to be a translation. Minimally, this would require a different production for each object of translation. For example, if a representation of a stop sign is in the visual buffer, to put a representation of “stop” in the goal buffer requires a production recognizing the visual mentalese representation of the stop sign on the *if* side and, on the *then* side, puts a goal mentalese representation of “stop” in the goal buffer. Alternatively, it is possible that there is a common mentalese for knowledge and the different modalities translate information into this common language before it arrives in the buffer. Most common model models are programmed as if the second option is true. Ideally, it will be possible to empirically answer this question.

However, our common experience with feelings indicates that, although we can label them, we often have difficulty putting them into words. The entire field of poetry is arguably dedicated to this effort. Another distinguishing factor is that we cannot alter our feelings in the same way we can alter our knowledge or goals. For example, if I have stopped at a stop sign and there is no traffic, I can quickly alter the content of my goal buffer from “stop” to “go.” In contrast, if I am angry and I realize that it is unwise to be angry, I cannot simply change the feeling in the buffer to another emotion. These examples suggest that feelings have their own mentalese and that the production system cannot directly alter this mentalese. Combined with the fact that some people have difficulty labeling their feelings, this suggests that the production system learns, through experience, to associate knowledge-mentalese labels with different feeling-mentalese representations. This suggests that, feeling-mentalese functions more like a sign system, similar to what animals are capable of.

Feelings are also associated with phenomena such as facial expressions and hormonal release. However, at the 50 millisecond time scale of productions we are talking about choice. For example, in an approach avoidance scenario such as a monkey contemplating food left out in a clearing, we simultaneously experience the feelings of hunger for food and fear of predators in the clearing. The result is a vacillating, back and forth behavior accounted for by opposing productions firing back and forth.

Related to this, mindfulness training in Cognitive Therapy can be understood as learning to translate feelings to knowledge in order to use the more advanced properties of knowledge to gain a better purchase on our behavior. Once a feeling is labeled it has been translated to knowledge, but this new representation is not a feeling, and our experience tells us that the feeling still independently exists. For example, if you are walking home in the dark after a watching a vampire movie, you might experience fear. By translating the fear-feeling to knowledge, you can reason that vampires are not real and so you are not in danger. However, while this will help, and may decrease the fear feeling, the fear feeling will independently persist in the short term.

Feelings as Metadata

We propose that feelings are metadata and that feeling-mentalese is a language appropriate for expressing metadata, whereas knowledge-mentalese is a language appropriate for expressing knowledge. This makes sense since we know that, computationally, metadata expressions are typically different from knowledge expressions. For example, metadata is often best expressed through statistics and high dimensional spaces, whereas knowledge is often best expressed through propositional statements and logical operators. This also accounts for the fuzzy, non-verbal qualia of feelings.

To maintain the distinction between knowledge and metadata, we argue that knowledge statements about feelings, such as, I feel angry, or, I feel confused, are translations performed by productions that recognize metadata states and create knowledge based statements about them. So, as such, these statements are knowledge and not feelings. Questions about whether feeling-metadata can be considered propositional or symbolic, we believe, depends on how the data is used in the model.

Another computational distinction we think we can make is that feelings are bottom-up, read-only statements. That is, feeling-representations are placed in buffers by their associated modules and the central production system cannot alter them. Only the module that created them can alter them. The production system may or may not have direct access to the module. In contrast, knowledge representations in the buffers can be altered directly by the production system, as is common in common model architectures.

Conclusion

We have presented a computational theory of feelings based on the common model architecture. More broadly, we have shown how cognitive architectures can be applied to clarify philosophical issues, particularly in CTM. We believe this type of work is important as conceptual confusion over issues, such as the difference between knowledge and feelings, can conceptually impede the creation or acceptance of cognitive models involving these phenomena. Finally, by stating our ideas in terms of a cognitive architecture we have made them computationally unambiguous. Other, different models are possible, but they should be stated clearly, in computational terms, and grounded in a viable cognitive architecture.

References

Arango-Muñoz, S., Michaelian, K. (2014). Epistemic Feelings and Epistemic Emotions (Focus Section). Philosophical Inquiries.

Anderson, J. & Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J., Qin, Y., Stenger, V., & Carter, C. (2004). The relationship of three cortical regions to an information-processing model. Journal of Cognitive Neuroscience, 16 (4), 637-653.

Anderson, M. L. (2010). Neural reuse: A fundamental organizational principle of the brain. Behavioral and Brain Sciences, 33(4), 245–266.

Alston, William P. Feelings. The Philosophical Review 78(1): 3-34, 1969.

Barnes, A., Nelson, T., Dunlosky, J., Mazzoni, G., & Narens, L. (1999). An integrative system of metamemory components involved in retrieval. Attention And Performance Xvii–Cognitive Regulation of Performance: Interaction of Theory And Application, 17, 287–313.

Barrett, L. F. (2006). Are Emotions Natural Kinds? Perspectives on Psychological Science, 1(1), 28–58.

Bechtel, W. (1994). Levels of Description and Explanation in Cognitive Science. Minds and Machines, 4, 1-25.

Chalmers, David J. (1996). The Conscious Mind: In Search of a Fundamental Theory. Oxford University Press.

Computational Theory of Mind. (2015). Stanford encyclopedia of philosophy. Retrieved from https://plato.stanford.edu

Conway M.A. (2009). Episodic memories. Neuro psychologia. 47: 2305-13. PMID

Damasio, A. (2015). Scientific American 2019. Oct. 16 Nation, 294(20), 11-18. Retrieved from MAS Ultra.

de Sousa, R. Epistemic feelings. In: Georg Brun, Ulvi Doˇguoˇglu, and Dominque Kuenzle, eds., Epistemology and emotions. Ashgate, 2008.

Dougherty, M. R. P. (2001). Integration of the ecological and error models of overconfidence using a multiple-trace memory model. Journal of Experimental Psychology: General, 130, 579–599.

Flavell, J.H. (1971). What is memory development the development of? Human Development, 14, 272-278.

Fodor, J. (1983). Modularity of Mind: An Essay on Faculty Psychology. Cambridge, Massachusetts: MIT Press.

Fodor, J. (2000). The Mind Doesn't Work That Way:The Scope and Limits of Computational Psychology.

Hart, J. T. (1965). Memory and the feeling-of-knowing experience. Journal of Educational Psychology.

Laird, J. & Lebiere, C. & Rosenbloom, P. (2017). A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. AI Magazine. 38.

Laird, J. E. (2012). The SOAR cognitive architecture. MIT Press.

Metcalfe, J. (1993). Novelty monitoring, metacognition, and control in a composite holographic associative recall model: Implications for Korsako amnesia. Psychological Review, 100, 3–22.

Metcalfe, J., & Son, L. (2013). Anoetic, noetic, and autonoetic metacognition. Foundations of Metacognition.

Metcalfe, J., & Son, L. K. (2012). Anoetic, noetic, and autonoetic metacognition. In M. Beran, J. L. Brandl, J. Perner, and J. Proust (Eds.). Foundations of Metacognition (pp. 289-301). Oxford University Press

Metcalfe, Janet & Son, Lisa. (2013). Anoetic, noetic, and autonoetic metacognition. Foundations of Metacognition

Nelson, T. O., & Narens, L. (1990). Metamemory: a theoretical framework and new findings. In G. H. Bower (Ed.), e psychology of learning and motivation (Vol. 26, pp. 125–173). New York: Academic Press.

Nelson, T. O., & Narens, L. (1994). Why investigate metacognition? In J. Metcalfe, & A. J. Shimamura (Eds.), Metacognition: Knowing about knowing (pp. 1–26). Cambridge, MA: MIT Press.

Newell, A. (1990). Unified theories of cognition. Cambridge: Harvard University Press.

Panksepp, J. (1998). Affective neuroscience: The foundations of human and animal emotions. New York: Oxford University Press.

Pinker, S. (1997). How the Mind Works. New York: Norton.

Prinz, J. (2006). Canadian Journal of Philosophy, Volume 36, Supplement Col. 32, pp. 137-160.

Proust, Joelle (2009). Is there a sense of agency for thought? In Lucy O'Brien & Matthew Soteriou (eds.), Mental Actions. Oxford University Press.

Reder, L. , & Ritter, F. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. journal of Experimental Psychology: Learning, Memory, and Cognition, 18, 435-451.

Reder, L. M., & Schunn, C. D. (1996). Metacognition does not imply awareness: Strategy choice is governed by implicit learning and memory. In L. M. Reder (Ed.), Implicit memory and metacognition (pp. 79–122).

Ritter, F. E., Reifers, A. L., Klein, A. C., & Schoelles, M. J. 2006. Lessons from Defining Theories of Stress. In W. Gray (Ed.) Inte- grated Models of Cognitive Systems. New York, NY: Oxford University Press.

Russell, James A.; Barrett, Lisa Feldman (1999). Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant" (PDF). Journal of Personality and Social Psychology. 76 (5): 805–819.

Searle, J. (1984), Minds, Brains and Science: The 1984 Reith Lectures, Harvard University Press.

Schneider, W., Visé, M., Lockl, K., Nelson, T. (2000). Developmental trends in children's memory monitoring - Evidence from a judgment-of-learning task. Cognitive Development. 15. 115-134.

Singer, M., & Tiede, H. L. (2008). Feeling of knowing and duration of unsuccessful memory search. Memory and Cognition, 36, 588-597.

Son, L. K., & Kornell, N. (2005). Meta-confidence judgments in rhesus macaques: Explicit versus implicit mechanisms. In Terrace, H.S. & Metcalfe, J. (Eds.), The Missing Link in Cognition: Origins of Self-Knowing Consciousness. Oxford University Press.

Standish, P. (1992), In Praise of the Cognitive Emotions. Journal of Philosophy of Education, 26: 117-119. opmental inquiry. American Psychologist, 34, 906–911.

Strawson, P. F. (1948). Truth. Analysis 9 (6): 83-97.

Sikström, S., & Jönsson, F. (2005). A model for stochastic drift in memory strength to account for judgments of learning. Psychological Review, 112, 932–950.

Thompson VA, Prowse-Turner J, Pennycook G. (2011) Intuition, reason & metacognition. Cogn Psychol 63:107–140

Wang, M., Haertel, G., & Walberg, H. (1990). What Influences Learning? A Content Analysis of Review Literature. The Journal of Ed. Research, 84(1), 30-43.

West, R. L., & Young, J. (2017). Proposal to add emotion to the standard model. 2017 AAAI Fall Symposium Technical Report Volume 17, Symposium 6: A Standard Model of the Mind

SEEV-VM: ACT-R Visual Module based on SEEV theory

Sebastian Wiese (sebastian.wiese@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23
Technische Universität Berlin, D-10587 Berlin

Alexander Lotz (rene_alexander.lotz@daimler.com)
Daimler AG, Truck Product Engineering (TP/VES)
D-70546 Stuttgart

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23
Technische Universität Berlin, D-10587 Berlin

Abstract

In this publication an adaptation of the ACT-R visual module is presented based on the SEEV theory on attention allocation. By including this theory into the methodology of how the visual module works, a top-down control of attention guidance and bottom-up processing capabilities were implemented. The visual field of the model shifts according to current fixations, mimicking human behavior. Finally, we introduce a possibility of linking this new visual module with environmental sensors of a vehicle to generate data for the model without the need of a modeler generating environmental data. As of now the interpretation of the environment could be visualized differently depending on the understanding of the modeler. Now, the modeler benefits by having a time efficient reproducible source for data generation for driver modeling.

Keywords: ACT-R; Visual Module; SEEV theory; External Sensory Data; Driver Modeling

Introduction

Cognitive architectures, which are based on theoretical constructs with the objective to model real-world thoughts and interactions, offer a possibility to abstract human cognition. While these architectures deploy a method to test applied tasks, these tasks are also required to validate the implemented theory (Russwinkel, et al., 2018). At the same time, with abstraction, there is also a loss of information when forcing data into the required format of these cognitive architectures to create cognitive models. ACT-R (Anderson, et al., 2004), as an established cognitive architecture, offers the abovementioned complexity of separated modular modalities. Especially the visual module, as a main modality delivering information to most cognitive models and interpreting the environment, limits the applied tasks significantly. Currently, visual information is presented in a GUI (Graphical User Interface) and needs configuration. As the modeler dictates the position and characteristics of objects in the GUI, subjective interpretation of these characteristics can make the environment of a model inherently differently. This can affect the outcome of the model.

A task that has been addressed manifold in research is the modelling of the driving task and the ambition to model driver behavior (Salvucci, 2005; Salvucci & Taatgen, 2008). This task mainly consists of knowledge and experience that is applied with motoric outputs based on visual and auditory information. Therefore, multiple ACT-R modules are required to interact during modeling. Especially since (conditional) automated driving, Level 3, is prospected in near future, effects of attention and distraction (Haring, 2012), of driver drowsiness (Gunzelmann, et al., 2011), multi-tasking (Kosanke & Russwinkel, 2016) as well as insights on non-driving related tasks (Salvucci, 2009) have been modeled. Ultimately, combination of a human driver model in ACT-R with the possibility of a direct connection to a vehicle, to possibly adjust vehicle behavior according to the needs of the driver, is a promising vision. This proposal is similar to the ACT-Droid approach (Doerr, et al., 2016), in which a direct connection of a computational model in ACT-R to a system is configured. Also, the interpretation of simulation data through computer vision, ACT-CV (Halbrügge, 2013), has been presented as a means to develop visual information for ACT-R. While ACT - Droid connects to a self-contained system or interface, our new approach utilizes the vehicle as a means to monitor the real-world environment similarly to ACT-CV. Secondly, through the connection with the vehicle it is possible to model driver behavior and direct results to driver assistance systems, to increase system acceptance and possibly assist drivers in difficult situations.

