A Computational Model of Functional Category Learning in a Cognitive Architecture

Yongjia Wang (yongjiaw@umich.edu)
John E. Laird (laird@umich.edu)
Computer Science and Engineering, EECS Department
University of Michigan, Ann Arbor, MI 48109 USA

Abstract

Categorization of objects is an important cognitive capability for human and higher animals. Phenomena related to category learning have been investigated both in human subjects and in animal behavior studies. However, it is less well understood in the computational processes that are responsible for the emergence of functionally meaningful categorizations from specific learning contexts. Here we present a unique computational model integrating object categorization and reinforcement learning (RL) in the Soar cognitive architecture. Our model simultaneously captures how object categorization affects behavior adaptation, and how behavioral adaptation influences object categorization over time in a specific functional context. Results from synthetic data demonstrate that our model successfully improves the speed of RL via categorization. The qualitative predictions from our model are consistent with existing theories of category learning.

Keywords: cognitive architecture; category learning; reinforcement learning; behavioral adaptation

Introduction

Category learning has been actively studied in higher animals including human (Ashby & Maddox 2005) and primates (Smith 2010). Categorization enables an individual to response to a novel stimulus, which resembles some other stimuli with known responses.

In this paper, we model several related phenomena in human category learning. The most important one is related to the notion of basic-level category as described by Rosch (1978). Consider the following two examples of abstraction hierarchies: furniture-chair-rocker and vehicle-car-sedan. The middle categories, chair and car, are basic categories, because they dominate both their subordinate and superordinate categories in terms of how fast they can be retrieved when a person is asked to describe the object without being put in a specific context. The original theory about basic-level categories was mainly concerned with this 'uniformity' aspect of category recognition across different individuals. On the other hand, there are also variations. First, non-basic level categories are frequently chosen in specific task contexts. Second, basic-levels are dependent on long term learning experience and can be significantly different across individuals in specific domains. All of these are characteristics of category learning. However, there has been a lack of computational models that coherently explain the combination of basic-level effects, context effects, and long-term learning effects in a specific functional setting, where a cognitive agent has to interact with the world to achieve some goals.

We present a unique computational model of category learning that integrates a hierarchical perceptual category learning component and a reinforcement learning component in the Soar cognitive architecture (Laird 2008). In our model, the underlying computation mechanism improves the agent's behavioral adaptation through category learning and at the same time results in the emergence of functionally meaningful categorizations as a result of feedback from reinforcement learning. We term our model a *functional category learning model*.

Our functional category learning model relies on perceptual category learning, and has the following features. First, functional categorization requires additional functional properties as input that are non-perceptual. For example, a venomous snake is in a different functional category from a harmless snake, but they may look very similar and fall under the same perceptual category of snakes. Second, functional categories are by definition specific to a particular functional context. For example, categorizations of animals as sources of food versus as pets are very different. Third, functional categories are directly related to decision making and are adaptive relative to the agent's experience. For example, a domain expert develops more detailed categorizations than novices do. Our hypothesis is that basic-level categories are rooted in people's experience, and depending on how objects are used, the categories can be significantly different across cultures, even individuals within the same culture.

Our functional category learning model involves two components. One is a perceptual category learning system, which can automatically learn hierarchical category structures based on innate perceptual features. The other is a reinforcement learning system, which uses the perceptual categories as the representational basis and incrementally forms functionally meaningful categories based on their utility values.

Hierarchical Categorization

There is a long history of hierarchical models of category learning. Quillian (1968) proposed the semantic network model, which can represent categorical relationships among objects in a hierarchical structure. However, the semantic network model does not include a learning mechanism to build the structure. COBWEB (Fisher 1987) is an algorithm that can incrementally learn a hierarchical organization of categories. A previous version of the ICARUS cognitive architecture used a COBWEB-based system, called LABYRINTH for its declarative learning and memory (Langley *et al.* 1991). Ambros-Ingerson *et al.* (1990)

described a neurologically inspired hierarchical clustering algorithm, which operates in a way very different from COBWEB and Granger (2005) has demonstrated the plausibility of using such hierarchical clustering algorithm as a principled computational instruction for human cognition.

