Comparison of Instance and Strategy Models in ACT-R

Cleotilde Gonzalez[1] (coty@cmu.edu)
Varun Dutt[1] (varundutt@cmu.edu)
Alice F. Healy[2] (alice.healy@colorado.edu)
Michael D. Young[2] (mdyoung@psych.colorado.edu)
Lyle E. Bourne, Jr.[2] (lyle.bourne@colorado.edu)
[1]Dynamic Decision Making Laboratory
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[2]Center for Research on Training
University of Colorado, Boulder, CO 80309-0345, USA

Abstract

This paper presents a comparison of two models, built on the same architecture, ACT-R, and on the same dynamic decision making task, RADAR. The two models represent the Strategy-Based Learning (SBL) approach and the Instance-Based Learning (IBL) approach. The SBL approach assumes a certain set of predefined strategies, and learning occurs by selecting the most successful strategy over time. The IBL approach proposes that decisions are made based on retrieval of good past experiences stored in memory. This approach assumes no previous initial experience apart from that gained while performing the task. Both models were tested with respect to two criteria: fit to human data during a training exercise with RADAR and adaptability to test conditions that are either similar to or different from the training conditions. Our comparison results demonstrate that both models fit learning human data successfully, but the IBL model is more robust than the SBL model. This exercise initiates a discussion of the SBL and IBL approaches to modeling choice and decision making in ACT-R and a reevaluation of how to compare and assess computational models.

Keywords: dynamic decision making; instance-based learning; strategy-based learning; consistent mapping; varied mapping; ACT-R.

Introduction

In cognitive psychology there have been at least two views of the world: that humans understand the world by means of rules and by particular domain-related events (Nisbett, 1993). In cognitive modeling these same two views are often reproduced in the behaviorism and connectionism debate (Anderson & Lebiere, 2003). The debate in the late 1980s led to an opposition between the two modeling approaches, in which connectionism was perceived to resemble the underlying neural structure better than did behaviorism, a focus on learning from environmental stimuli rather than from generic rules, and a focus on subsymbolic manipulations rather than symbolic representations. In reality the two approaches have more in common than what was recognized in this debate.

ACT-R is a hybrid architecture composed of both symbolic and subsymbolic aspects (Anderson & Lebiere, 1998, 2003). The symbolic aspects are declarative and procedural. The declarative knowledge is represented in chunks, and the procedural knowledge is represented in productions (if-then rules). The subsymbolic elements of ACT-R are the neural-like statistical and mathematical mechanisms that manipulate the symbolic representations.

ACT-R allows for two different approaches to modeling human behavior that are particularly relevant for decision making and learning: the Strategy-Based Learning (SBL) and the Instance-Based Learning (IBL) approaches.

The SBL approach is the most popular approach to modeling choice and decision making in ACT-R (Lovett, 1998). Under this approach, modelers determine the strategies by which humans perform a task, and they represent these strategies in the form of production rules. Choice among competing production rules is controlled by the ACT-R subsymbolic utility learning mechanisms. Each production has a utility value that represents the rule's probability of success and the costs involved in reaching the goal. The utility learning mechanism produces a gradual switch from less successful to more successful strategies over time.

The IBL approach, although less popular, has been used successfully in representing decision making, mostly in dynamic situations (Dutt & Gonzalez, 2008; Gonzalez, Lerch, & Lebiere, 2003). Under the IBL approach, modelers determine the representation of declarative knowledge (chunks) in a task and represent a generic decision making process in production rules. This approach has been the basis for the development of a theory of decision making in dynamic tasks, called Instance-Based Learning Theory of Dynamic Decision Making, which provides IBL models with a generic decision making process (Gonzalez et al., 2003).

The main learning in this approach occurs at the declarative rather than the procedural level, where actions are based on the storage and retrieval of similar chunks in and from memory. Selection among chunks is based on ACT-R's activation subsymbolic learning mechanisms. Each chunk has a value of activation determined by a number of factors including the recency and frequency of use of that chunk. For example, recency and frequency of usage of a chunk determine the base-level activation, which represents the probability that a chunk is needed. The activation is also modulated by the degree to which a chunk matches the retrieval cues, with chunks encoding similar situations to the current one receiving some activation.