Building rich environments with the default ACT-R device system is difficult. Standard ACT-R provides too few visual object types, making it near impossible to build real world scenes without defining a notation for object connotation like mapping colors to semantic meaning. This makes models hard to understand and extend. The difficulty of designing interfaces within the ACT-R toolchain can be bypassed by using external tools to generate the world around the agent or the interface, such as with abovementioned ACT-Droid. Thus far, the implementation of vehicle environments is tedious. Additionally, the environmental configuration underlies personal interpretation of the modeler and one task can be programmed in multiple ways, yielding the possibility of different calculated results. This drawback is addressed in the present concept by introducing a new adaptation of the

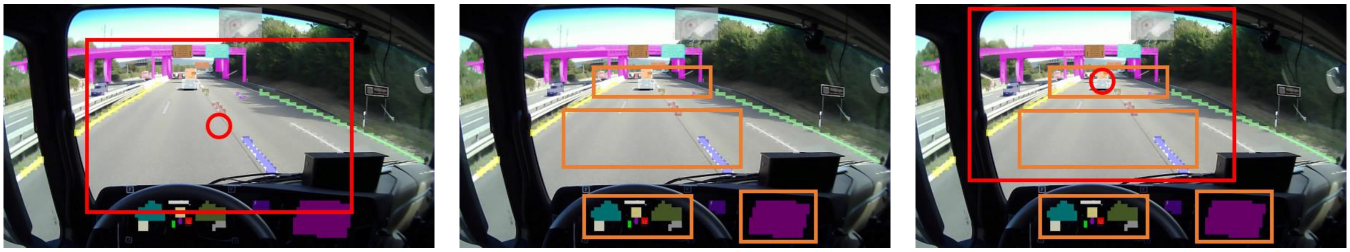


Figure 1: For the module to work, it needs some input data. In this case an annotated camera image. The field of view (red) does not span the entire image. The agent only perceives the color-coded parts, i.e. road surface is invisible to it. The first step is to setup a world simulation and stream data into the ACT-R runtime. The cognitive model manages AoRs (orange). Together they move attention to a car in front (white).

ACT- R visual module. Additionally, a framework for the connection of ACT-R to vehicle data is presented, in which the data of external sensors are interpreted and scaled to allow to feed the visual module with information, evading the necessity to model the environment. A detailed description of this process is presented in this publication.

A new vision module – SEEV-VM

Perceiving real world scenes is a hard task for a cognitive model. It requires the model to comprehend the scene, extract meaning and make assumptions about location and type of information. There is a lot of uncertainty involved, where to precisely find requested information or whether it is present at all. Henderson (2003) identified three different kinds of knowledge that are involved in a gaze guiding mechanism. That are: episodic scene knowledge, remembering where objects were seen lastly or, on a long term, where to expect task-relevant information, but about a specific scene. Scene-schema knowledge provides generalized semantic and context information, e.g. we know how car interfaces look like and can easily orientate oneself in a yet unknown car cockpit. The third is task-related knowledge. This type of knowledge includes learned fixation sequences, e.g. monitoring traffic before and while changing lanes with a car. They have in common that they encode a location with a meaning. This idea, also present in the works of (Oliva, et al., 2003), constitutes the foundation of our proposed visual module: SEEV-VM.

The SEEV approach (Wickens, 2015) can predict a scan path in rich visual environments like airplane cockpits. The visual workspace consists of displays, also called areas of interest that attract attention and contain task relevant information. Every display is defined by four numeric factors: salience, effort, expectancy and value. The SEEV algorithm decides which display will be attended by summing up factors for all displays and comparing the results. This approach combines bottom-up and top-down factors. Wickens (2015) describes **salience** as the physical properties of a display that increase its attraction for the human eye, e.g. high contrasts or bright colors. **Effort** correlates to the distance between the target display and the current point of fixation. **Expectancy** and **value** form the top-down factors: value describes the relevance of information in a display and expectancy the frequency with which information updates. I.e. a high frequency and a high value display will be attended more often, because its information is important and changes frequently, therefore needs to be sampled often.

The proposed visual module (SEEV-VM) is based on the ideas of the SEEV theory and existing vision modules like EMMA (Salvucci, 2000) and PAAV (Nyamsuren & Taatgen, 2013). EMMA extends ACT-R with realistic eye movements by integrating physiological constraints of the human eye. PAAV extends the attention guidance mechanism itself. It integrates bottom-up factors into the existing top-down control of attention.

The SEEV theory provides not only an algorithm for guiding visual attention, but also a representation for top-down control of the attention guidance mechanism. Information is expected to be found in certain places in the environment. In the SEEV-VM these locations are called areas of relevance (AoR) to differentiate from AoI in eye tracking experimental set ups. Both PAAV and SEEV integrate top-down and bottom-up processing into their algorithms, both use numeric values and calculate an attraction value (SEEV) or an activation value (PAAV). It is a reoccurring idea to fuse all factors into a single parameter to base an attention selection decision on. The SEEV-VM module uses a very similar approach and calculates a guidance value for each visual object and AoRs.

The algorithm selects an object to attend based on the guidance values, see Figure 1, then shifts attention towards this object and starts encoding. After encoding, the algorithm immediately repeats the process, searching for an object to fixate. This mechanism runs in an endless loop without directing instructions through the buffer interface. It is assumed that the human eyes always look at something and provide information about the visually perceivable environment. Only when the production system accesses this information, which is using the vision module buffer content, attention is directed at the given object.

Arbitrary visual objects

SEEV-VM supports two different modes of operation: In the **traditional mode of operation**, the SEEV-VM module manages visual objects with function calls. Functions can create, modify (also their semantic meaning, e.g. a traffic

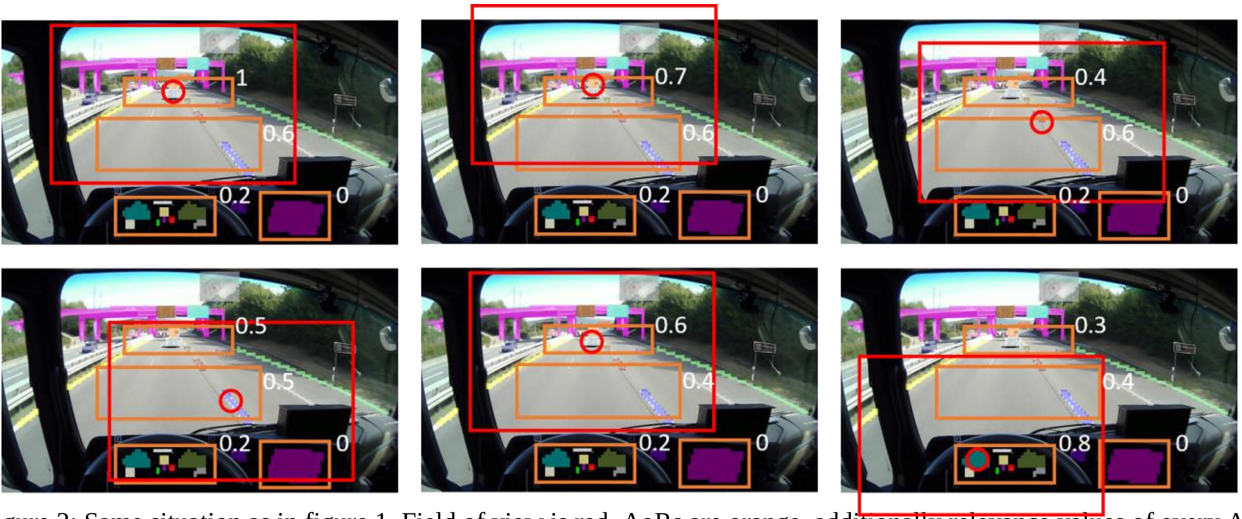


Figure 2: Same situation as in figure 1. Field of view is red, AoRs are orange, additionally relevance values of every AoR are shown next to the AoR. Numbers show the relevance before attention is shifted; red dot shows the result of the attention shift. In the example, attention is directed at the white car in front, reducing relevance value of its AoR to 0.7. After the white car was encoded, the attention guidance mechanism starts again. Inhibition of return prevents the module from looking at the same object again, hence the orange car is attended. On the last part, a production fired that increases relevance of the instruments AoR. Based on salience and effort influences, attention is directed at the speed indicator instead of other instruments.

light can turn green) and remove objects (removing an object makes it invisible to the vision module). The attention guidance algorithm works the same in both modes. The difference is, that in this mode the algorithm iterates over all objects, checking whether an object is inside the current field of view and calculating attraction values. This mode does not support occlusion of objects, if an object is added, it should be visible. For very complex 3D environments, another mode was implemented. In the **pixel-based mode** the module receives a semantically annotated map of the world that matches the current field of view of the agent. The external data source produces images (like in a video game), but instead of pixels with color values, every pixel contains a numeric object identifier. Sensors, such as vehicular external sensors, with internal object identification algorithms can provide this information.

As of the time of implementation, the ACT-R architecture did not provide a standardized and easy way to integrate external data into the simulation runtime. Hence, the SEEV-VM module provides its own communication protocol. The world simulation is linked bidirectional with the vision module, as the SEEV-VM module interacts directly with the world by moving the field of view.

The SEEV-VM module allows a modeler to define visual objects that are not bound to a limited number of categories such as geometric forms, text or buttons. A visual object can be everything, ranging from a smartphone display or other complex objects to its content like icons or lines. The result of the encoding process is a chunk that is placed into the vision modules buffer. This chunk holds characteristics that the modeler can define and physical properties of the object (i.e. location and dimension). The chunk can contain information like distance to the object or other information that is expected to be processed or calculated by the vision system. Figure 1 displays an exemplary scenario of a head-mounted camera with an automatic object recognition that can be passed to the SEEV-VM module. The module can process every color-coded object. The red dot is the center of fixation, the red box the field of view. Orange boxes show areas of relevance. In the example, the vision system manages AoRs to monitor the traffic in front of the car, the instruments and a display for a non-driving related task.

The encoding time is the amount of time the vision system fixates an object until it can place its chunk into the buffer. A modeler can also choose to set this value. This allows the module to adapt to the scope of the simulation. The environment can be made up of several displays that take longer to encode but also provide more information; similar to SEEV approach it takes one long fixation to sample a display. Or, in a more detailed simulation, the content of each display is modelled, these items take less time to encode but only provide their information (their semantic chunks). That means to sample all information of a display, every object of this display needs to be attended.

In order to enable the module to function properly, the modeler must define salience values of objects. Unlike the PAAV module salience is not calculated by the module, because not every object has features like color or shape. It's optional to specify these features. The vision system can be instructed to look for certain features, but there is no guarantee that only objects that match are attended. This works very similar to the PAAV module: feature selection is one factor of many that form the guidance value.

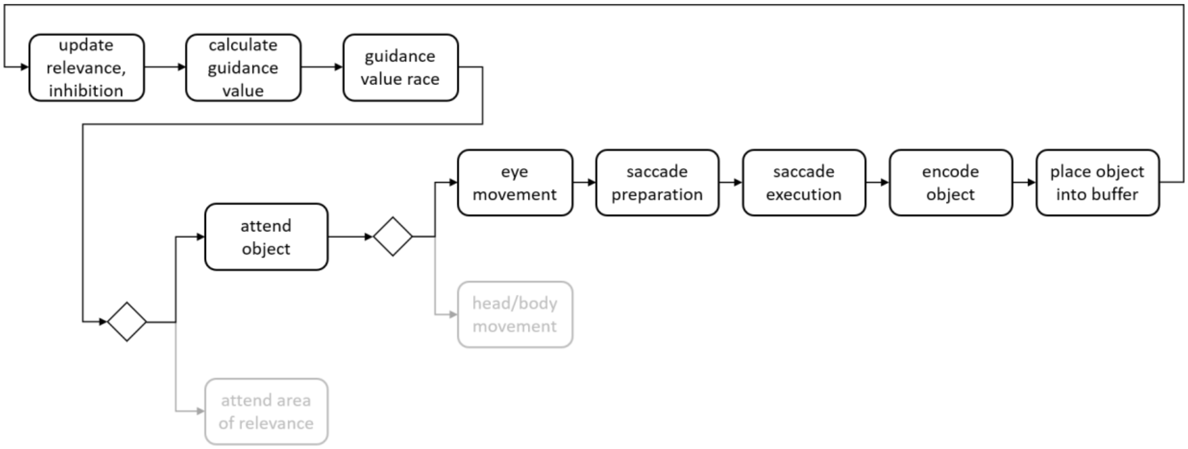


Figure 3: Process diagram of the attention guidance algorithm. The algorithm starts with a recalculation of relevance and inhibition. Inhibition decays over time and relevance increases over time. In the example, an object won the guidance value race, that is a comparison of all guidance values, and an eye movement is necessary to reach its location.

Using SEEV-VM

The SEEV-VM requires an external world simulation, the device interface is no longer used for vision. This world simulation can use many ways to produce semantically annotated maps or lists of objects, that are then communicated with the ACT-R environment and translated to function calls (adding, modifying or removing objects). E.g., the world simulation can use computer vision algorithms to annotate camera images or be a virtual world entirely, similar to ACT-CV (Halbrügge, 2013).

The communication protocol guarantees that both cognitive model and world simulation are synchronized. Hence, besides object extraction, the simulation must allow to stop and advance simulation time.

On the cognitive model side, the visual organization buffer allows to instruct the visual system by providing chunks to create, modify or remove areas of relevance and to set a feature search vector. AoRs have a location and a dimension, they form a rectangular space (orange AoRs in Figure 2), have a relevance value and two additional values that correspond closely to the expectancy value of the SEEV theory. It is possible that an AoR encompasses multiple objects. To sample all information inside this AoR all objects need to be attended. Unlike the original SEEV approach, relevance (in the SEEV approach called value) here changes over time: once an object is attended, relevance of its AoR is reduced (or consumed) for a certain amount of time (based on its refresh rate). The consumption value relates to the number of expected objects, the refresh value to the frequency with which changes are expected. These values are optional. By setting these values to zero the module will not update relevance values of AoRs, but relevance values can be updated via the production system. The SEEV approach is an abstraction of the whole cognitive process, in ACT-R this process is subdivided into smaller, parallel executable processes. Therefore, it is possible to update relevance of an AoR once all information is sampled. This is done by defining productions that count the number of objects in a given AoR. After a certain number of objects attended, another production reduces this AoRs relevance via the visual organization buffer. Later a production fires that increases relevance value again. This approach is more akin to the ACT-R way of modelling cognition. And requires a very detailed modelling of involved processes.