Reinforcement Learning

Hierarchical category learning provides the necessary representational basis, however the representation itself is insufficient for functional category learning because it has no direct connection to how the learned knowledge can be used. Another learning process is required to connect the category representations with the agent's intrinsic functional meanings. We consider reinforcement learning (RL, Sutton & Barto 1998) as a candidate mechanism to establish such connections via incremental trial-and-error learning with feedback.

RL has been successfully applied in adaptively learning optimal control policies in the field of machine learning. The general model of RL has also been considered as a mechanism for human skill learning (Fu & Anderson 2006). Cognitive architectures such as Soar (Laird 2008) and ACT-R (Anderson *et al.* 2004) both have a reinforcement learning mechanism. However, there has not been a computational model integrating category learning and RL in these cognitive architectures.

Demonstration Task

We briefly describe our demonstration task before describing the implementation of our model, so that we can illustrate how the model works using a concrete example.

The demonstration task models a hunting scenario where the agent is presented with pairs of prey and hunting tools. There are diverse types of prey and tools, and different tools have different effectiveness on different prey. For example, a slingshot is good for small birds, but it will not work for larger prey. We assume that the agent does not have prior knowledge to predict the outcomes based on perceptual features of the objects. The agent must incrementally acquire such connections based on its experience through RL. During interaction with the world, the agent receives a positive reward if hunting is successful and a negative reward if it is unsuccessful. In order to learn faster, the agent needs to generalize its predictions based on perceptual similarity. For example, if the agent has learned that a bow is good for hunting rabbits, then it is likely to work against a woodchuck as well. Meanwhile, to improve generality, the agent must adapt its learning to the right level of abstraction through the course of using RL.

Model Implementation

Overview

Our model is implemented by combining a hierarchical category learning (HCL) system with Soar-RL (Nason & Laird 2005), which has been shown to successfully model animal behavioral data (Wang & Laird 2007). Our model uses the HCL component to perform perceptual learning.

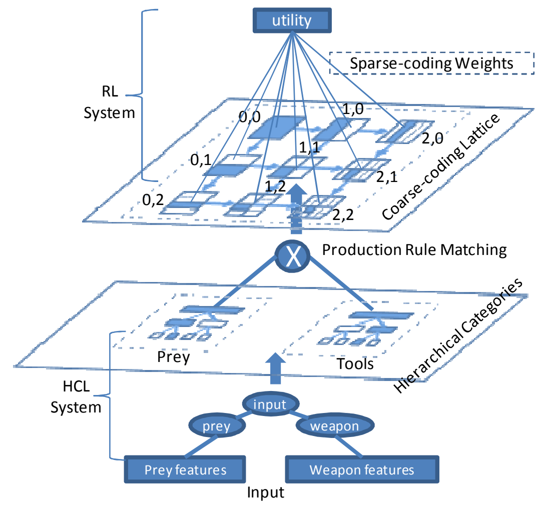

Figure 1: Overall structure of the system viewed as a two-layer network

The output of the HCL system is the input to the RL system. We have experimented with both COBWEB and a biologically inspired hierarchical clustering algorithm (Ambros-Ingerson *et al.* 1990). In general, any incremental hierarchical clustering system will be compatible with our model.

Learning Algorithm

In a functional category learning model, the functional utilities of objects are associated with specific actions, and can be naturally represented as value functions in the RL system. Soar-RL encodes the value function as a set of production rules, with an expressive syntax equivalent to first-order logic. The left-hand side of a rule tests state and action features, while the right-hand side generates the expected value for the matching state action pair. The expected value of an action is the sum of the values of all rules matching the current state and that action. The Soar-RL model is a special instance of the sparse-coarse coding approach to value function approximation (Sutton 1996).

In our functional category learning model, instead of using raw perceptual features of the objects in the state representation, the RL system uses the symbolic category representations from the HCL system. The entire structure of our model can be viewed as a two-layer network as shown in Figure 1. The bottom layer represents the HCL system. In this paper, we assume such hierarchical structure has been learned by the agent through regular perceptual category learning before the hunting task. And we investigate the emerging properties of doing reinforcement learning with such hierarchical categorization. The dark colored nodes in the hierarchies represent symbolic categories matching with the input objects. These symbolic categories are used in the state representation and are matched by rules in the RL system. Rules are represented as cells in the coarse-coding layer. A rule testing general category symbols will be coarser than a rule testing more specific category symbols. Dark colored cells represent rules that match the current state. The numbers on each grid indicates the hierarchy levels for component hierarchies, which will be explained later. The grids form an emerging lattice structure, with the transitive relationship *coarser-than*, represented by the arrows.