Over time, an IBL model transitions from the use of a general heuristic to the use of instances, as determined by the number of instances stored and the similarity of the situations confronted in the task (Gonzalez et al., 2003).

This paper presents a comparison of two models, IBL and SBL models, both interacting with the same real-time decision making task, and both developed under the same architecture (ACT-R). This effort differs from other model comparison efforts in that other model comparisons are often done to evaluate different "architectures" and often aimed at determining the "winning" model (Anderson & Lebiere, 2003; Cassimatis, Bello, & Langley, 2008). By comparing two different modeling approaches that represent decision making behavior in the same task and in the same architecture, we highlight the real value of model comparison: to understand the processes by which behavior is represented, the constraints that the different approaches impose upon the task models, and the comparison of the theoretical assumptions of the two approaches (Lebiere, Gonzalez & Warwick, 2009). The models interacted in real-time with a dynamic decision making task called RADAR (Gonzalez & Thomas, 2008).

We compared the SBL and IBL models according to two different dimensions: (1) fit: how well each model fits human learning data in the task; and (2) adaptability: how well each model is able to reproduce the way humans having learned in one scenario of the task behave in a testing condition, in scenarios that are similar to or different from the training condition. The fit criterion is common in model comparisons, whereas the adaptability criterion is relatively new (Gluck, Bello, & Busemeyer, 2008). The adaptability criterion we use here is similar to the generalization criterion method (Busemeyer & Wang, 2000), which divides observed data into two sets: a calibration or training set to estimate model parameters and a validation or test set to determine predictive performance. However, we further test the adaptability of our models by examining the models' ability to adapt to test conditions that are either similar to or different from the training conditions.

Experiment on the RADAR Task

The task used for this modeling effort is a dynamic visual detection and decision making task that has been used in past research to study automaticity (Gonzalez & Thomas, 2008) and training principles (Young, Healy, Gonzalez, & Bourne, 2007). The task, called RADAR, is described in detail by Gonzalez and Thomas (2008), and thus here we only summarize the relevant elements.

The goal in RADAR is to detect and eliminate hostile enemy aircrafts by visually discriminating moving targets among moving distractors in a radar screen. RADAR is similar to military target visual detection devices, in which a moving target needs to be identified as a potential threat and a decision is made on how to best destroy the target. The task has two components: (a) visual and memory search and (b) decision making. The visual and memory search component requires the participant to memorize a set of targets and then look for the presence of one or more targets on a radar grid. A target threat may or may not be present among a set of moving blips that represent incoming aircraft. The blips—in the form of digits, consonants, or blank masks—begin at the four corners of the radar grid and approach the center at a uniform rate. The detection of an enemy aircraft must occur before the blips collapse in the middle of the grid. This is the main component used in the experiment described below. The decision-making component is not relevant for this human experiment.

General Experimental Methods

Forty-eight participants at the University of Colorado, Boulder were asked to interact with RADAR to respond as quickly as possible to target letters or digits occurring among distractor letters or digits. In addition to target detection, participants were required to count deviant tones (low and high frequency) among standard tones (medium frequency) that played in the background during the target detection task. The experiment consisted of a training session and a test session with a 1 week-delay between the two sessions. Half the participants trained with both the tone-counting task and the target detection task and half performed the target detection task in silence. At test, half resumed their training condition and half switched.

There were 8 blocks during training and 8 blocks during testing, each consisting of 160 total trials. A trial is a group of 7 frames (RADAR screen and individual attempt to detect a target). A memory set of 1 or 4 possible targets was shown to participants prior to starting a trial. At most 1 frame within each trial contained a target. Each frame included either 1 or 4 non-blank blips among which there could be one target and zero or more distractors in the 7 frames of a trial. Targets and distractors were consistently mapped (CM: a target in the memory set never appeared as a distractor within a block) or varied mapped (VM: a target in memory set could appear as a target in one trial and as a distractor in another trial of a block).

Half the participants saw digits as the targets on CM trials. For these participants, letters were the distractors on CM trials and were both the targets and distractors on VM trials. The remaining participants saw letters as the targets on CM trials. For them, digits were the distractors on CM trials and were both the targets and distractors on VM trials. There were 9 integers 1 to 9 and 9 consonants C, D, F, G, H, J, K, L, M used as targets or distractors.