The attention guiding functionality works in three steps: (1) a guidance value is calculated for every visual object and AoR by adding up salience, relevance, feature weights, inhibition of return and effort. (2) A guidance value race determines the object with the highest guidance value. This allows the agent to look at areas that are not currently in the field of view, e.g. to look at the passenger’s door mirror (see Figure 2). (3) An attention shift is then initiated, it follows the EMMA model in three stages: (1) preparation of a saccade, (2) execution of the saccade and (3) encoding of the object. Figure 3 shows the workflow of the module.

Attending an AoR forms a special case, which allows the agent to look at an area that is not currently visible. Because there is no object to encode, the module immediately starts to search for objects to attend. The algorithm can initiate a head movement, as the default motor module of the ACT-R system cannot move the agents head, the vision module simulates head movements. A shift of the field of view (red bounding box in Figure 2) simulates this movement. In some cases, a movement of the whole body is needed to look at certain locations; in these cases, the module assumes that it can control the body entirely. This allows the model to visually perceive rich 3D environments regardless of these missing functionalities. SEEV-VM uses parameters to control when to make a head or body movement and how fast these movements are executed.

Vehicular data generation

As described previously, complex 3D environments are difficult to model and the proposed second mode SEEV-VM can receive semantically annotated maps. Modern vehicles are equipped with multiple internal and external sensors to allow advanced driver assistance systems to function. This information is available within a vehicle on CAN-Bus (ISO 11898-1:2015), Ethernet or FlexRay (ISO 17458-1:2013, 2013) networks for microcontroller communication and holds semantic information about surrounding objects. Depending on the sensor and data definition, multiple value signals are calculated and retrievable within these networks. Sensor types that can observe and classify objects in the proximity of the vehicle are radar, lidars and cameras. These sensors function as the 'eyes' of modern vehicles to provide environmental data for assistance systems (e.g. Adaptive Cruise Control, Lane Keep Assist and Emergency Brake Assistance).

Typically, these sensors are capable of identifying several vehicles, objects or pedestrians during driving, similarly to the way ACT-R models these objects in its environment. Apart from the classification, precise speed, distance and trajectories are calculated as properties of the objects. This information is communicated within the networks and updated rapidly (approximately 0.01-0.1 seconds). In order to model driver behavior, it would be ideal to make this data available to ACT-R. This has three major benefits: (1) modelling of environmental objects would be automated and several different scenarios could be analyzed through the proposed second mode of SEEV-VM operation. (2) Obtained data would only vary depending on sensor setups and are reproducible (attributes are not defined by modelers). (3) A framework could work with offline data after drives or online with a model predicting driver behavior.

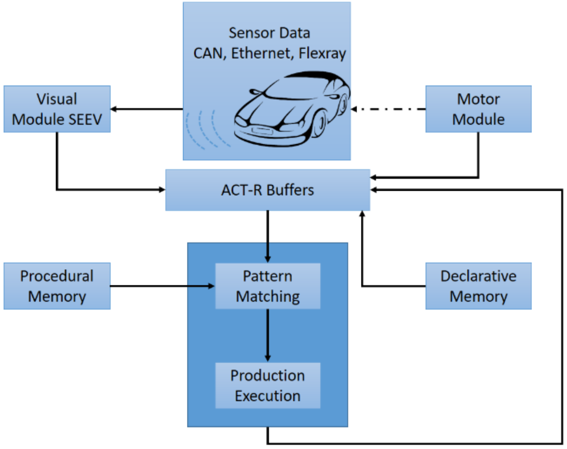


Figure 4: Adapted ACT-R framework with direct link to vehicle and modified visual module.

The adaptations of the proposed data generation enable interpretation of environmental data through sensors of a vehicle, airplane or robot, as presented in Figure 4. In this form, the bus data is interpreted and is available as CSV-files (Comma Separated Values). These CSV-files include the relevant data in lists with timestamps and the values of sensor data (e.g. distance to object, object type, speed of object). Data is interpreted through a parser to translate the data into the three function types included in the SEEV-VM (i.e. add-visual-object, modify-visual-object and remove-visual-object). The adapted ACT-R framework does not connect the motor module to the environment and motoric actions do not alter the environment. This is because the ACT-R motoric module would need an upgrade to allow for extensive motoric control needed during driving. However, if an online interface were implemented the possibility of connecting ACT-R to a prototypic vehicle would be possibly. Nevertheless, the integration of external sensory data does allow simpler modeling of real-world environments with reproducible interpretation of data according the SEEV theory.

Discussion

SEEV-VM is far from finished, there are still some open issues. The module does not support some features found in the default ACT-R vision module. E.g. it has no explicit attended field but introduces inhibition of return to reduce changes to fixate the same object repeatedly. SEEV-VM aims to offer a less precise way to instruct visual attention, giving the model more flexibility to react to variations and dynamics in known situations. As an example, the model does not know precisely where a traffic light is located, but it knows where to expect one. Combined with bottom-up processing capabilities the vision system will find a certain traffic light. The productions can work with a special chunk for the object type of a traffic light, reducing the burden to share productions with different models. It is possible to establish a library of object chunk definitions.

Arbitrary vision chunks not only increase maintainability of a model, but also allow vastly different simulation environments and affordances to connect to an ACT-R agent. This flexibility might also have a downside, as it does not restrict modelers to plausible models: A vision object chunk can contain unrealistically complex information.

In a future version, we plan to standardize the communication protocol to provide an easy to use API to establish a connection between SEEV-VM and simulation environments. It can be envisioned that many different modules (motor, audio) connect to the same simulation server, that delegate commands and information between agent and world. JNI (Hope, Schoelles & Gray, 2014) already provides this functionality and could be modified to support SEEV-VM.

The module has not yet been validated. The SEEV model works well (Wickens, McCarley & Steelman-Allen, 2009), but it's less detailed than its SEEV-VM adaption. In SEEV-VM attention is directed at objects and not at displays that could span entire scenes (e.g. rear window of a car). The module is able to work in the same way, but in regard to ACT-R, cognition is modelled on a finer resolution, requiring chunks of information at a certain point in time. The next step will be to conduct a validation study to evaluate SEEV-VMs approach to modelling visual attention.

Predicting a scan path is essential in determining whether unexpected visual stimuli were recognized or not. In the module, it is very easy to guide attention towards a visual location (not necessary towards an object) by setting relevance of an AoR. However, finding plausible relevance values is not trivial. Relevance and expectancy (consumption and refresh values) can be seen as results of a learning process, allowing to model experts and novices. In a future work, SEEV-VM has to be validated and we expect to change some parts of the implementation like the default set of parameters. While the SEEV-VM benefits from large flexibility, the subsymbolic parameters need to restrain it in such a way that realistic behavior is generated.

Conclusions

The SEEV-VM module adaptation offers unique development by incorporating the SEEV theory as a foundation for visual attention in ACT-R. While the modeler holds the task of attributing the salience of objects in the environment, SEEV-VM enables ACT-R modes to perceive semantically annotated real-world scenes. By integrating top-down and bottom-up processing it allows the model to react to unexpected events. Setting up AoRs is an easy and abstract way to instruct the visual system, thereby allowing the model to see unexpected things or process objects that are not explicitly represented by productions.

The current substantial effort necessary of modeling visual information in ACT-R needs to be improved to increase the applicability of cognitive modeling to real-world usability testing and to integrate it into applications. Especially tasks and environments that require a lot of visual information are thus far difficult to analyze with ACT-R. This includes the automotive sector in which rich environments can influence drivers in a plethora of facets. The SEEV-VM module adaptation provides the possibility of connecting vehicular BUS-communication to ACT-R and therein deliver semantic data from the surrounding. Multiple and quickly changing scenes are far easier to incorporate into cognitive models, offering the possibility of modeling human-machine-interaction in the vehicular context.

References

Anderson, J. R. et al., 2004. *An Integrated Theory of the Mind*. Psychological Review, 111(4), pp. 1036-1060.

Doerr, L., Russwinkel, N. & Prezenski, S., 2016. *ACT-Droid: ACT-R interacting with Android applications*. In: Proceedings of the 14th International Conference on Cognitive Modeling. University Park, PA: Penn State: s.n.

Gunzelmann, G., Moore, L. R., Salvucci, D. & Gluck, K. A., 2011. *Sleep loss and driver performance: Quantitative predictions with zero free parameters*. Cognitive Systems Research, 12(2), pp. 154-163.

Halbrügge, M., 2013. *ACT-CV: Bridging the Gap between Cognitive Models and the Outer World*. In: E. Brandenburg, et al. eds. Grundlagen und Anwendung der Mensch-Maschine-Interaktion - 10. Berliner Werkstatt Mensch-Maschine-Systeme. Berlin: Universitätsverlag der TU Berlin, pp. 205-210.

Haring, K. S., 2012. *A Cognitive Model of Drivers Attention*. In: Russwinkel, Drewitz & van Rijn, eds. Proceedings of the 11th International Conference on Cognitive Modeling, Berlin. Berlin, Germany: Universitaetsverlag der TU Berlin.

Henderson, J. M., 2003. *Human gaze control during real-world scene perception*. Trends in cognitive sciences, 7(11), pp. 498-504.

Hope, R. M., Schoelles, M. J., & Gray, W. D., 2014. *Simplifying the interaction between cognitive models and task environments with the JSON Network Interface*. Behavior research methods, 46(4), pp. 1007-1012.

ISO 11898-1:2015, 2015. *Road vehicles - Controller area networt (CAN) - Part 1: Data link layer and physical signaling*.

ISO 17458-1:2013, 2013. *Road vehicles - FlexRay communications system*.

Kosanke, H. & Russwinkel, N., 2016. *Doing all at once? Modeling driver workload in an abstract multitasking scenario*. Abstractband der 58. Tagung experimentell arbeitender Psychologen (TeaP).

Nyamsuren, E. & Taatgen, N. A., 2013. *Pre-attentive and attentive vision module.* Cognitive Systems Research, Issue 24, pp. 62-71.

Oliva, A., Torralba, A., Castelhano, M. S. & Henderson, J. M., 2003. *Top-down control of visual attention in object detection.* Proceedings 2003 International Conference on Image Processing, pp. 253-256.

Russwinkel, N., Prezenski, S., Dörr, L. & Tamborello, F., 2018. *ACT-Droid Meets ACT-Touch: Modelling Differences in Swiping Behavior with Real Apps.* Proceedings of the 16th International Conference on Cognitive Modeling (ICCM 2018), 21-24 07, pp. 120-125.

Salvucci, D. D., 2000. *A model of eye movements and visual attention*. Proceedings of the International Conference on Cognitive Modeling, pp. 252-259.

Salvucci, D. D., 2005. *Modeling tools for predicting driver distraction*. Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting.

Salvucci, D. D., 2009. *Rapid prototyping and evaluation of in-vehicle interfaces.* ACM Transactions on Human-Computer Interaction, 16(2).

Salvucci, D. D. & Taatgen, N. A., 2008. *Threaded cognition: An integrated theory of concureent multitasking*. Psychological Review, 115(1), pp. 101-130.

Wickens, C. D., 2015. *Noticing events in the visual workplace: The SEEV and NSEEV models.* In: R. R. Hoffman, et al. eds. Part VI - Perception and Domains of Work and Professional Practice. Cambridge: Cambridge University Press, pp. 749-768.

Wickens, C., McCarley, J., & Steelman-Allen, K., 2009. *NT-SEEV: A model of attention capture and noticing on the flight deck*. In Proceedings of the human factors and ergonomics society annual meeting. Sage CA: Los Angeles, CA: Sage Publications. Vol. 53, No. 12, pp. 769-773.

Syntactic Priming Depends on Procedural, Reward-Based Computations: Evidence from Experimental Data and a Computational Model

Yuxue Cher Yang (chery@uw.edu)
Department of Psychology, University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Andrea Stocco (stocco@uw.edu)
Department of Psychology and Institute for Learning and Brain Sciences (I-LABS), University of Washington
Campus Box 351525, Seattle, WA 98195 USA

Abstract

Syntactic priming (SP) is the effect by which, in a dialogue, the current speaker tends to re-use the syntactic constructs of the previous speakers. SP has been used as window into the nature of syntactic representations within and across languages. Because of its importance, it is crucial to understand the mechanisms behind it. Currently, two competing theories exist. According to the surprisal theory, SP is driven by the mismatch with internal predictions and enhanced by factors that enhance surprise (i.e., use of low-frequency verbs). According to the declarative theory, SP is driven by the re-activation of declarative memory structures that encode template structures. Here, we propose a third and novel hypothesis, namely, that SP is driven by the successful application of procedural knowledge, in agreement with Ullman's model. This hypothesis makes the unique prediction that SP will be reversed when the prime sentence includes grammatical errors, but not semantic errors. The theory is supported by a computational model. An experiment confirmed the prediction of the theory.

Keywords: Syntactic Priming, Procedural Knowledge, Reinforcement Learning, Computational Modeling

Introduction

Syntactic Priming (SP, also known as "Structure Priming") is the linguistic phenomenon by which speakers tend to re-use syntactic structures across utterances (Bock,1996). Its existence is often touted as the strongest evidence that the same syntactic mechanisms are used in both language comprehension and language production. As such, manipulations that affect SP can be used to gather insight into how brain perceives, represents, and applies syntactic structures. For example, two notable studies (Loebell & Bock,2003;Hartsuiker, Pickering, & Veltkamp,2004) have show that that SP effects occur across languages, demonstrating that syntactic structure is represented in a way that is language-independent.

In this paper, we will use a novel manipulation of SP effects to investigate whether syntactic structures are represented within declarative or procedural memory. Our results, backed by computational models, strongly suggest that SP is based on procedural representations, and that these representation are learned and refined through Reinforcement Learning.