We formally describe the general algorithm below. To learn the target value function of a state action pair, the system first maps the input objects into a vector of functional roles R, which represents the argument types of the target function. The vector O represents the input objects binding with R:

$$R = (r_1, r_2, \dots, r_n)$$
$$O = (o_1, o_2, \dots, o_n)$$

In the example, the function is to predict the utility of hunting some prey with some tool, and for a particular instance, the inputs are two objects: rabbit and bow. According to our notation, input to the system will look like R=(prey, tool), O=(rabbit, bow). After matching objects with functional roles, the HCL system incrementally builds a set of hierarchies H correspondingly:

$$H = (h_1, h_2, \dots, h_n)$$

Let *height*(h_i) denote in the height of the hierarchy h_i, and k_i denote a cluster/category/node within the hierarchy. Let *level*(k_i) denote the level of cluster k_i in hierarchy h_i, with the root level being 0. Cells, grids and their relations, shown in Figure 1, are defined as following:

$$Cells = \{C_K, K = (k_1, k_2, \dots, k_n) | k_i \in clusters\ in\ h_i\}$$
$$Grids = \{G_L, L = (l_1, l_2, \dots, l_n) | 0 \le l_i \le height(h_i)\}$$
$$C_K\ belongs\ to\ G_L \equiv \forall i \in [1, n], level(k_i) = l_i$$
$$G_L \prec G_M \equiv \forall i \in [1, n], l_i \le m_i$$

More intuitively, each cell represents a rule in our RL system. A set of cells are composed into a grid that partitions the state space at a specific level of resolution. There is an emerging lattice structure among the grids with the transitive relation *coarser-than* (≺). For a given object o_i, the activation of a cluster k_i is denoted as $a(k_i)$:

$$a(k_i) = \begin{cases} 1\ if\ o_i \in k_i \\ 0\ if\ o_i \notin k_i \end{cases}$$

The mapping from o_i to k_i is achieved via category recognition in the HCL system, and only a single path of clusters are activated for a particular input as shown in Figure 1. Details of the COBWEB algorithm can be found in Fisher (1987). $a(k_i)$=1 means object o_i in the current state, bound to the corresponding functional role r_i, is an instance of the category represented by the cluster k_i. The activation of a cell, $a(C_K)$, is defined as:

$$a(C_K) = \prod_{i=1}^{n} a(k_i)$$

$a(C_K)$=1 only when all the objects match with the rule, which will fire to participate in predicting and learning the target value. The weight, $w(C_K)$, from the cell to the output unit is represented as a numeric value associated with the rule in the RL system. The learning algorithm updates the weights according to the *delta rule* for the identity activation function used in our RL system, where y is the predicted value and o is the target output value (current reward + discounted future rewards). The learning rate α for a specific rule C_K is chosen to decay over time t, where t is represented by the times the rule has been trained:

$$y = \sum_{C_K} (w(C_K) \times a(C_K))$$
$$\Delta w(C_K) = \alpha(C_K, t) \times (o - y) \times a(C_K)$$

The connection between the coarse-coding layer and the output unit is always sparse, since, for any input, only one cell from each grid in the lattice has non-zero activation. This is due to the competitive learning nature of the hierarchical clustering layer – only one cluster is activated at each level.

Simulation and Results

We use a hunting task as described earlier with synthetic data to evaluate our model. The data used in the task is shown in Figure 2. The hierarchies represent natural perceptual categories based on unsupervised learning with perceptual features, which are outputs from the HCL system as shown in Figure 1. We assume the agent has innate feature detectors that result in such perceptual categorization purely based on observing the objects without any hunting experience with the objects.