The 160 trials were divided into two session halves, each with 4 blocks (i.e. 80 trials), separated by a 5-min break. Blocks varied by mapping and processing load (number of items in the memory set and number of blips in each trial) condition. The four blocks in each session half included one of each combination of mapping condition and processing load (CM 1+1, VM 1+1, CM 4+4, VM 4+4). For the first session half these conditions occurred in the order CM 1+1, CM 4+4, VM 1+1, VM 4+4. For the second session half these conditions occurred in the reverse order VM 4+4, VM 1+1, CM 4+4, CM 1+1. Thus, the average block position was the same for each condition across session halves.

We use correct detection time (in ms) as the dependent variable. Results are presented in a later section, where they are compared to the results from the IBL and SBL computational models.

Instance-Based Learning Model

The IBL model was based upon the Instance-Based Learning Theory (IBLT) and other IBL developments (Gonzalez et al., 2003). IBLT was originally developed as a way to explain and predict decision making in dynamic, complex tasks (Dutt & Gonzalez, 2008; Gonzalez et al., 2003). For the RADAR task an instance (referred to as a chunk in ACT-R) had the structure shown in Table 1.

Table 1: Structure of an Instance in RADAR

Slot Name	Description	Chunk
Blip-Situation	Value of Blip	Situation
Decision	Spacebar Press	Decision

The *Blip-Situation* slot corresponded to the blip value (letter or number) occurring on the RADAR screen in one of the north-west, north-east, south-west, or south-east locations, respectively at a time. In the case of 1+1 trials, three out of the four slot locations contained a NIL value. For the purpose of linear similarity calculations (discussed later), the nine consonants were numbered from 10 to 18. The *Decision* slot refers to the act of pressing or not the spacebar. Although typically instances have a *Utility* slot to categorize an experience as good or bad in a situation after the IBL model gets feedback, in this model, due to the task's trial structure and the trivial feedback, we did not use such a slot.

As per Gonzalez et al. (2003), the IBL starts with the recognition process in search for alternatives and the classification of the current situation as *typical* or *atypical*. A situation is typical if there are memories of similar situations (i.e., instances of previous trials that are similar enough to the current situation). If it is typical then the retrieved instance is used in judging the value of the decision to be made in the current situation. If the situation is atypical (i.e., no instance similar to the current conditions is found in memory), a judgment heuristic is applied (in the present case, the heuristic is "wait for next blip"). When a decision point comes into place at one of the four blip positions, NW, NE, SW, and SE, a choice has to be made whether to search for more alternatives or to execute the current best alternative. In the RADAR task, the choice is simply made by seeing if the retrieved instance is similar enough to the one of the current blip situations (in case nothing was retrieved or the instance that was retrieved did not equal the current blip situation, then a choice is made to wait for the next blip situation and not to press the spacebar key, i.e. by a "wait for next blip" judgment heuristic). Thus, if something was retrieved from declarative memory, then the decision is to press the spacebar only if the retrieved instance is exactly the same as the current blip situation.

Before the IBL process starts for each frame's blips in a trial, the IBL model notices a set of target letters or numbers at the beginning of the trial in memory set and stores them in its declarative memory. Also the IBL process moves from one blip situation to another applying the process described below to each filled-in blip situation. The pattern of traversal between blip situations forms a Z (i.e., NW, NE, SW, and SE, respectively) until the frame time of 2.062 s runs out. If the IBL model cannot process all the filled-in blips before the frame time runs out, then it resets and starts at the NW filled-in blip for the next frame. Each of the IBL stages suggested in the IBLT (Gonzalez et al., 2003) is represented by production rules (if-then rules) in ACT-R:

Recognition On a trial if there is a recognition (or retrieval) failure or if the retrieved blip does not match the current situation blip, then apply the "wait for next blip" heuristic; otherwise if there is a recognition (or retrieval) success and a match between retrieved and current blips, then apply an instance-based judgment procedure.

