The Importance of Action History in Decision Making and Reinforcement Learning

Yongjia Wang (yongjiaw@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109-2121

John E. Laird (laird@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109-2121

Abstract

We investigate the hypothesis that historical information plays an important role in learning action selection via reinforcement learning. In particular, we consider the value of the history of prior actions in the classic T maze of Tolman and Honzik (Tolman & Honzik 1930). We show that including a sequence of actions in the state makes it possible to learn the task using reinforcement learning. Moreover we show that learning over sequences of length 0 ~ 4 is necessary to model rat behavior. This behavior is modeled in Soar-RL and compared to an earlier model created in ACT-R.

Introduction

In many tasks, immediate sensory data is insufficient for decision making. Enriching the state with information about previous actions or previous situations can disambiguate between situations that would otherwise appear identical, which makes it possible not only to make correct decisions but also to learn the correct decision. Moreover, knowledge of the past can replace the need for unrealistic sensors, such as knowing the exact location in a maze.

Using historical information as part of the state representation poses some challenges. For the tasks we describe here, we use a simplified version of history – a sequence of prior actions. This leaves open the length of sequence, and how to model the relation between similar sequences to achieve proper level of generalization and specialization during learning. We demonstrate how these issues can be addressed in Soar-RL (Nason, & Laird, 2005) by proposing a simple model on an animal based experiment. We analyze the task and compare results to a recent ACT-R model (Fu & Anderson 2006).

The T Maze Task

The task we will explore is the T maze task of Tolman and Honzik (Tolman & Hoznik 1930) in which a rat is put at the start location and it is rewarded if it gets to the end location. As shown in Figure 1, the T maze contains 14 numbered blinds (dead-ends), each corresponds to a binary choice point (the task is designed to prohibit the rats from going back at T-junctions). Whenever the rat turns into a dead-end, that is considered an error. In such a maze, there are few if any salient features. Rats are able to maintain a sense of direction, so that would provide the ability to create for different classes of T's. The only other salient features appear to be a history of the rat's behavior – that is the sequence of turns it made before coming to a T at which it must make a decision.

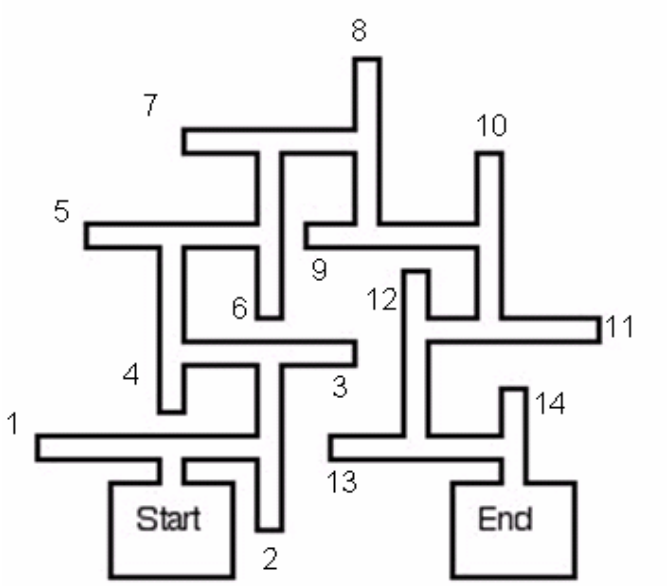


Figure 1. T maze used in Tolman and Honzik (1930)

To cast this as a problem conducive to reinforcement learning, we use the same conventions as a recent ACT-R model on this task (Fu & Anderson 2006). Moving into dead-ends and turning back results in immediate negative reward, while reaching the final goal results in positive reward. Figure 2 shows a picture of the actual environment, where the maze is embedded in a grid world and the subject moves one unit at a time. Dark boxes represent penalties, and the light box represents the final reward.