The functional interaction structure in this domain is represented in the two-dimensional table in Figure 2. A dark cell means the corresponding tool is good for hunting the prey and the agent will receive a reward of +1 if it chooses the action 'hunt'. The white cell means the corresponding tool is bad for hunting the prey and the agent will receive a reward of -1 if it chooses the action 'hunt'. The agent can alternatively choose the default action 'avoid', which will always give a 0 reward. We expect that the hierarchical categorizations will help the agent generalize its experience from a specific instance to similar combinations of objects. For example, the experience of hunting a rabbit with a longbow can be successfully generalized to hunting all four-legged animals with bow expect for one situation (longbow is not strong enough for hunting deer), so that both the category of Four-leg and Bow are useful abstractions. On the other hand, since there are variations within the group,

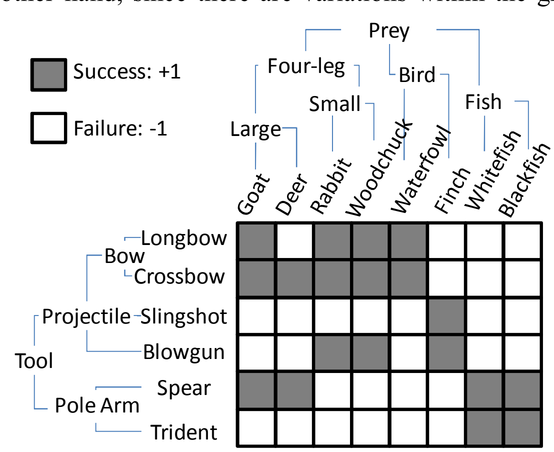

Figure 2: Input Data – Perceptual Category Hierarchies and Interaction Outcome Table

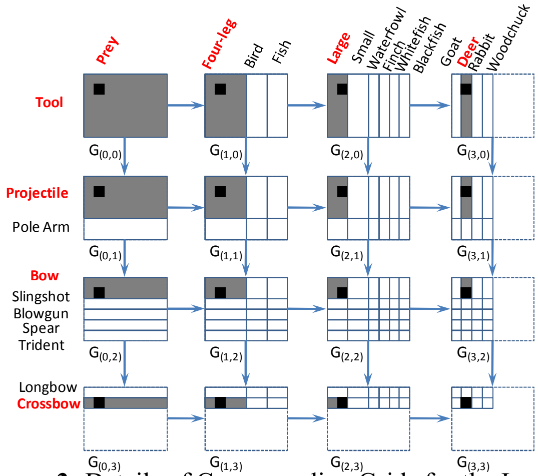

Figure 3: Details of Coarse-coding Grids for the Input (Deer, Crossbow)

we expect both concepts will be dominated by their subordinate categories in certain situations. In addition to trying to be close to reality, we designed the data so that it is complex, while at the same time, it has structure that tests specific aspects of the system, and it is simple enough to interpret the results.

To emphasize that the initial categorizations are based on innate perceptual features as opposed to taxonomic features, we use the labels such as Four-leg, Large, and Small, instead of Mammal, Ungulate, and Rodent to indicate they are perceptual categories. Birds have feathers, sharp beaks, and can fly. Fish all have similar shape, scales and swim in the water. A hierarchical clustering algorithm such as COBWB can automatically discover such statistical correlations among high dimensional perceptual features and incrementally build up a hierarchical structure as shown in Figure 2. Since we focus on the interaction between hierarchical categorization with RL, we did not include a detailed perceptual learning step in our simulation.

The effectiveness of tools with regard to prey may appear obvious to the reader. We make the assumption that the agent has no relevant prior knowledge to derive the effectiveness of a tool based on perceptual features. It has to incrementally learn the effectiveness of a tool for a prey through experience and build up the connections from perceptual similarities to functional outcomes piece by piece via the RL mechanism.

Figure 3 shows the details in the layer of coarse-coding rules for a specific input: hunting a deer with a crossbow. The black dots spatially represent the specific input in different grids. The gray areas represent the generalization effects when the more general rules fire. In this case, the agent receives a reward of +1 and each of the 16 rules participates in prediction and updating. Since a general rule (a larger cell) receives more training samples than a more specific rule (a smaller cell), it converges to the target value faster. On the other hand, the smaller cell will tend to compensate for the value in the context of the larger cell. The region with the dotted border in 7 of the grids on the lower and right borders means there are no more specific rules generated for those regions because it has already reached the leaf level of the categorization hierarchies.