Background

In the past few decades, many researchers have attempted to determine the most likely mechanistic explanation for SP (Hartsuiker et al.,2004;Reitter, Keller, & Moore,2011;Chang, Dell, & Bock,2006). Experimental studies show that a range of factors could impact the strength of priming. For example, the priming effect is enhanced by the presentation of multiple primes, which is referred as the cumulativity of SP (Jaeger & Snider,2008). Not only the occurrence of primes matters, the lexical overlapping between prime and target also enhances priming, which is known as the lexical boosting effect (Pickering & Branigan,1998). Moreover, there is evidence for an inverse frequency interaction, showing that that the less frequently used syntactic structures are associated with with stronger priming effects (Jaeger & Snider,2008).

These effects have been used to support different underlying mechanisms that might account for SP. A main source of disagreement between these putative mechanisms is whether syntactic processing is relying on declarative or procedural representations. A group of researchers, for example, advocate a short-term residual activation mechanism account (Snider,2008;Jaeger & Snider,2008;Pickering & Branigan,1998) that implies a declarative representation, while another group of researchers believe that syntactic persistence is depending on implicit learning mechanisms (Chang et al.,2006;Bock & Griffin,2000) that point to a procedural representation. By incorporating both short-term activation account and long-term implicit account, a further dual mechanism account, Declarative/Procedural model of language is proposed by (Ullman,2004). Based on different mechanisms, different computational models have been developed to account for structural priming effects.

Most psycholinguistic studies have investigated syntactic priming effects using carefully controlled experimental items, ensuring that the linguistic stimuli have no mistakes and are produced flawlessly. However, in natural conversation, disfluencies and errors are very common when people are speaking. Usually, erroneous message is considered as interference that either slows down the processing or impedes peoples comprehension. Speech errors include ungrammatical construction, inappropriate word choice, ambiguous meaning, or absolute nonsense. Even though people may ignore minor speech errors in daily conversation, there is evidence that erroneous information does affect language processing, and might provide a further cue to the underlying representation of syntax. For example, people often change their mind and correct themselves mid-sentence while speaking. Slevc and Ferreira (2013) examined the priming effect in the context of correcting speech errors. They found that SP is significantly reduced when primes are corrected to the alternative syntactic structure.

The prediction error (i.e., surprise) associated with the syntactic structure of prime also affects subsequent language processing. There was evidence that the more surprising the prime is, which means higher prediction errors, the more likely to expect the same structure would occur later (Jaeger & Snider,2008).

The role played by errors in SP introduces a third point of view on the nature of SP, which can be catalogued under the "procedural" account. According to this point of view, syntactic structures are represented procedurally and their selection is guided by their perceived utility in terms of Reinforcement Learning, i.e., their estimated future amount of "rewards" or positive feedback signals (Sutton, Barto, et al.,1998). It is widely accepted that procedural knowledge, in general, is refined in a Reinforcement Learning-like manner through the backpropagation of reward or feedback signals. In fact, procedural knowledge and reward signals share the same computational substrate, in the dopamine-rich basal ganglia (Schultz, Dayan, & Montague,1997;Yin & Knowlton,2006). Furthermore, although the basal ganglia are not considered part of the cortical language network, an increasing number of studies have shown their involvement in language processing (Friederici,2006;Stocco, Yamasaki, Natalenko, & Prat,2014).

The connection between reward signals and procedural knowledge is apparent in some prominent general theories of cognition. For example, in the ACT-R cognitive architecture (Anderson,2009;Anderson et al.,2004), procedural knowledge is represented as production rules or simply *productions*, and productions are typically used to represent syntactic micro-operations in ACT-R models of language processing (Lewis & Vasishth,2005;Stocco & Crescentini,2005). But, in ACT-R, productions are selected on the basis of their expected , a scalar quantity that represents future rewards and is updated through repeated feedback signals according to a standard Reinforcement Learning rule:

$$U_{t+1}(p) = U_t(p) + \alpha \times (R_t - U_t(p)) \quad (1)$$

where $U_t(p)$ represents the utility U of production p at time point t.

In the case of linguistic phenomena, feedback signals could be provided directly by the process of successfully comprehending or producing a sentence. Thus, according to this view, SP would be the effect of increased utility of a syntactic structure following its successful use in comprehension.

If that is the case, we expect that ungrammatical sentences, in which rules are applied *unsuccessfully* and lead to a error signal and a re-analysis of a sentence, would result in negative feedback signals. These negative feedback signals would ultimately *decrease* the utility of the corresponding production, thus making the application of the same syntactic structure less likely to occur.

In this study, we set forward to test this alternative, RL-based account for syntactic priming, and to answer the question of whether perceiving incorrect linguistic information such as ungrammatical syntactic constructions would affect peoples subsequent language representation, particularly in syntactic choices of production. Furthermore, we will attempt to explain the observed patterns under Reinforcement Learning theory and simulate the behavioral results using ACT-R model.

Theoretical Hypotheses

Based on the proposed theories of SP, we can derive three different predictions about the effect of syntactically incorrect primes (See Figure 1). Across all predictions, we expect that syntactic priming effect will occur regardless of syntactic correctness. Specifically, the proportion of producing same construction is expected to be higher than producing alternative construction. We also expect that the priming effect will be different depending on whether the syntactic structure of prime is correct or not.

According to a purely declarative model (as exemplified, for instance, by Reitter's 2011 model), an ungrammatical prime should not have any differential SP effect than a grammatical one. In as much as the prime sentence can be correctly interpreted despite the syntactic error (and, in our experiment, we made sure this is the case), the same grammatical structure would be retrieved, thus causing the same activation boost for subsequent use. Thus, our Hypothesis 1, driven by the repetition between prime and target, states that there is no difference between grammatical and ungrammatical primes.

According to procedural, prediction-driven model (as exemplified by Jaeger & Snider,2008 and Snider,2008's exemplar-based model), the ungrammatical prime, being a low-frequency and unexpected structure, would generate greater surprisal and therefore *enhance* priming effect for same constructions production, but to weaken priming in alternative construction production. Specifically, Hypothesis 2 states that priming with ungrammatical sentence makes people more likely to produce same constructions, and less likely to produce alternative constructions than priming with grammatical one. Finally, according to our procedural/RL account, SP is due to the update of the perceived utility of a procedural syntactic structure, which is increased for successfully parsed (grammatical) sentences and decreased following unsuccessfully parsed (ungrammatical) ones. Driven by reward, Hypothesis 3 predicts an opposite pattern as Hypothesis 2, stating that, priming ungrammatical sentences is expected to increase the likelihood of producing alternative structures than those used in the priming sentences.

To explicitly formulate our hypothesis, we implemented it as an ACT-R model[1]. The model performs a simplified version of canonical SP task, first comprehending a sentence (in

[1] The code for all the models described in this paper is available on our laboratory's GitHub page: http://github.com/UWCCDL/SyntaxPriming

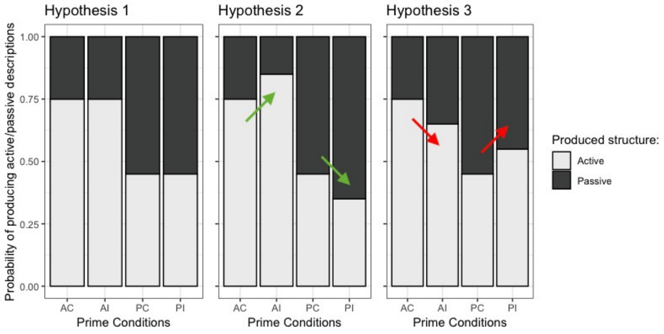


Figure 1: Three hypotheses driven by on different predictions. (white indicates active-form prime., gray indicates passive-form prime). Hypothesis 1: Declarative, driven by activation, predicts no effect of syntactic errors. Hypothesis 2: Driven by expectations, it predicts enhanced priming for (unexpected) ungrammatical sentences. Hypothesis 3: Driven by reward, it predicts reduced priming for ungrammatical sentences.

either active or passive form) and then producing a sentence to describe a picture. Both comprehension and production depend on the use of two production rules that implement the active and the passive sentence structures. In comprehension, these rules are used to mediate from the underlying sentence to its higher-level semantic representation. In language production, these rules are used to create a mental plan of the sequence of words to produce a description of the picture. Feedback signals are generated by detecting whether the comprehended sentence is grammatically correct or not. For simplicity, the process of parsing a sentence is drastically simplified (not unlike in Reitter et al.,2011), so that all the sentence information is available at once in a single visual "chunk" of information in ACT-R and feedback signals are only generated at the end of the comprehension process.

To examine the predictions of our model, we conducted a parameter space partitioning analysis of the model's behavior, and found that, across different initial utility values of the two syntactic structures and different reward values, the model produces the qualitative pattern of Figure 4.

To test between these alternative hypothesis, we conducted a novel SP experiment, introducing the novel manipulation of syntactic grammaticality of the priming sentences.

Materials and Methods

Participants

Ninety participants (35 female, 54 male, 1 other) were recruited online through Amazon Mechanical Turk, and performed the experiment in exchange for monetary compensation. Ethnicity includes 51.1% White, 36.7% Asian, 6.7% African American, 3.3% Latino or Hispanic American, and 2.2% Others. All participants were screened through a pre-experimental survey that gathered information about their language experience and background; only native English speakers without any history of brain damage, reading problems, nor language-related disorder were allowed to proceed to the experiment. Twenty-one were later excluded for failing to construct complete sentences in the language production task. The experimental protocol and inclusion criteria were approved by the Institutional Review Board at the University of Washington.

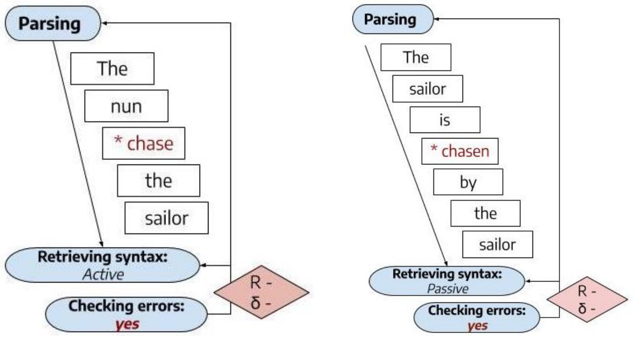


Figure 2: Two priming examples of the simple Reinforcement Learning model. Left: modeling AI priming. Right: modeling PI priming. White rectangles represents chucks encoding words. The blue rounded rectangles represents productions: *parsing* - parse in the prime; *retrieving syntax* - retrieve corresponding syntactic structure of the prime; *checking error* - check whether there is grammar errors in the prime. Diamond shapes represent feedback, either positive or negative. R indicates the reward term in Eq. 1, and δ reflects the reward prediction error term $R_t - U_t(p)$. When the model detects error, it sends a negative feedback signal to all the previous productions that have fired since the last reward. The predictions of this model are illustrated in Fig. 6

.

Materials

This picture description task is modified based on Hardy, Messenger, and Maylor's experiment (2017). A total of 36 trials with prime target pairs were created. Each picture is depicting a ditransitive action involving an agent and a patient. The verb of the action is printed under each picture. The prime sentence is either active-tense form grammatically correct (AC), passive-tense form grammatically correct (PC), active-tense form grammatically incorrect (AI) or passive-tense form grammatically incorrect (PI).

Ungrammatical prime sentences in the Passive Incorrect syntax condition (PI) were generated using seemingly correct but non-existing past participles modeled after existing verbs, such as "chasen" instead of "chased", "slapt" instead of "slapped", and "shooted" instead of "shot". In half of the trials within each condition, (N = 18 total), the prime picture and prime sentence are perfectly matched, while in the other half, the prime sentence is modified as semantically incorrect by which the identity of either agent or patient is wrong. This latter manipulation was designed to both make sure that participants were performing the task correctly and to separately measure the effect of syntactic errors from semantic errors.

Design

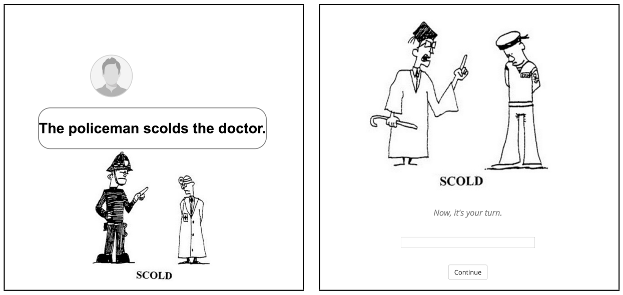
This study is a 2 × 2 × 2 within-subject design, with three the factors being prime syntax (active vs. passive), grammatical correctness (correct vs. incorrect), and semantic correctness (correct vs. incorrect). In our notation, 4 syntax conditions: AC, AI, PC, PI x 2 semantic conditions: SC (semantically correct) and SI (semantically incorrect). Because, previous studies have demonstrated a stronger syntactic priming effect as prime and target are overlapping (Pickering & Branigan,1998), in this study prime and target always share the same action verb. The combination of three independent variable pairs are pseudo-randomized so in each syntax condition (AC, AI, PC, PI), each verb only occurs once, and each verb is modified as both semantic-correct and semantic-incorrect form.

Procedure

Most SP experiments make use of realistic, in-person dialogue between two participants, one of which is a confederate. The confederate verbally utters the primes and the participants responses are recorded for transcription. To simulate this seemingly realistic dialog situation online, the study described here used deception to convince participants that they were paired with another online "partner" and they were to take turns providing a description for a sentence and verifying the accuracy of their partner's description. In fact, there was no paired partner and all sentences typed by the partner were decided beforehand. At the end of the study, participants were fully debriefed about the use of deception.

In the online task, participants see a prime picture and are asked to verify whether the sentence constructed by the partner was correctly describing the picture or not. Followed by the verification task, there is a picture description task (see Figure 3). In the picture description phase, a picture and an appropriate verb are given, and participants need to type a sentence to describe the picture using the given verb. Participants are told that the game is proceeding in which the partner and the participant alternate between verifying if sentence-picture pair is matching, and constructing a sentence to describe the picture to the other. The game sets a randomly generated waiting time to simulate the amount of time needed by the fictional partner to type their own description.