Judgment On a trial if there is a recognition failure or if the retrieved blips do not match the current blip situation, then apply a wait for next blip judgment heuristic in which the spacebar is not pressed but the next blip situation is considered in a Z order. In case of recognition (or retrieval) success where the retrieved instance matches the current blip situation, apply an instance based judgment where the decision is to press the spacebar.

Choice The choice refers to picking the spacebar to press once the decision to press or not to press the spacebar has been made.

Execution Execute the spacebar or no spacebar press decision and wait for feedback from the system.

Also, in the above algorithm, the productions were assumed to take a commonly used value of 50 ms in ACT-R. There were some steps executed to read and encode the blip stimulus from the screen (i.e., visual time) in the model as well as some time expended in hearing deviant tones in the tone counting task that ran in the background. The visual and auditory times to see and hear each blip situation or each tone respectively were assumed to be at the ACT-R default values of 185 ms and 100 ms, respectively.

Sub-Symbolic Level of the IBL model

In ACT-R each instance (or chunk) has an activation value that is used for making retrieval in the recognition phase of the IBL model. An instance is retrieved from memory if the activation exceeds a retrieval threshold (RT), which sets the minimum activation with which an instance can be retrieved, and if the activation is the highest of all instance activations at the time of retrieval. ACT-R defines activation of an instance as:

$$A_i = B_i + \sum_l PM_{li} + \varepsilon \qquad (1)$$

Where B_i is the base-level activation and reflects the recency and frequency of practice of the *i*th instance, which is given by

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \qquad (2)$$

Where n is the number of presentations of the *i*th instance; t_j is the time since the *jth* presentation; and *d* is the decay parameter (bll) which is usually set at 0.5.

Specification elements l in the PM summation are computed over the slot values of the retrieval instance specification. Match Scale *P* reflects the amount of weighting given to the similarity in Slot l, which is a constant across all slots with the value set at 1.0. Match Similarities *Mli* represent the similarity between the value l in the retrieval specification and the value in the corresponding slots of the current instance *i*. The PM mechanism as described above was computed by the Blip-Situation slot of the instance. We used a function to calculate the similarity based on the absolute value of the distance between the Blip-Situation slot of the current instance and those retrieved from memory.

Finally, ε is the noise value, which is composed of two components: permanent noise associated with each instance and instantaneous noise computed at the time of a retrieval request. Both noise values are generated according to a logistic distribution characterized by a parameter *s*. The mean of the logistic distribution is 0 and the variance σ^2 is related to the *s* value by

$$\sigma^2 = (\pi^2/3)\, s^2 \qquad (3)$$

We set the instantaneous noise *s* value in the IBL model to make it a part of the activation equation.

For the purpose of modeling the RADAR task, the parameters described above had the values given in Table 2.

Table 2: IBLT (ACT-R) Parameters with Values

Parameter/Slots	Value
RT	-18.0
bll	0.5
s	0.25
P	1.0
Blip-Situation	Integers from 1 to 18

Strategy-Based Learning Model

In the SBL model we used four strategies. One of these strategies called "exhaustive equals" strategy was an optimal strategy, which would always yield the optimal press of the spacebar key and produce 100% accuracy in the detection task. The other three strategies were suboptimal strategies. These strategies represent practically feasible strategies for the task, and they provide competition that can be used to model performance, through the utility learning mechanism in ACT-R. The chunk structure for the SBL model was exactly the same as the one for the IBL model.

The SBL model starts by making use of one of the four strategies defined in the model (if a strategy could not execute before a frame ended, then the model resets and tries to apply strategies again in the next frame). When the model executes, there is a competition set up between the three suboptimal strategies and the optimal “exhaustive equals” strategy. The initial utility of the optimal strategy is set lower than that of the suboptimal strategies, and one of the suboptimal strategies executes in the task during the initial blocks. The suboptimal strategies give negative rewards, whereas the optimal strategy gives a positive reward whenever executed. The end effect is that although the suboptimal strategies fire initially, later the optimal strategy picks up because it has increased its utility through repeated positive rewards. Given below are the details of the different strategies in the RADAR’s SBL model.

Exhaustive Equals Strategy Compare all filled-in blips on the RADAR screen with all targets seen at the beginning of the trial and press spacebar if a match is found.