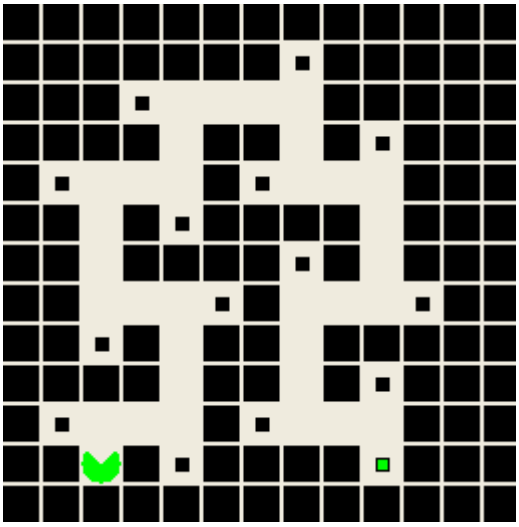


Figure 2. T-maze model

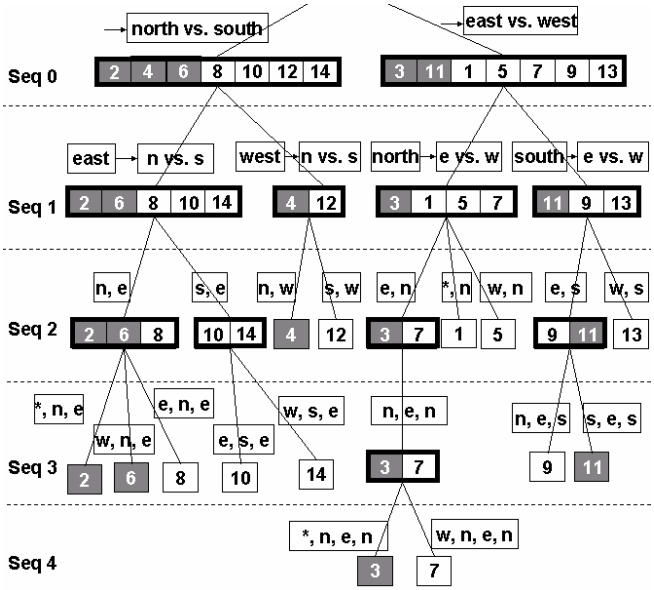


Figure 3. Relations among choice points

Qualitative Analysis of Task Constraints

Given the dearth of features in the environment, the only external features available to the rat are its prior moves. Thus, we assume the representation of the state includes a sequence of previous moves. The moves could be encoded relative to the current heading: left, right, forward, backward; however, as pointed out in (Fu & Anderson 2006), the rats have strong directional bias, and thus we assume they have knowledge of absolute direction and have available the absolute directions of their movement To describe the model, we use north, east, south and west as labels for these directions. For example, at choice point 6, the state includes the sequence of [east, north, west, …] ordered left-to-right by recency, so that the first item in the sequence is the current direction.

Figure 3 shows the relationships among the choice points associated with each numbered dead-end based on the sequence representation described earlier. Choice points that are grouped together have the same previous input sequence and face with the same set of choices. Within the same group, points are further divided based on what is the correct choice. Decision points, for which moving north (2, 4, 6) or moving west (3, 11) are correct, are colored in light number with dark background; other points are colored in dark number with light background. Points in the same group but with different color are competing points in that learning to reduce the error for one type of points will simultaneously increase the error for the other type of points. Interference is most intense for the most general level (Seq 0), and disappears at the most specific level (Seq 4), where the correct decision can be learned for each choice point. The tree structure in Figure 3 therefore captures all such constraints in the task model.

Our hypothesis is that choice points with similar state representations (in this case the sequence of prior moves) will appear similar to the rat and it will learn to make the same decisions in those states. Choice points with different correct directions but similar state representations will interfere with each other during learning. According to Figure 3, if the agent makes decision based on Seq 0, for example, it will tend to move south more than north and east more than west at each choice point where those options are available, since south and east correspond to the correct choice for the majority of the choice points within each group (4 south vs. 3 north and 5 east vs. 2 west). At Seq 4 (the most specific level), all choice points are completely discriminated and the correct decision can be made at each choice point.

Our assumption is that sequences of prior actions are maintained and available for decision making. Figure 3 provides the information necessary to determine what impact each sequence can have on learning. Relying solely on sequences of length 0, a rat should tend to make more errors at points 2, 4, 6, 3, and 11. Relying solely on sequences of length 1, point 4 should involve less error than point 2 and 6, since point 4 is discriminated from majority of conflicting points (especially the strongest point 14) but only interfere with point 12. Point 4 will be correctly learned at the next specificity level, while point 2 and 6 are still confused with point 8. Point 3 will involve more errors than point 11 since it is not discriminated from point 7 until sequence length of 4.

One important property of most approaches to learning these discriminations is that learning is quicker for more general levels because they are exposed to more examples. For example, there are 4 different rules (different combinations of states and legal actions) at the level of seq 0, each of them will receive a quarter of the total training instances, while at the most specific level of Seq 4, there are

28 different rules, each of them only receives less than 4% of total training instances. This suggest there is an advantage to including selection knowledge based on all levels of the sequences so that some rough knowledge can come into play early, but more and more specific knowledge is learned over time. No deliberate mechanism is required to achieve this effect.

These conclusions are largely consistent with the experimental data from the T-maze task as shown in Figure 4.