The participant needs to complete a pre-screen survey that only eligible ones can continue. After giving consent, participants begin with a three-trial practice phase to familiarize themselves with the procedure. Between verification task and picture description task, the game sets a randomly generated waiting time to simulate verifying period of the "partner". At the end of the study, participants are given the debrief about the deception involved and are asked to complete a post-experiment survey.



The policeman scolds the doctor.

Figure 3: Example trial from the Online ASP Task. Left: During the Verification phase, subjects are asked to verify the congruence between the sentence and the picture. Right: During the Production phase, participants are asked to describe the target picture by typing a complete sentence that contains the given verb.

Results

The responses typed by participants were automatically analyzed with the Natural Language Toolkit (NLTK) package in Python and double-checked manually. Thirteen responses that could not be coded as neither active nor passive sentence were removed for data analysis.

The total of 2507 responses yield 75.79% active descriptions, 24.21% passive-voice. The analysis is conducted with the proportion of producing active out of active and passive responses. As expected, there is a significant effect of syntactic priming (Active vs. Passive), $F(1, 69) = 59.52$, $p < 0.001$, and a significant main effect of syntactic correctness (Syntax-Correct vs. Syntax-Incorrect), $F(1, 69) = 13.28$, $p = 0.001$. As expected, we find that there is no significant effect of semantic correctness on syntactic production. $F(1, 69) = 1.37$, $p = 0.25$.

Post-hoc analyses for significance indicate that the mean proportion of active descriptions is significantly lower in PC condition (M = .64, SD = .32) than that in PI (M = .69, SD = .34), $F(1, 69) = 5.05$, $p = 0.03$. The mean proportion of active descriptions is also significantly lower in the AC prime condition (M = .84, SD = .24) than in the AI conditions (M = .8, SD = .21), $F(1, 69) = 6.09$, $p = 0.01$.

As for the accuracy in verification task, overall accuracy rate is 79.92%. We find a significant effect of syntactic correctness on the accuracy rate $F(1, 68) = 57.66$, $p < 0.001$. People tend to verify picture more accurately when the sentence is grammatically correct (M = .8796, SD = .16) than the sentence is grammatically incorrect (M = .72, SD = .21). We also find that there is a significant effect of syntactic voice on the accuracy rate $F(1, 69) = 16$, $p = 0.001$. The accuracy of verification is significantly higher for active sentences (M = .83, SD = .19) than for passive sentences (M = .77, SD = .21). Interestingly, there is significant interaction effect on accuracy rate between syntactic correctness and syntactic voice, $F(1, 69) = 12.33$, $p = 0.001$.

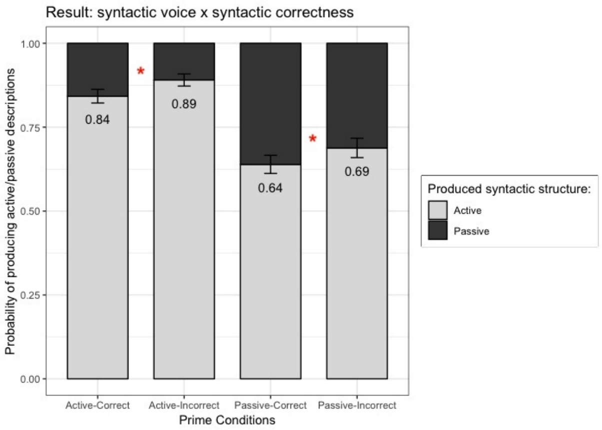


Figure 4: The proportion of active structures across conditions Asterisks "*" denote significant differences between conditions.

Summary

Taken together, the results of our experiment provide a picture that is not entirely consistent with any of the previously discussed models, while the SP was present and robust (albeit less dramatic that in previous studies). Contrary to Reitter's model, there was a robust effect of syntactic grammaticality. These effects, however, did not comply precisely with either of the two competing accounts, that is, the procedural/expectancy and the procedural/RL hypotheses. In the passive sentences, an ungrammatical prime increased the likelihood of producing another active sentence, consistent. However, the data also show that semantic errors do not produce any effect, and, therefore, that the effect of errors can be localized to the processes of syntactic parsing.

A Sequential Procedural Model

One possible explanation for the lack of correspondence between the experimental results and our model is that the our procedural model was too naïve and did not appropriately take into account the different ways in which active and passive sentences are parsed. To explore this issue, we created a second computational model (See Figure 5).

This second model closely follows the structure of the procedural model described above. However, the new model simulates, at least partially, the sequential and incremental nature of sentence parsing. In particular, while the first model immediately detects the structure of the sentence (active vs. passive) and generates all feedback signals at the very end of the comprehension process, the second model delays the choice of the correct syntactic form until the first verb is encountered, and generates feedback signals both the end (when all sentences are successfully understood) and as soon as the first incorrect word is found (for ungrammatical ones).

This creates a novel asymmetry between the ungrammatical, active (AI) and ungrammatical, passive (PI) sentences. In the case of passive sentences, the first verb form encountered by the model is the word "is" (as in "the robber *is* chased (...)"); when the word "is" is encountered, the model can confidently select the production rule that encodes the passive structure. The grammatical mistake is then detected immediately thereafter (as in "the robber is *chasen* (...)"), thus generating an negative feedback that decreases the utility of the passive form. In this condition, therefore, the effect of grammaticality is identical to what was predicted by the previous model.

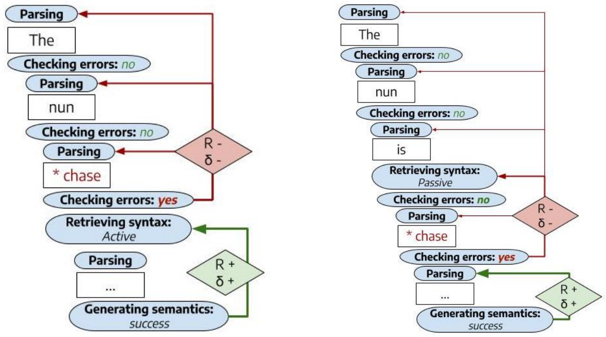


Figure 5: Two examples of the Sequential Procedural model. Left: Parsing of an ungrammatical, active sentence (AI). Right: parsing of an ungrammatical, passive (PI) sentence. This revised model explains both the activation boost found in AI priming and the activation drop in PI priming (see Fig. 6, Right)

In the case of ungrammatical active sentences, however, the first verb form is also the first word for which a negative feedback signal can be generated (as in "the robber *chase* (...)"). In this case, the negative feedback is generated at the same time as the active sentence structure is selected, and, thus, does not affect the utility of the corresponding production. When the model successfully completes the sentence comprehension goal, a *positive* feedback signal is generated that propagates back to active form, thus increases its utility even if the sentence was ungrammatical.

This dynamic is further complicated by the fact that, to be selected in the face of a grammatically incorrect verb, a mechanisms of *procedural partial matching* had to be enabled. With this mechanism, productions are allowed to be selected even if their requirements are not perfectly satisfied. The price to pay for this imperfect selection is a temporary reduction in the associated utility. That is, instead of using a production's "true" utility $U_t(p)$, Eq. 1 uses the the reduced term $U_t^*(p)$:

$$U_t^*(p) = U_t(p) - MP(p) \tag{2}$$

where $MP(p)$ is the *mismatch penalty*, a fixed cost associated to applying a production rule to a condition in which not all the requirements are verified. This reduction reflects an intuitive greater uncertainty in the predicted future rewards for cases (such as ungrammatical sentences) in which productions are applied outside of their ideal conditions.

In turn, this reduced expectation affects the RL-based adjustments of utility. This is because these adjustments, according to the ACT-R theory and Eq. 1), reflect the magnitude of the reward prediction error δ_t, which is the difference between effective feedback signal and expected utility: $\delta_t = \alpha \times (R_t - U_t(p))$. It is easy to see that, for ungrammatical sentences, $U_t^* < U_t$ (because of the penalty match in Eq. 2) and, therefore, $\delta_t^* > \delta_t$: the as the utility U gets smaller, the adjustment δ_t gets larger, resulting in even greater benefit for the active form when it is selected while successfully parsing an ungrammmatical sentence.

To test our theory, we simulated the behavior of this model under different parameters. We found that the model consistently yields results consistent with our data. Fig. 6 depicts prototypical results (using $R = 1.0$, $\alpha = 0.2$, and $MP = 0.2$). Specifically, while the effect of grammaticality on the SP of passive sentences remains unchanged, the effect of grammaticality for active sentences either disappears (yielding equal probabilities of using the active form after a grammatical and an ungrammatical sentence) or results in *higher* rates of active sentences following ungrammatical primes.

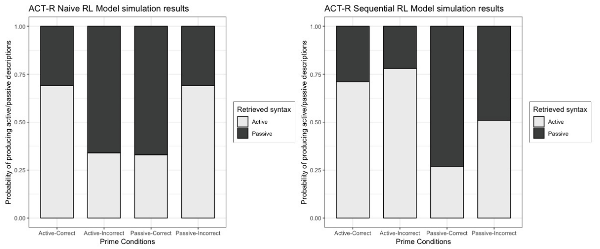


Figure 6: Simulation results from the two Reinforcement Learning models of syntactic priming. Left: Predictions of the simple, naïve models. Right: prediction of the Sequential model. The sequential model correctly predicts the general pattern of the experimental findings in Fig. 4.

Discussion

As demonstrated in many syntactic priming studies, people tend to re-use the same syntactic structures they are primed with. Consistent with this body of literature, our experiment shows a overall syntactic priming effect for active and passive structures, regardless of syntactic correctness and semantic correctness. This implies that the tendency of reproducing primed syntactic structures persists even if the linguistic information is noisy and erroneous.

In addition, our experimental results showed that syntactic priming is modulated by the grammaticality of the priming sentence. This result poses difficulty for purely declarative accounts (Reitter et al.,2011), which ascribe priming effects to the frequency and recency of syntactic structure retrieval. Furthermore, we found that this effect was specific to *grammatical errors*, and not to semantic errors (i.e., incorrectly labelled figures), thus restricting the effect to syntactic processes and excluding a general effect of surprise or attention to errors.

In general, these results support the idea that the priming effect of syntactic structures is dependent on procedural, rather than declarative memory, thus suggesting that syntactic structures are represented procedurally (Ullman,2004).

However, contrary to our expectations, our results show that participants actually always produce more active sentences following an ungrammatical sentence, regardless of the syntactic voice of the prime. This interesting pattern is at odds with both the declarative memory accounts (Hypothesis 1), the Activation Spreading account (Hypothesis 2), and a naïve procedural memory account (Hypothesis 3).

We found that this effect could be accounted for if our original naïve procedural model is expanded to included sequential parsing. Under these conditions, the order in which syntactic forms are selected and grammatical feedback signals are delivered becomes important. In particular, in ungrammatical active sentences, the negative feedback signal is delivered before the active form is selected, and the adjustment to the expected rewards of active structures is greater, thus reproducing the effects we found in our data.

Although successful, our second model is limited by the fact of having being designed post-hoc. To test its validity, the same experiment should be replicated using different syntactic structures, so that new predictions can be made and tested. Even within these limitations, however, the models describe herein have two important implications. First, they highlight the role of basic reinforcement learning mechanisms in learning, whose contribution might shed light on the basic computations underlying syntactic parsing as well as the contributions of subcortical structures to language (Hernandez et al.,2019). Second, our results highlight the importance of detailed computational models to explain psycholinguistic effects.

Acknowledgments

This research was made possible by a Top Scholar award from the University of Washington to YCY, and by a grant from the Office of Naval Research (ONRBAA13-003) to AS.

References

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Bock, K. (1996). Language production: Methods and methodologies. *Psychonomic Bulletin & Review*, *3*(4), 395–421.

Bock, K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, *129*(2), 177–192.

Chang, F., Dell, G. S., & Bock, K. (2006). Becoming syntactic. *Psychological Review*, *113*(2), 234–272.

Friederici, A. D. (2006). What's in control of language? *Nature neuroscience*, *9*(8), 991.

Hardy, S. M., Messenger, K., & Maylor, E. A. (2017). Aging and syntactic representations: Evidence of preserved syntactic priming and lexical boost. *Psychology and Aging*, *32*(6), 588–596.

Hartsuiker, R. J., Pickering, M. J., & Veltkamp, E. (2004). Is syntax separate or shared between languages? cross-linguistic syntactic priming in spanish-english bilinguals. *Psychological science*, *15*(6), 409–414.

Hernandez, A. E., Claussenius-Kalman, H. L., Ronderos, J., Castilla-Earls, A. P., Sun, L., Weiss, S. D., et al. (2019). Neuroemergentism: A framework for studying cognition and the brain. *Journal of neurolinguistics*, *49*, 214–223.

Jaeger, T. F., & Snider, N. (2008). Implicit learning and syntactic persistence : Surprisal and cumulativity. *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 827812).

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, *29*(3), 375–419.

Loebell, H., & Bock, K. (2003). Structural priming across languages. *Linguistics*, *41*(5).

Pickering, M. J., & Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and language*, *39*(4), 633–651.

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive science*, *35*(4), 587–637.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*(5306), 1593–1599.

Slevc, L. R., & Ferreira, V. S. (2013). To err is human to structurally prime from errors is also human. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *39*(3), 985–992.

Snider, N. (2008). An exemplar model of syntactic priming. unpublished doctoral dissertation, stanford university.

Stocco, A., & Crescentini, C. (2005). Syntactic comprehension in agrammatism: A computational model. *Brain and Language*, *95*(1), 127–128.

Stocco, A., Yamasaki, B., Natalenko, R., & Prat, C. S. (2014). Bilingual brain training: A neurobiological framework of how bilingual experience improves executive function. *International Journal of Bilingualism*, *18*(1), 67–92.