Random Equals Strategy Compare a randomly selected filled-in blip on the RADAR screen with a randomly selected target seen at the beginning of the trial and press spacebar if a match is found.

Bottom Two Equals Strategy Compare the bottom two (SW, SE) filled-in blips with all targets seen at the beginning of the trial and press spacebar if a match is found.

Top Two Equals Strategy Compare the top two (NW, NE) filled-in blips with all targets seen at the beginning of the trial and press spacebar if a match is found.

Each strategy is represented in an ACT-R production rule. Each production has a utility associated with it that can be set directly by setting a parameter :u. Like activations, utilities for productions could have noise added. The noise is controlled by the utility noise parameter s, which is set with the parameter :egs in ACT-R. The noise is distributed according to a logistic distribution with a mean of 0 and a variance of σ^2. If there are a number of productions competing with expected utility values Uj the probability of choosing production *i* is described by the formula:

$$\text{Probability (i)=Exp } (Uj/(2)^{0.5}s) \text{ / Sum(Exp } (Uj/(2)^{0.5}s)) \qquad (4)$$

The summation is over all the productions that are currently able to execute (their conditions were satisfied during the matching). Note however that Equation 4 only describes the production selection process. It is not actually computed by the system. The production with the highest utility (after noise is added) is the one chosen to execute. Also the utility learning mechanism updates the utility of a production (strategy) using the following equation:

$$Ui(n) = Ui(n-1) + \alpha * (Ri(n) - Ui(n-1)) \qquad (5)$$

If $U_i(n-1)$ is the utility of a production i after its n-1st application and $R_i(n)$ is the reward the production receives for its nth application (set by :reward parameter), then its utility is $U_i(n)$ after its nth application. In the above equation, α is the learning rate and is typically set at .2 (this value can be changed by adjusting the :alpha parameter with the sgp command). According to this equation the utility of a production is gradually adjusted until it matches the average reward that the production receives. A reward is delivered when a strategy fires, and the reward $R_i(n)$ that production i receives is the external reward received minus the time from the production’s selection to the reward. This subtraction serves to give less reward to more distant productions. This reinforcement goes back to all the productions that have executed between the current reward and the previous reward.

For the purpose of the RADAR task, the parameters as described above had the following values.

:egs 0.1 :ul t (9)
Exhaustive-Equals-Strategy :u -4 :reward +1
Random-Equals-Strategy :u 5 :reward -1
Bottom-Two-Equals-Strategy :u 10 reward -1
Top-Two-Equals-Strategy :u 5 :reward -1

The utility of the optimal strategy is lower than that of the three non-optimal strategies because we want to model to make errors similar to humans when it executes but reduce these errors overtime. The reward given to the suboptimal strategies decreases their utility, whereas the reward given to the optimal strategy increases its utility over time. The structure on utility and rewards might yield a monotonic dominance from the SBL approach even when changing environments and incorporating changes in the reward structure based upon changes in the environment is part of future work. Also, production compilation was not used in this model and it is a part of future work i.e. whether doing production compilation will make the SBL approach behave more like an IBL approach to modeling the experiment.

Model Fits to Human Data

The IBL and SBL models were run over 8 simulated participants in training and test conditions in RADAR. Figures 1 and 2 present the average times for correct responses during the training phase, including human data (Young et al., 2007) and SBL and IBL predictions. Figure 1 gives the average data for the within-subjects blocks CM1+1, CM4+4, VM1+1 and VM4+4. Both, the IBL and the SBL models fit the human data quite well, R^2=0.98 and RMSD=69 ms for IBL, and R^2=0.90 and RMSD=163 ms for SBL.

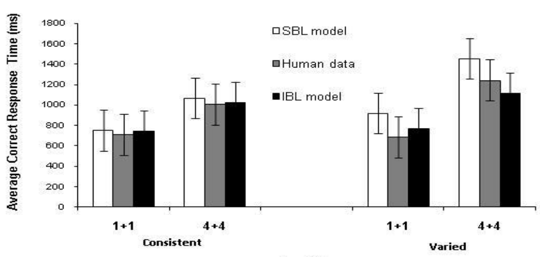

Figure 1: Average correct response times (ms) for CM 1+1, VM 1+1, CM 4+4, and VM 4+4 blocks in human data and SBL and IBL models during training. The error bars show 90% confidence intervals.