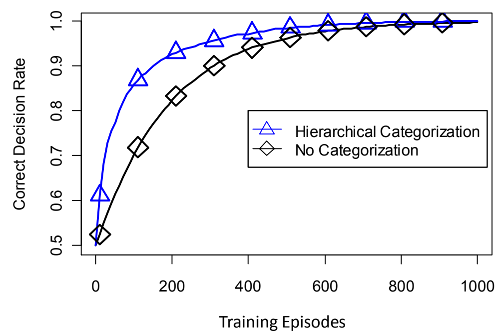

Figure 4: Learning with and without Hierarchical Categorization

Result 1: Category Learning to RL

Figure 4 compares the learning performance of hierarchical categorization with a baseline that uses the leaf level nodes without generalization. In the training data, there are two instances under each of the leaf nodes shown in Figure 2. For example, there are two instances of *Goat* that look different but have the same functional properties. Therefore, the size of the input space is: 16 (prey) times 12 (tools) equals 192. We evaluate the performance improvement during the course of learning. The agent is trained with random samples from the input space with replacement. The learning rate is set at 0.1. For a given amount of training episodes, we evaluate the rates of correct decisions it makes if it follows the policy derived from the current value function. The result shows that the model successfully integrates hierarchical categorization to speed RL.

Result 2: RL to Category Learning

Next, we analyze how functionally meaningful categorizations emerge from the process of RL. For a given input, there are multiple rules firing simultaneously, each coming from a different grid as shown in Figure 3. We define the *dominant rule* as the rule with the highest absolute value, or equivalently the *winning cell* with largest magnitude in its weight:

$$WinningCell = ArgMax_C\{\|w(C)\|\}$$

Correspondingly, we define the *dominant categories* as the categories associated with the dominant rule. In the hunting task for a specific input, there will be a dominant category for prey and a dominant category for tool. For example, the rule testing *Fish* and *Pole Arm* (the lower-right dark square consisting of 4 cells) dominates all the more specific rules that involve subtypes of *Fish* or subtypes of *Pole Arms* because it receives more training samples. It also dominates more general rules because there are inconsistent updates for those rules that cancel out each other. Consequently, the categories for *Fish* and *Pole Arm* are the dominant categories in these particular situations. The general principle is that a rule simultaneously maximizing both generality and consistency will dominate other rules. Intuitively, the associated dominant categories are more

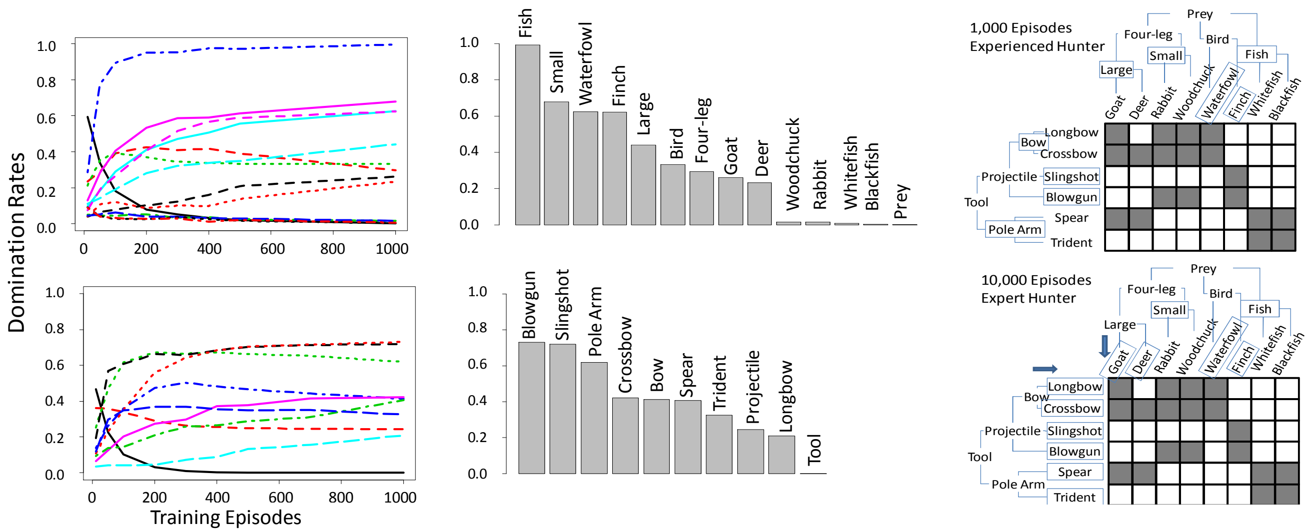

Figure 5: Dynamics of Domination Rates up-to 1,000 Episodes (on the left),
Domination Rates at 1,000 Episodes (in the middle),
Push-down of Basic-level Categories (boxed) with more Training Episodes – 1,000 vs. 10,000 (on the right)