Sutton, R. S., Barto, A. G., et al. (1998). *Reinforcement learning: An introduction* (Vol. 135). MIT Press.

Ullman, M. T. (2004). Contributions of memory circuits to language: the declarative/procedural model. *Cognition*, *92*(1-2), 231–270.

Yin, H. H., & Knowlton, B. J. (2006). The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience*, *7*(6), 464.

Multi-Armed Bandit Problem:
A New Belief-Resilience Algorithm

Nick Hollman (nhollma@umich.edu)
Weinberg Institute for Cognitive Science, 500 Church St,
Ann Arbor, MI 48109 USA

Qianbo Yin (ygrayson@umich.edu)
Weinberg Institute for Cognitive Science, 500 Church St,
Ann Arbor, MI 48109 USA

Keywords: Multi-Armed Bandit; Belief Resilience; Exploration and Exploitation; Reinforcement Learning

Introduction

The Multi-Arm Bandit (MAB) Problem captures a dilemma in decision-making under uncertainty. Agents are faced with *n* choices that have various unknown rewards, in which they can either exploit choices with greater certainty for rewards or explore the unknown choices hoping for a better result. Ultimately the goal of each agent is to maximize the total rewards as much as possible.

In our current project, we develop a new algorithm based on the resilience of a belief each agent has towards the expected reward. As more information accumulates, the agent's belief becomes more resilient and consequently helps the agent to make better choices.

Existing Algorithms

The Multi-Armed Bandit problem is a well-researched problem in reinforcement learning. To test the performance of our new algorithm, we will compare it with the following previously developed algorithms:

Epsilon Greedy

1 - $\boldsymbol{\varepsilon}$ probability of exploitation

Epsilon First

$\boldsymbol{\varepsilon}$ * N number of random trials (exploration) followed by a phase of exploitation

Epsilon Decreasing

Same as epsilon greedy, but with a decreasing $\boldsymbol{\varepsilon}$:
($\boldsymbol{\varepsilon}$ = 1 / *n* + 1)

Pure Random

Arm is selected at random on each trial.

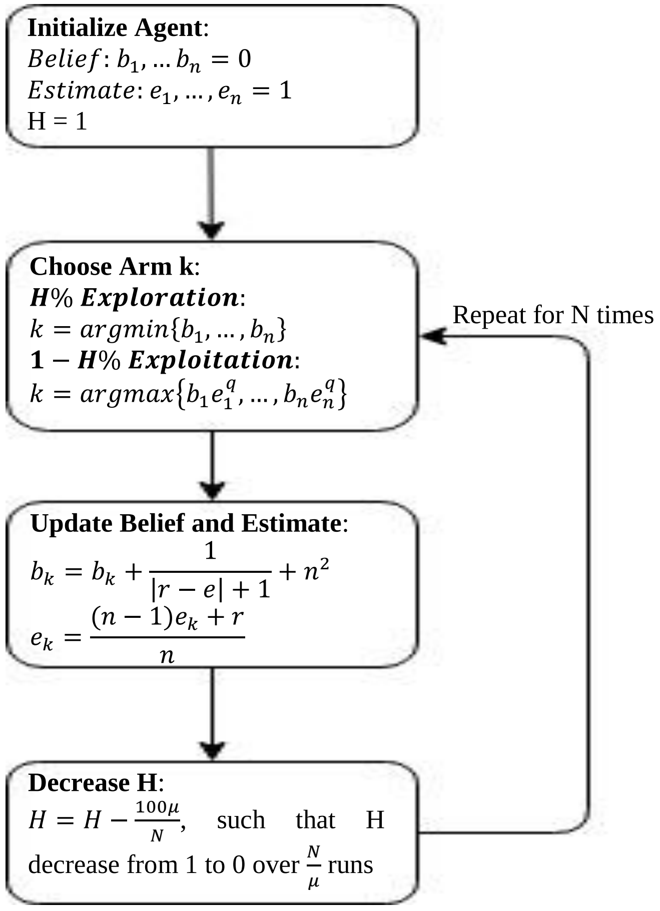
Upper Confidence Bound

Probability of choosing an arm is proportional to the probability of that arm giving the highest payoff.

Belief Resilience Algorithm

This algorithm is built on the assumption that a belief towards an expected reward falls on a spectrum of resiliency. Resiliency in beliefs relates to the amount of evidence and strength of justification. If a belief is low resilience, it has a high chance to be changed based on future evidence.

According to this algorithm, both the belief resiliency and estimation for reward are used in decision making. The exploration phase aims at increasing belief on all bandits, and the exploitation phase aims at optimizing robust, high rewards. The algorithm is formulated in Figure 1.



Initialize Agent:
$Belief\colon b_1, \dots b_n = 0$
$Estimate\colon e_1, \dots, e_n = 1$
H = 1

Choose Arm k:
$\boldsymbol{H\%\ Exploration}$:
$k = argmin\{b_1, \dots, b_n\}$
$\boldsymbol{1 - H\%\ Exploitation}$:
$k = argmax\{b_1 e_1^q, \dots, b_n e_n^q\}$

Repeat for N times

Update Belief and Estimate:
$$b_k = b_k + \frac{1}{|r - e| + 1} + n^2$$
$$e_k = \frac{(n-1)e_k + r}{n}$$

Decrease H:
$H = H - \frac{100\mu}{N}$, such that H decrease from 1 to 0 over $\frac{N}{\mu}$ runs

Figure 1: Belief Resilient Algorithm

Results and Discussion

First, we tested the undetermined parameters q and μ in the Belief-Resiliency Algorithm, generating the reward graph as a function of q and μ, shown in Figure 2. Therefore, we conclude the best parameter for the algorithm.

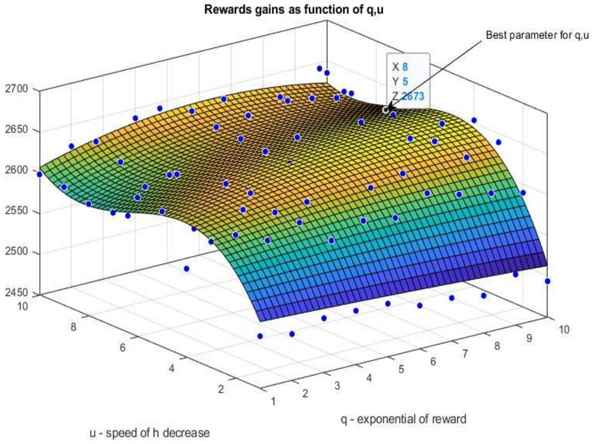


Figure 2: Determine parameters q,u in Belief-Resilience Algorithm. (z-axis: Total Reward)

After determining the parameters in the algorithm, we tested the Belief-Resilient Algorithm against the existing MAB algorithms. Results shown in Figure 3, 4 and 5.

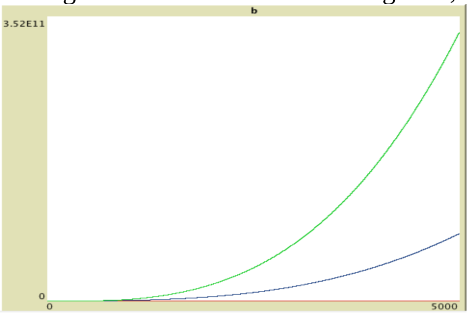


Figure 3: Belief factor (b) increase as a function of trials (plotting 4 arms)

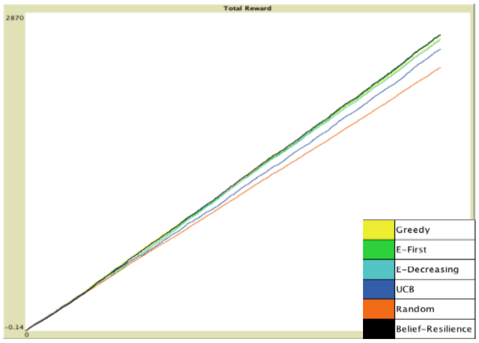


Figure 4: Comparison of 6 different algorithms (Total Rewards)

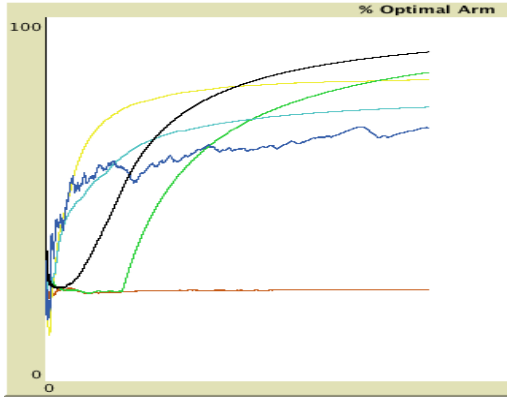


Figure 5: Percentage of Optimal Choices) for 6 different algorithms

Conclusion

After implementing and testing the Belief-Resilience Algorithm, we conclude that this algorithm competes with the standard existing reinforcement learning algorithms, with the optimal parameter q=10 and $\mu = 8$. In some cases, the new algorithm outperforms the leading algorithms in MAB paradigm. Generalizing the idea of belief resiliency in decision making, the robustness of belief can play a crucial role in evaluating a certain decision. Finally, we argue that the Belief-Resilience Algorithm, inspired by human beliefs and decision-making, is potentially an efficient algorithm of human decision making.

Future Directions

In further research, we would like to construct a more sophisticated relationship between *b* and the estimated reward in the Belief Resilience Algorithm. In addition, we would like to further expand the Multi-Armed Bandit problem to more diverse settings to model real-life decision-making situations.

Acknowledgment

We thank the Weinberg Institute for Cognitive Science (WICS) at the University of Michigan, Ann Arbor for its generous support. We highly appreciate the Undergraduate Research Symposium (URS) for providing us the opportunity to present our preliminary results.

References

SUTTON, RICHARD S. BARTO, ANDREW G. (2018). *REINFORCEMENT LEARNING: An introduction*. Cambridge: MIT Press.

Neural Network Modeling of Learning to Actively Learn

Lie Yu[1], **Ardavan S. Nobandegani**[2,3], **& Thomas R. Shultz**[1,3]
{lie.yu, ardavan.salehinobandegani}@mail.mcgill.ca
thomas.shultz@mcgill.ca
[1]School of Computer Science, McGill University
[2]Department of Electrical & Computer Engineering, McGill University
[3]Department of Psychology, McGill University

Abstract

Humans are not mere observers, passively receiving the information provided by their environment; they deliberately engage with their environment, actively participating in the information acquisition stage to improve their learning performance. Despite being a hallmark of human cognition, the computational underpinnings of this active (or self-directed) mode of learning have remained largely unexplored. Drawing on recent advances in machine learning, we present a neural-network model simulating the process of learning how to actively learn. To our knowledge, our work is the first neural-network model of learning to actively learn. Extensive simulations demonstrate the efficacy of our model, particularly in handling high dimensional domains. Notably, our work serves as the first computational account of the recent experimental finding by MacDonald and Frank (2016) showing that prior passive learning improves subsequent active learning. Our work exemplifies how a synergistic interaction between machine learning and cognitive science helps develop effective, human-like artificial intelligence.

Keywords: Active learning; deep neural networks; deep reinforcement learning; example generation

1 Introduction

Humans are not mere passive observers of their environment, but actively search for information which helps to improve their learning performance (Gureckis & Markant, 2012). For example, we purposefully search for information online to learn about a topic of interest, decide how to interact with an unfamiliar device to learn its functionality, or ask questions from people around us to learn more about them, helping us to interact with them more effectively in the future. Relatedly, past educational research shows that people learn better if the flow of experience is under their control (e.g., Cherney, 2008; Michael, 2006).

Although active (aka self-directed) information acquisition is a fundamental and extensively studied topic in the educational sciences (e.g., Bruner, Jolly, & Sylva, 1976; National Research Council, 1999), it has been comparably understudied in the psychological literature (Gureckis & Markant, 2012; Markant & Gureckis, 2014), with the psychological processes underpinning this mode of learning remaining largely unexplored (Gureckis & Markant, 2012). Experimental studies of human learning are predominantly passive in that the experimenter tightly controls what information is presented to the learner on every trial.

A growing, but highly theoretical, research area in computer science, called active learning, aims to formally characterize the extent to which self-directed information acquisition can speed up learning (see Hanneke, 2014, for a survey). Despite notable theoretical successes (e.g., Hanneke, 2016), this research area has made little contact with the psychological literature, primarily focused on highly abstract learning problems amenable to theoretical investigations, and predominantly investigated mathematically the performance gain obtained by following specific active learning strategies, paying no attention to the key problem of how learners learn their active learning strategies in the first place.

Drawing on recent advances in machine learning (particularly deep reinforcement learning), we present a novel neural-network model of active learning aiming to simulate the process of learning how to actively learn. By conceptualizing the problem as a reinforcement learning task, our neural-network model learns, during the passive phase of learning (wherein the learner passively receives information from their environment) an effective active learning strategy allowing for faster learning. As an instantiation of our active learning model, in this work we focus on the task of category learning (aka classification).

Our model has several notable features elevating its cognitive plausibility. First, our model uses Markov-adjusted Langevin (MAL) (Savin & Deneve, 2014; Moreno-Bote, Knill, & Pouget, 2011; Nobandegani & Shultz, 2017, 2018), a well-known gradient-based Markov chain Monte Carlo (MCMC) method, allowing active search for maximally informative examples in a computationally-efficient manner. Notably, recent work in theoretical neuroscience has shown that MAL can be implemented in a neurally-plausible manner (Savin & Deneve, 2014; Moreno-Bote et al., 2011). MCMC methods are a family of algorithms for sampling from a desired probability distribution, and have been successful in simulating important aspects of a wide range of cognitive phenomena, e.g., temporal dynamics of multistable perception (Gershman, Vul, & Tenenbaum, 2012; Moreno-Bote et al., 2011), developmental changes in cognition (Bonawitz, Denison, Griffiths, & Gopnik, 2014), category learning (Sanborn, Griffiths, & Navarro, 2010), causal reasoning in children (Bonawitz, Denison, Gopnik, & Griffiths, 2014), and cognitive biases (Dasgupta, Schulz, & Gershman, 2016).