Figure 2 gives the average time for correct responses for the IBL, SBL, and human data across the silent and tone between-subjects conditions in the RADAR task. Again, both the IBL and the SBL models fit the human data very well, R^2=1.00 and RMSD=43 ms for IBL, and R^2=1.00 and RMSD=174 ms for SBL. In Figures 1 and 2, the SBL model seems to give generally higher time values compared to human data, and the SBL model has higher RMSD. This difference may be because in the SBL model the four strategies execute in productions in a fixed time (50 ms per production) and there is not speedup in the correct response times due to this fixed strategy execution time, whereas in the IBL model the speedup comes on account of activation-retrieval time speedup. The retrieval time decreases if the activation of instances increases over blocks (Anderson & Lebiere, 1998). Also, it is clear from Figure 1 that both models (i.e., IBL and SBL) take more time in 4+4 blocks than 1+1 blocks (for both consistent and varied mapping). This finding demonstrates the effects of workload well known in behavioral studies of automaticity (Gonzalez & Thomas, 2008). The workload effect results from the extra time taken to process four rather than one item.

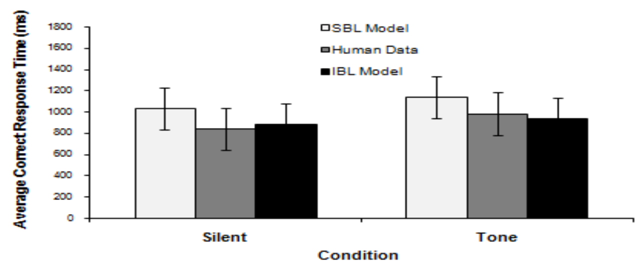

Figure 2: Average correct response times (ms) for silent and tone conditions for human data and SBL and IBL models during training. The error bars show 90% confidence intervals.

Similarly, the tone takes slightly more time to process than silent trials for both IBL and SBL models, as a result of the auditory productions to process the tones. Also, the difference is greater for the SBL model than the IBL model from the human data because in the SBL model there is no activation-retrieval speedup to compensate for time spent in tone counting whereas in the IBL model there is such a speedup, which reduces the overall time.

To test the adaptability of both SBL and IBL models and given the limited space in this paper, we report the data for only those groups that switch: tone-to-silent (Figure 3) and silent-to-tone (Figure 4). The R^2s for both the SBL and IBL models are very high at test (all are 1). Thus, the main difference between the models at test is in the RMSD measure. The SBL model has an RMSD = 160 ms when it is trained in tone and transferred to silent, whereas the IBL model's RMSD = 50 ms. The SBL model's RMSD when trained in silent and transferred to tone is 248 ms, whereas the RMSD value for the IBL model is 62 ms.

Thus, one can conclude that both models, SBL and IBL, are quite good according to the adaptability criterion, but the IBL model produces values closer to the human data than the SBL model does.

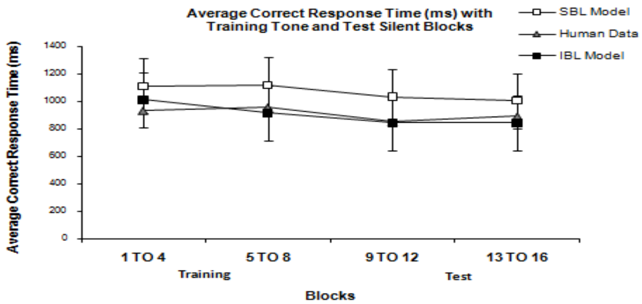

Figure 3: Average correct response times (ms) for human data and SBL and IBL models across blocks, for training in the tone and testing in the silent condition. The error bars show 90% confidence intervals.