functionally salient than their superordinate and subordinate categories, since they are the sources contributing to most of the decisions made by the voting mechanism. We use the overall *domination rates* across all possible inputs to measure the functional saliency of a category in a more context-free manner, which indicate how likely a category will become a basic-level category when there is no context effect.

The left side of Figure 5 shows the dynamics of domination rates up to 1,000 training episodes for all the categories of prey and tools. The trend is that the more general categories initially have higher domination rates because they cover more inputs and are trained with higher frequencies. As more experience is gained, consistent categories under a less consistent parent category have increasing domination rates (such as the two subtypes of birds), while less consistent superordinate categories become less dominant (such as the general category *Prey*, *Four-legged* animal, and *Bird*). On the other hand, a perceptual category that does not have any functional differences from other members under the same superordinate category does not arise as a functionally salient category (such as *Rabbit*, *Woodchuck* and the two subtypes of *Fish*). The middle of Figure 5 shows the domination rates after 1,000 training episodes. Since the ordering of inputs causes variations in the value of rules, we measure the mean domination rates across 300 independent learning trials, and the estimated standard errors for the means (not shown in the figure) are all less than 0.01. For example, the category for *Small Four-legged* animal dominates its superordinate and subordinate categories (including *Prey*, *Four-legged* animal, *Rabbit* and *Woodchuck*) in about 68% of all possible inputs. The category of *Rabbit* rarely dominates because its superordinate category completely captures the decision boundaries.

The right side of Figure 5 shows the context-free basic-level categories in boxes, which are the dominating categories along a path. The top figure shows the situation at 1,000 training episodes (for an experienced hunter) and the bottom figure at 10,000 episodes (for an expert hunter). The additional training experiences can "pull down" the basic-level towards more specific categories (indicated by the arrows). This effect arises naturally in our model and corresponds to the fact that a human domain expert possesses more specific basic-level vocabularies than a less experienced person.

Discussion

The general definition of category learning is the process that groups similar stimuli together so that similar responses can be made. Traditional cognitive theories of category learning include two competing views: the prototype view (Rosch 1973) and the exemplar view (Medin & Schaffer 1978). The prototype view is based on the principle of cognitive economy (Rosch 1978) and is supported by the existence of linguistic representations of abstract categories. However, there has been a shift of favor from the prototype towards the exemplar view because exemplar models provide superior empirical results in a variety of experimental settings (Nosofsky & Zaki 2002). A practical concern about the prototype view is that a prototype may fail to retain sufficient discriminative information. More recent models reconcile the two extreme forms and rely on representations at multiple abstraction levels (Vanpaemel & Storm 2008, Love *et al.* 2004).

Our model is consistent with both the prototype and exemplar views. In addition, it explicitly models the learning process and can deal with the more challenging situations where the input states involve multiple objects (such as the interaction between prey and tools). In terms of decision making, our model is more like exemplar based models, where the agent acquires information about specific inputs, and then makes generalizations to novel inputs based on perceptual similarity. In terms of category abstraction, our model agrees with prototype models. In particular, it predicts a similar trend as in the phenomenon of basic-level category (Rosch 1978) where the most prominent categories (basic-level categories) reside in the middle of a categorization hierarchy.

Furthermore, our model predicts that category domination is context specific. For example, in the hunting context used as our demonstration task, *Pole Arm* is the dominant category if the sub-context is hunting *Fish* (all subtypes of *Pole Arms* are good for fishing). In a different context, however, *Spear* and *Trident* will dominate if the sub-context is hunting *Deer*. Our model explicitly supports the hypothesis that the "context-free" basic level categories, as described by Rosch, are the overall effects acquired across multiple functional contexts. Since the everyday activities related to common objects are largely the same across individuals, the context-free basic-level categories appear to be consistent as manifested in natural language.