Second, to improve its active learning strategy, our model uses *memory replay*: the idea of accessing memories of multiple past events and integrating them to make useful predictions about an action's consequences (e.g., Káli & Dayan, 2004; Lengyel & Dayan, 2008; Momennejad, Otto, Daw, & Norman, 2018). Mounting evidence shows that memory replay supports reinforcement learning and planning (e.g., Ólafsdóttir, Bush, & Barry, 2017; Momennejad et al., 2018).

Finally, our model effectively adapts its learned active-learning strategy as it gradually acquires more knowledge about a learning task. This feature of our model is supported by mounting evidence suggesting that people adapt their strategies according to their knowledge and environmental conditions (e.g., Rieskamp & Otto, 2006; Hoffart, Rieskamp, & Dutilh, 2018; Payne, Bettman, & Johnson, 1988; Bröder, 2003; Pachur, Todd, Gigerenzer, Schooler, & Goldstein, 2011; Lieder & Griffiths, 2017).

Our paper is organized as follows. We begin by introducing our neural-network model, and proceed to show the efficacy of our model with extensive simulations. We conclude by discussing the implications of work for active learning research and point out several fruitful lines of future work.

2 Neural Network Model

Our model consists of three neural network modules:

- **Encoder Network (E-Net)**: This neural network module takes a raw input x_i and ouputs a corresponding state representation s_i. As such, this module simulates perception systems, mapping a stimulus to its representation in psychological space.

- **Classification Network (C-Net)**: This neural network module takes state representation s_i and outputs a class label y_i. As such, this module simulates information processing cortices in the brain supporting concept categorization.

- **Action-Value Network (Q-Net)**: For each representation state s_i (corresponding to raw input x_i), this neural network module, parameterized by a set of weights θ, outputs an *affinity score* $Q(x_i, \theta)$ modeling the learner's confidence in choosing x_i to boost learning. That is, a higher $Q(x_i, \theta)$ corresponds to a higher confidence level. Crucially, the network's output, i.e., affinity scores, encodes information enabling our MCMC method, MAL, to actively search for exemplars most helpful for improving the classification performance of the C-Net.

When searching actively for an informative example x which is likely to maximally improve learning accuracy, our model samples from a target distribution $\pi(x)$ given by:

$$\pi(x) \propto \exp(\beta Q(x, \theta)) \qquad (1)$$

where θ denotes the parameters of the Q-network (i.e., the set of network weights), and $\beta \in \mathbb{R}^{>0}$ is a damping factor.

By assigning higher probabilities to those examples x the Q-network believes to maximally improve learning accuracy (i.e., the classification accuracy of the C-Net), Eq. (1) ensures that sampling from $\pi(x)$ yields effective active learning.

To jointly train the E-Net, C-Net, and Q-Net modules of our neural networks model, we use a novel variant of the well-known Deep Q-learning Algorithm (Mnih et al., 2015); see Algorithm 1. Our novel variant of the Deep Q-learning Algorithm has the added advantage of incorporating MCMC in its functionality (Algorithm 1, Line 8), ensuring that sampling from the target distribution $\pi(x)$ would likely yield informative examples x whose knowledge maximally improves the learner's classification accuracy, thus yielding effective active learning.

Algorithm 1 MCMC-Enhanced Deep Q-Learning Algorithm

```
1:  Initialize replay memory $D$ to capacity $N$
2:  Initialize action-value function $Q$ with random weights $\theta$
3:  Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
4:  Initialize classifier $C$ and encoder $E$ with random weights $w_c$ and $w_e$, respectively
5:  for episode = 1 to $M$ do
6:      Randomly pick an input $x_0$ and encoded state representation $s_0$
7:      for t=1 to $T$ do
8:          With probability $\varepsilon$ sample a random data point $x_t$
9:          Sample a new data point $x_t$ via MCMC with the affinity function:

                    $\pi(x_t) \propto \exp(\beta Q(x_i, \theta))$

10:         Compute $q_0 = Q(s_t, a_0; \theta)$ and $q_1 = Q(s_t, a_1; \theta)$
11:         If $q_0 > q_1$, discard these data and go to step $T + 1$. Otherwise, feed $s_t$ into
            $C$ and update its parameters $w_c$.
12:         Do evaluation on $C$ and obtain reward $r_t$
13:         Set $s_{t+1} = s_t$, store transition pair $(s_t, a_t, r_t, s_{t+1})$ in memory D.
14:         Sample minibatch of transitions $(s_j, a_j, r_j, s_{j+1})$ from $D$
15:         Set $y_j = r_j + \gamma max_{a'} \hat{Q}(s_{j+1}, a', ; \theta^-)$
16:         Perform a gradient descent step on $(y_j - Q(s_j, a_j; \theta))^2$ with respect to $\theta$
17:         For every $N_Q$ steps reset $\hat{Q} = Q$
```

The rationale behind Algorithm 1 is as follows. Line 1 initializes the memory replay capacity of our model. Lines 2-4 randomly initialize the weights of E-Net, C-Net, and Q-Net modules. Crucially, by so doing, we assume no prior knowledge on the part of the learner at the onset of learning. Lines 5-9 (except Line 8) use MCMC to effectively guide the active search toward informative samples, the knowledge of which likely maximally improves learning performance. Line 8, for only a small fraction of times, performs random exploration of the input space during the active learning phase. Being a standard approach in machine learning, Line 8 aims to achieve an effective exploration-exploitation trade-off. Lines 10-12 compute the reward associated with each active learning episode by evaluating learning accuracy on a held-out evaluation set: A higher reward implies that the learning performance of our model has considerably improved by using the samples recommended by the Q-Net module. Line 15 updates the model parameters according to the reward obtained in Line 12. Finally, Lines 12-17 (except Line 15) implement the well-known Q-learning process widely used in modeling model-free reinforcement learning in the machine learning, psychology, and neuroscience literatures (Watkins & Dayan, 1992).

3 Simulations

In this section, we demonstrate with simulations the efficacy of our neural network model in learning how to actively learn. We tackle several learning tasks, ranging from simple (the continuous-XOR Problem) to moderate (the Two-Spirals

Problem) to quite demanding (recognizing high-dimensional images of hand-written digits).

To experimentally investigate optimal scheduling for the active learning phase (i.e., the phase in which the learner begins actively looking for informative examples to improve learning performance), we simulate three types of active learners: Early-Starter, Intermediate-Starter, and Late-Starter. As a learner, by definition, has no control over the information provided passively by the environment, and this passive flow of information can continue indefinitely, we assume that these three types of active learners are constantly engaged in passive learning; that is, they are constantly engaged in improving their learning performance using the information that is passively, yet constantly, provided by the environment. The Early-Starter begins the active learning phase right at the start, together with the passive learning phase. The Intermediate-Starter begins the active learning phase with some delay, at an intermediate stage of passive learning (i.e., when the learner has already acquired some knowledge of the learning task of interest). Finally, the Late-Starter does not begin the active learning phase until a very late stage of passive learning (i.e., when the learner has nearly mastered the learning task at hand). As such, the Early-, Intermediate-, and Late-Starters are constantly engaged in passive learning (using the information passively provided by the environment) even *during* their active learning phase—they only differ in terms of when their active learning phase begins.

Although being simultaneously engaged in both passive and active learning (as our three Early-Starter, Intermediate-Starter, and Late-Starter learners are) is a more psychologically plausible assumption—compared to having learners who either only perform pure active learning or pure passive learning—the foregoing three learners, due to benefiting from different amounts of information, do not provide a fair characterization of the potential boost in learning accuracy afforded by active vs. passive learning.

To provide a completely fair comparison between active and passive modes of learning, and, furthermore, to theoretically corroborate several experimental findings on the efficacy of active learning, in Sec. 3.3 we simulate two new learners (the Active-Passive (AP) learner and Passive-Active (PA) learner), allowing us to directly investigate how active learning fares against passive learning.

3.1 Continuous-XOR Problem

As our first learning task, in this subsection we consider the continuous-XOR classification problem (see Fig. 1(a)). For the passive learning phase, the training set consists of 1000 samples, generated uniformly at random, in the input square $[0,1]^2$, paired with their corresponding labels. The learner receives these training samples in the form of batches of size 32. We implement the C-Net module by a 3-layer perceptron neural network (Rumelhart, Hinton, & Williams, 1985).

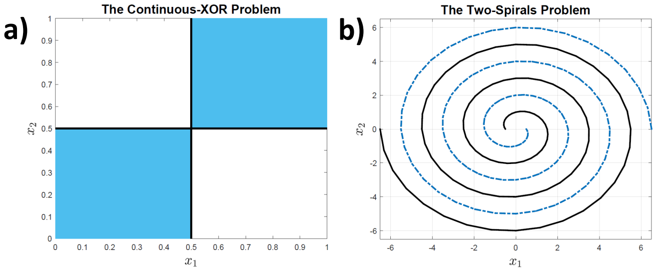


Figure 1: **(a)** The continuous-XOR learning task. the two blue quadrants correspond to the positive category and the two white quadrants correspond to the negative category, with the two solid black lines indicating the boundaries of the two categories. **(b)** The two-spirals learning task. The solid black spiral corresponds to the negative category and the dashed blue spiral corresponds to the positive category.

To quantitatively evaluate the efficacy of our model in learning to actively learn, we simulate the Early-,

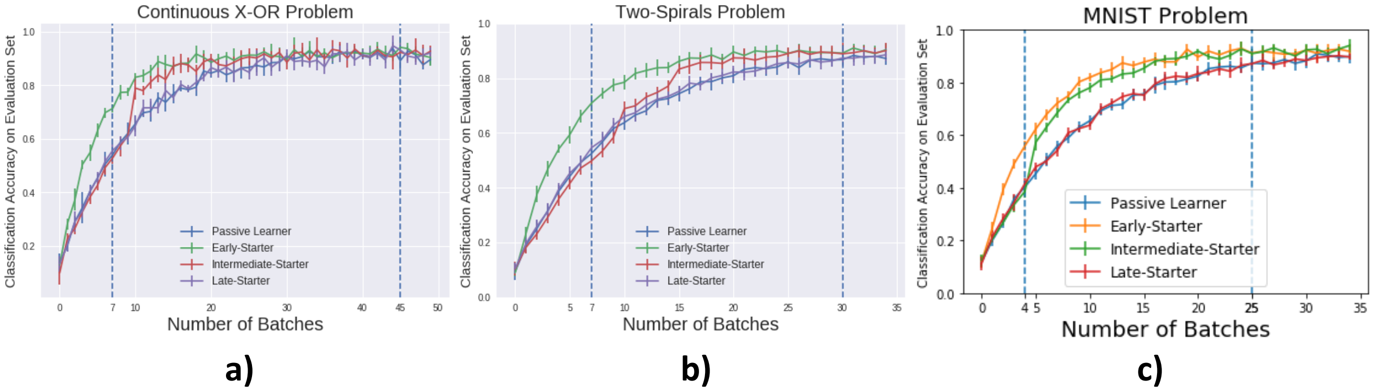


Figure 2: Classification accuracy on a held-out evaluation set by the Early-Starter, Intermediate-Starter, Late-Starter, and a purely passive learner. In each subfigure, the leftmost and the rightmost vertical dashed lines indicate the onset of the active learning phase for the Intermediate-Starter and Later-Starter, respectively. Error bars indicate ± 1 SEM. **(a)** The continuous-XOR problem. **(b)** The two-spirals problem. **(c)** The MNIST hand-written digits recognition task.

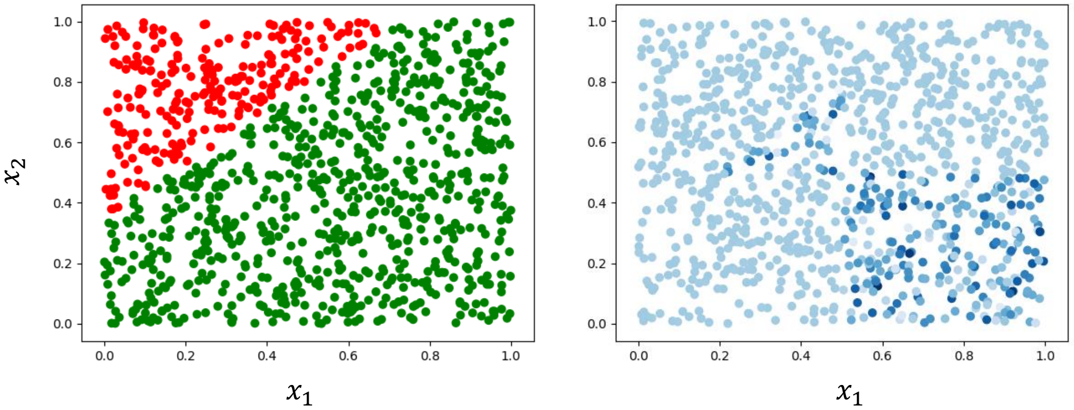


Figure 3: Left: An intermediate learning stage of the Intermediate-Starter learner in the continuous-XOR task. Red and green dots indicate examples that the learner classifies as negative and positive patterns, respectively. Right: The guidance provided by the Q-Net module at the stage of learning indicated in the Left subfigure. By assigning higher affinity scores (indicated by darker blue dots) to those regions of the input space about which the knowledge of the C-Net is lacking/incorrect, the Q-Net ensures that, by actively selecting those darker blue dots, the learning performance of the C-Net module likely improves.

Intermediate-, and Late-Starter learners, and compare their learning accuracy against a purely passive learner (as a baseline condition); see Fig. 2(a). As a measure of learning accuracy, we report percent of correct classification on a held-out evaluation set of size 100. The evaluation set comprises 100 samples, selected uniformly at random from the input square $[0,1]^2$. Note that the training and evaluation sets do not overlap—their intersection is an empty set.