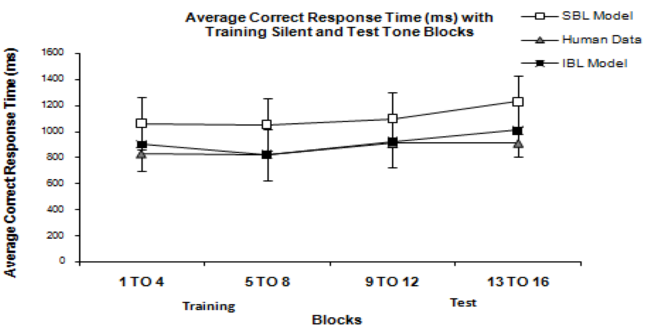

Figure 4: Average correct response times (ms) for human data and SBL and IBL models across blocks, for training in the silent and testing in the tone condition. The error bars show 90% confidence intervals.

Discussion and Future Work

Researchers often evaluate computational models of human behavior by comparing how different architectures or modeling approaches would represent a common task. This mode of model evaluation has been highlighted more recently by several model comparisons and competitions. The research we present here compares SBL and IBL approaches to modeling choice, but in this comparison in addition to using the same task, RADAR, we compare SBL and IBL approaches under the same architecture, ACT-R.

According to traditional goodness of fit measures, R^2 and RMSD, both SBL and IBL approaches to model choice fit human performance during a training experiment in RADAR quite well. Both representations are able to reproduce human data during the training conditions that varied both between subjects in tone/no tone training, and within subjects on the consistency of mapping and workload. When we compare the models in terms of their ability to adapt to transfer conditions, just as humans do, again both the SBL and IBL models have equally high values of R^2. But the IBL model was found to be closer to human data than the SBL model according to the RMSD measure during both training and test.

These results demonstrate that the numerical measures might not be good enough to tease two models apart. Further, the generalization criterion might not be sufficient either. To us, the IBL model has some advantage over the SBL model that the numerical measures do not show: Because the IBL model continues filling the chunk structure from the environment during test, the changes in conditions of the environment are captured in the instances stored and retrieved from memory, whereas the SBL approach is blind to changes in the environment. The SBL model continues applying the same strategies at test, which might not be as effective as they were during training, once the conditions of the task change. In addition, in dynamic situations the strategies are often unknown a priori or difficult to define at all. These are often discovered with task practice, and there is much evidence that learning in dynamic decision making tasks is implicit (Gonzalez et al., 2003). Often humans are unable to explain any rules or strategies used to solve a dynamic problem. Thus, we think that the IBL approach is more appropriate to model dynamic decision making (Gonzalez et al., 2003) than the SBL approach.

Acknowledgments

This research was supported by a MURI grant from the Army Research Office (W911NF-05-1-0153).

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of mind. *Behavioral and Brain Sciences, 26*, 587-639.

Busemeyer, J. R., & Wang, Y.-M. (2000). Model comparisons and model selections based on generalization criterion methodology. *Journal of Mathematical Psychology, 44*, 171–189

Cassimatis, N., Bello, P., & Langley, P. (2008). Ability, breadth, and parsimony in computational models of higher-order cognition. *Cognitive Science, 32*, 1304-1322.

Dutt, V., & Gonzalez, C. (2008). Instance and strategy based models in ACT-R. *Proceedings of 2008 Modeling, Simulation & Gaming (MS&G) Student Capstone Conference* (p. 19). Suffolk, VA: ODU-VMASC.

Gluck, K., Bello, P., & Busemeyer, J. (2008). Introduction to special issue: Model comparison. *Cognitive Science, 32*, 1245-1247.

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

Gonzalez, C., & Thomas, R. P. (2008). Effects of automatic detection on dynamic decision making. *Journal of Cognitive Engineering and Decision Making, 2*, 328-348.

Lebiere, C., Gonzalez, C., & Warwick, W. (2009). *Emergent complexity in a dynamic control task: Model comparison*. Manuscript submitted for publication.

Lovett, M. C. (1998). Choice. In J. R. Anderson, & C. Lebiere (Eds.). *The atomic components of thought* (pp. 255-296). Mahwah, NJ: Erlbaum.

Nisbett R. E. (Ed.) (1993). *Rules of reasoning*. Erlbaum Associates.

Young, M. D., Healy, A. F., Gonzalez, C., & Bourne, L. E., Jr. (2007, July). *The effects of training difficulty on RADAR detection*. Poster presented at the joint meeting of the Experimental Psychology Society and the Psychonomic Society, Edinburgh, Scotland.