Our model does not involve a dedicated process of selecting functional meaningful categories. Selection is achieved as an emerging by-product of the RL process. As a consequence, our model cannot explain certain types of category learning that rely on deliberate reasoning or higher degrees of abstractions, where the agent generalizes across instances that are perceptually distinctive but functionally similar. Such deliberate categorization is better described by rule based category learning model (Rouder & Ratcliff, 2006), or analogical reasoning processes such as in the structure-mapping engine (SME, Falkenhainer *et al.* 1989).

Conclusion

In this paper, we have presented the first computational model that integrates hierarchical category learning and RL in a general cognitive architecture, which can be used to coherently model basic-level effects, context effects and long-term learning effects in category learning. The unique feature of this model is that it simultaneously captures how categorization affects behavior adaptation, and how behavior adaptation influences categorization in a functional context. The general trends predicted by our model are consistent with existing category learning theories. Although the Soar-RL model has been successfully applied to match animal behavior data (Wang & Laird 2007), further empirical experiments are required to confirm its validity in our category learning model.

Acknowledgement

This research was supported in part by the Ground Robotics Reliability Center (GRRC) at the University of Michigan, with funding from government contract DoD-DoA W56H2V-04-2-0001 through the Joint Center for Robotics.

References

Ambros-Ingerson, J., Granger, R., & Lynch, G. (1990). Simulation of paleocortex performs hierarchical clustering. *Science*, *247*(4948), 1344-1348.

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

Ashby, E. G., & Maddox, W. T. (2005). Human category learning. *Annu. Rev. Psychol.*, *56*, 149-178.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine - algorithm and examples. *Artificial Intelligence*, *41*(1), 1-63.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*(2), 139-172.

Fu, W., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *J. Exp. Psychol. Gen., 135*, 184-206.

Granger, R. (2006). Engines of the brain: the computational instruction set of human cognition. *AI Mag.*, *27*(2), 15-32.

Laird, J. E. (2008). Extending the Soar cognitive architecture. In *Proceeding of the 2008 Conference on Artificial General Intelligence*.

Langley, P., McKusick, K. B., Allen, J. A., Iba, W. F., & Thompson, K. (1991). A design for the ICARUS architecture. *SIGART Bull.*, *2*(4), 104-109.

Love, B. C., Medin, D. L., & Gureckis, T. M. (2004). SUSTAIN: A network model of category learning. *Psychol. Rev.*, *111*(2), 309-332.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychol. Rev.*, *85*(3), 207-238.

Nason, S., & Laird, J. E. (2005). Soar-RL: integrating reinforcement learning with Soar. *Cognitive Systems Research*, *6*(1), 51-59.

Nosofsky, R. A., & Zaki, S. R. (2002). Exemplar and prototype models revisited: response strategies, selective attention, and stimulus generalization. *J. Exp. Psychol. Learning*, 28(5), 924-940.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, *12*(5), 410-430.

Rosch, E. (1978). Principles of categorization. In *Cognition and Categorization* (pp. 27-48). John Wiley & Sons Inc.

Rosch, E. (1973). Natural categories. *Cognitive Psychology*, *4*(3), 328-350.

Rouder, J. N., & Ratcliff, R. (2006). Comparing exemplar- and rule-based theories of categorization. *Current Directions in Psychological Science*, *15*(1), 9-13.

Smith, J. D., Chapman, W. P., & Redford, J. S. (2010). Stages of category learning in monkeys (Macaca mulatta) and humans (Homo sapiens). *J. Exp. Psychol. Anim. B*, *36*(1), 39-53.

Sutton, R., & Barto, A. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press.

Sutton, R. S. (1996). Generalization in reinforcement learning: successful examples using sparse coarse coding. *NIPS 8*, 1038-1044.

Vanpaemel, W., & Storms, G. (2008). In search of abstraction: The varying abstraction model of categorization. *Psychon. B. Rev.*, *15*(4), 732-749.

Wang, Y., & Laird, J. E. (2007). The importance of action history in decision making and reinforcement learning. In *Proceedings of the Eighth International Conference on Cognitive Modeling*. Ann Arbor, MI.