As Fig.2(a) shows, the Early-Starter predominantly obtains the highest learning accuracy; this performance is later matched by the Intermediate-Starter when it begins its active learning phase. Fig. 2(a) also suggests that any form of active learner (Early-, Intermediate, or Late-Starter) generally outperforms, in learning accuracy, a purely passive learner.

Next, we provide intuition into how the Q-Net module helps the C-Net improves its classification accuracy, by actively guiding the C-Net module toward those input regions the knowledge of which likely maximally improves the learner's classification accuracy. Fig. 3(left) depicts an intermediate learning stage of the Intermediate-Starter learner. As Fig. 3(left) shows, our classifier, i.e., the C-Net module, has already learned some knowledge about the task (that, the top-left quadrant likely corresponds to the negative patterns), but its knowledge about the decision boundaries is still lacking. Fig.3(right) shows the guidance provided by the Q-Net module at this stage of learning: By assigning higher affinity scores (indicated by darker blue dots) to those regions of the input space about which the knowledge of the C-Net is lacking/incorrect, the Q-Net ensures that, by actively selecting those darker blue dots, the learning performance of the C-Net module improves.

3.2 Two-Spirals Problem

As our second learning task, in this subsection we consider the famously difficult Two-Spirals classification problem (see Fig. 1(b)). For the passive learning phase, the training set consists of 2000 samples (1000 samples per spiral), selected uniformly at random, on the two input spirals. The learner receives these training samples in the form of batches of size 32. As was the case in the previous subsection, we implement the C-Net module by a 3-layer perceptron neural network (Rumelhart, Hinton, & Williams, 1985).

To quantitatively evaluate the efficacy of our model in learning to actively learn, we simulate the Early-, Intermediate-, and Late-Starter learners, and compare their learning accuracy against a purely passive learner (as a baseline condition); see Fig. 2(b). As a measure of learning accuracy, we report percent of correct classification on a held-out evaluation set of size 100. The evaluation set comprises 100 samples, selected uniformly at random on the two input spirals. Note that the training and evaluation sets do not overlap—their intersection is an empty set.

As Fig.2(b) shows, the Early-Starter predominantly obtains the highest learning accuracy; this performance is later matched by the Intermediate-Starter when it begins its active learning phase. Fig. 2(b) also suggests that any form of active learner (Early-, Intermediate, or Late-Starter) generally outperforms in learning accuracy a purely passive learner.

3.3 Hand-written Digits Recognition Task

As our last (and hardest) learning task, in this subsection we consider the problem of recognizing high-dimensional images of hand-written digits, using the MNIST dataset, a popular dataset in the deep learning community (Fig. 4). For the passive learning phase, the training set consists of $60,000$ examples of 28×28-pixel hand-written digits. The learner re-

Figure 4: Hand-written digit examples from the widely used MNIST dataset.

ceives these training samples in the form of batches of size 32. We implement the C-Net module by a 6-layer convolutional neural network (LeCun & Bengio, 1995).

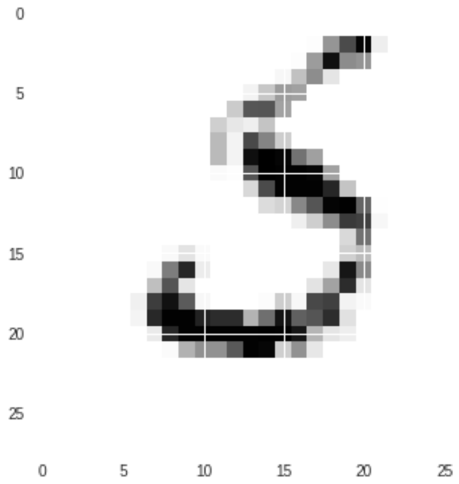


Figure 5: A 28×28-pixel digit actively selected by our model to improve learning performance. More precisely, the Q-Net believes that the classification accuracy of the C-Net can be improved by informing the C-Net that the shown 28×28-pixel image (as a whole) is a 5. Numbers on the vertical and horizontal axes indicate pixel number.

Fig. 5 shows an example produced in the active learning phase of our model; our model believes that, at this stage of learning, informing the C-Net about this example (i.e., that this 28×28-pixel image, as a whole, belongs to the class of Digit 5) significantly boosts the classification accuracy of the C-Net module. To visualize the example depicted in Fig. 5, we used a decoder neural-network module, allowing us to map the corresponding representation from the psychological space into the original 28×28-dimensional space of hand-written digits.

To quantitatively evaluate the efficacy of our model in learning to actively learn, we simulate the Early-, Intermediate-, and Late-Starter learners, and compare their learning accuracy against a purely passive learner (as a baseline condition); see Fig. 2(c). As a measure of learning accuracy, we report percent of correct classification on a held-out evaluation set of size 1000. The evaluation set comprises 1000 samples, selected uniformly at random from the original MNIST test set of size $10,000$. Note that the training and evaluation sets do not overlap.

As Fig.2(c) shows, the Early-Starter predominantly obtains the highest learning accuracy; this performance is later matched by the Intermediate-Starter when it begins its active learning phase. Fig. 2(c) also suggests that any form of active learner (Early-, Intermediate, or Late-Starter) generally outperforms a purely passive learner in learning accuracy.

Recently, MacDonald and Frank (2016) showed that passive-first learning yields better learning performance compared to active-first learning. More specifically, they showed that a passive learning phased followed by an active learning phase yields better ultimate learning performance, compared to the reversed order. As our three Early-, Intermediate-, and Late-Starter learners are constantly engaged in passive learning, even *during* their active learning phase, we cannot directly investigate the key question of which sequence of passive/active learning would ultimately yield better learning performance.

Next, we directly test the effect of passive/active learning sequence on learning. To this end, a la MacDonald and Frank (2016), we simulate two new types of learners: Passive-Active (PA) and Active-Passive (AP). PA performs passive learning during the first stage of his learning and then switches into a purely active learning phase (wherein PA only considers the samples recommended by the Q-Net module). Conversely, AP performs purely active learning during the first stage of his learning and then switches into a passive learning phase.

Fig. 6 clearly shows the superiority of PA, in learning accuracy, over AP. This finding theoretically corroborates, and serves as the first computational account of, the experimental finding by MacDonald and Frank (2016) showing that prior passive learning improves subsequent active learning.

Additionally, our finding that, during the first block of learning (Fig. 6, on the left-hand side of the vertical dashed line), AP performs worse, in learning accuracy compared to PA, is supported by the recent experimental study by Markant and Gureckis (2014) revealing that the quality of active learning is sub-optimal early in learning.

4 General Discussion

Humans are not mere passive observers of their environment, but actively search for information which helps to improve their learning performance. Despite being a hallmark of human cognition, the computational underpinnings of this active (or self-directed) mode of learning have remained largely unexplored (Gureckis & Markant, 2012).

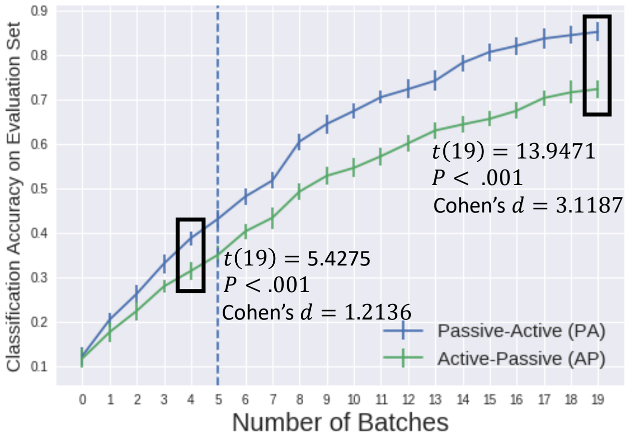


Figure 6: Investigating the effect of passive/active learning sequence on learning. Passive-Active (PA) performs passive learning first and then switches to active learning. Conversely, Active-Passive (AP) performs active learning first and then switches to passive learning. The vertical dashed line indicates the onset of the transition from one mode of learning to the other. Error bars indicate ± 1 SEM.

Building on recent advances in machine learning, particularly deep reinforcement learning, we present a novel neural-network model simulating the process of learning how to actively learn. Importantly, our neural-network model starts from scratch, having no a priori knowledge of the learning task, nor having any preset active learning heuristic(s) to choose from or to follow. To the contrary, by conceptualizing the problem as a reinforcement learning task, our neural-network model learns, during the passive phase of learning, an effective active learning strategy allowing for faster learning. Extensive simulations demonstrate the efficacy of our model, particularly in handling the high-dimensional learning task of MNIST hand-written digits.

Additionally, our model serves as the first computational account of the recent experimental finding by MacDonald and Frank (2016) showing that prior passive learning improves subsequent active learning, and provides a mechanistic explanation of why the quality of active learning is sub-optimal early in learning, as experimentally demonstrated by Markant and Gureckis (2014).

Markant and Gureckis (2014) also showed that passive learners did not benefit from being “yoked" to active learners' data. Future work should investigate whether our model can also account for this finding.

There is a growing consensus in the artificial intelligence and cognitive science communities that the two fields should establish stronger ties, much like at the dawn of the two fields. Several articles have recently called for bringing the fields of artificial intelligence, cognitive science, and neuroscience closer together (Hassabis et al., 2017, Gershman et al., 2015). Pursuing this approach, our work, like the work of many before us, attests to the effectiveness of this idea by exemplifying how a synergistic interaction between machine learning and cognitive science helps develop effective, human-like artificial intelligence.

Acknowledgments: This works was supported by an operating grant to TRS from Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Bonawitz, E., Denison, S., Gopnik, A., & Griffiths, T. L. (2014a). Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cognitive Psychology*, *74*, 35–65.

Bonawitz, E., Denison, S., Griffiths, T. L., & Gopnik, A. (2014b). Probabilistic models, learning algorithms, and response variability: sampling in cognitive development. *Trends in Cognitive Sciences*, *18*(10), 497–500.

Bröder, A. (2003). Decision making with the" adaptive toolbox": influence of environmental structure, intelligence, and working memory load. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*(4), 611.

Bruner, J. S., Jolly, A., & Sylva, K. (1976). Play: Its role in development and evolution.

Cherney, I. D. (2008). The effects of active learning on students' memories for course content. *Active Learning In Higher Education*, *9*(2), 152–171.

Dasgupta, I., Schulz, E., & Gershman, S. J. (2017). Where do hypotheses come from? *Cognitive Psychology*, *96*, 1–25.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gershman, S. J., Vul, E., & Tenenbaum, J. B. (2012). Multistability and perceptual inference. *Neural Computation*, *24*(1), 1–24.

Gureckis, T. M., & Markant, D. B. (2012). Self-directed learning: A cognitive and computational perspective. *Perspectives on Psychological Science*, *7*(5), 464–481.

Hanneke, S. (2014). *Theory of Active Learning* (Tech. Rep.). Available: http://www.stevehanneke.com/.

Hanneke, S. (2016). The optimal sample complexity of pac learning. *The Journal of Machine Learning Research*, *17*(1), 1319–1333.

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, *95*(2), 245–258.

Hoffart, J. C., Rieskamp, J., & Dutilh, G. (2018). How environmental regularities affect people's information search in probability judgments from experience. *Journal of Experimental Psychology. Learning, Memory, and Cognition.*

Káli, S., & Dayan, P. (2004). Off-line replay maintains declarative memories in a model of hippocampal-neocortical interactions. *Nature Neuroscience*, *7*(3), 286.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lengyel, M., & Dayan, P. (2008). Hippocampal contributions to control: the third way. In *Advances in neural information processing systems* (pp. 889–896).

Lieder, F., & Griffiths, T. L. (2017). Strategy selection as rational metareasoning. *Psychological review*, *124*(6), 762.

MacDonald, K., & Frank, M. (2016). When does passive learning improve the effectiveness of active learning? In *Proceedings of the* 38*th* *Annual Meeting of the Cognitive Science Society. Austin, TX: Cognitive Science Society.*

Markant, D. B., & Gureckis, T. M. (2014). Is it better to select or to receive? learning via active and passive hypothesis testing. *Journal of Experimental Psychology: General*, *143*(1), 94.

Michael, J. (2006). Where's the evidence that active learning works? *Advances in Physiology Education*, *30*(4), 159–167.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Momennejad, I., Otto, A. R., Daw, N. D., & Norman, K. A. (2018). Offline replay supports planning in human reinforcement learning. *eLife*, *7*, e32548.

Moreno-Bote, R., Knill, D. C., & Pouget, A. (2011). Bayesian sampling in visual perception. *Proceedings of the National Academy of Sciences*, *108*(30), 12491–12496.

Nobandegani, A. S., & Shultz, T. R. (2017). Converting cascade-correlation neural nets into probabilistic generative models. In *Proceedings of the* 39*th* *Annual Meeting of the Cognitive Science Society. Austin, TX: Cognitive Science Society.*

Nobandegani, A. S., & Shultz, T. R. (2018). Example generation under constraints using cascade correlation neural nets. In *Proceedings of the* 40*th* *Annual Meeting of the Cognitive Science Society. Austin, TX: Cognitive Science Society.*

Ólafsdóttir, H. F., Bush, D., & Barry, C. (2018). The role of hippocampal replay in memory and planning. *Current Biology*, *28*(1), R37–R50.

Pachur, T., Todd, P. M., Gigerenzer, G., Schooler, L., & Goldstein, D. G. (2011). The recognition heuristic: A review of theory and tests. *Frontiers in Psychology*, *2*, 147.

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1988). Adaptive strategy selection in decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*(3), 534.

Rieskamp, J., & Otto, P. E. (2006). Ssl: a theory of how people learn to select strategies. *Journal of Experimental Psychology: General*, *135*(2), 207.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (Tech. Rep.). California Univ San Diego La Jolla Inst for Cognitive Science.

Sanborn, A. N., & Chater, N. (2016). Bayesian brains without probabilities. *Trends in Cognitive Sciences*, *20*(12), 883–893.

Savin, C., & Deneve, S. (2014). Spatio-temporal representations of uncertainty in spiking neural networks. In *Advances in Neural Information Processing Systems*.

Watkins, C., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292.