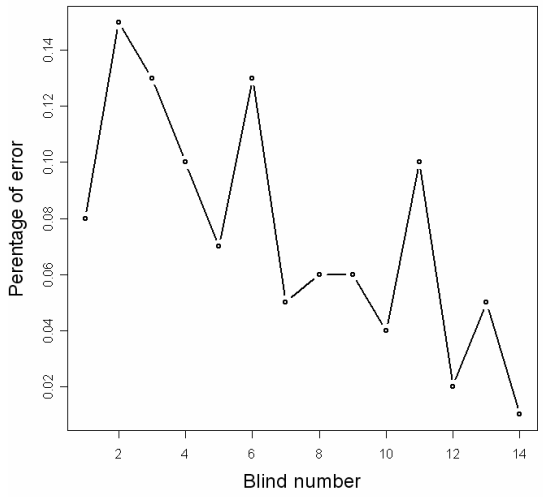


Figure 4. Percentage error in Honznik (1930)

Soar Reinforcement Learning Model

As mentioned above, our hypothesis is that the model must consider the spectrum of specificity levels of the state representations and that these will influence learning and behavior. In Soar, this effect can be readily modeled because Soar allows knowledge for selection of an action to be encoded in multiple rules that fire together in parallel, each providing its own prediction of the expected utility of the operator. The expected utilities for the same operator are combined, producing a single, joint expected probability. Thus, when making a decision, rules match and fire for each of the levels, for each of the possible actions. Thus, we can capture all of the levels of specificity in Figure 3. Once a decision is made, all the rules that contributed to the selected action update their expected utility values.

The effect is that general rules will have the most influence for decisions at novel situations where specific rule hasn't been learned yet. In these situations, the expected values created by the specific rules will be relatively weak with values still close to the initial value of 0. As learning progresses, more and more of the specific rules will have sufficient examples so that their learning stabilizes and their values, combined with corresponding general rules, reflect the expected utility of those situations.

Soar-RL

Soar reinforcement learning implements the general temporal-difference learning. The learned policy is represented as a Q value function as in standard Q learning. A Q value reflects the utility of taking a particular action in a particular state. In Soar-RL, a Q value is associated with each state-action pair represented as a Soar RL production rule. The update function in the case of multiple rules firing is as the following. A temporal difference is computed based on the sum of Q values for all rules that match the current condition, and is evenly distributed to update each rule. Since more general reinforcement learning rules fire more often, and a specific rule will always fire with the same general rule (there is a strict hierarchy in this task), the result is that the general rule quickly learns generalized Q value with relatively fewer trainings, while specific rules will fine tune the total Q value for specific situations and stabilize after receiving more training examples. Without general rules, the model has to directly learn the specific rules without useful initial bias in novel situations where more general rule could have helped. Without specific rules, on the other hand, it cannot learn the precise policy.

The probability of making a particular choice is calculated based on the Boltzmann distribution (equation 1). In the binary choice case of this task model, it can be rewritten as equation 2, therefore the probability of making the wrong choice P_{wrong} is a monotonic function of the Q value difference quantity $Q(s, a_{wrong}) - Q(s, a_{correct})$. Here the Q value represented as a function of a state-action pair, where a_{wrong} stands for the wrong action and $a_{correct}$ stands for the correct action.

$$P_i = \frac{e^{Q(s,a_i)/Temperature}}{\sum_i e^{Q(s,a_i)/Temperature}} \qquad \text{Equation 1}$$

$$P_1 = \frac{e^{(Q(s,a_1)-Q(s,a_2))/Temperature}}{1+e^{(Q(s,a_1)-Q(s,a_2))/Temperature}} \qquad \text{Equation 2}$$

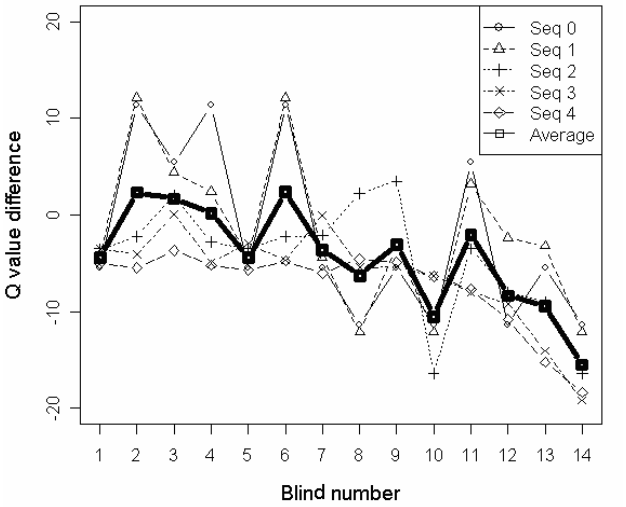
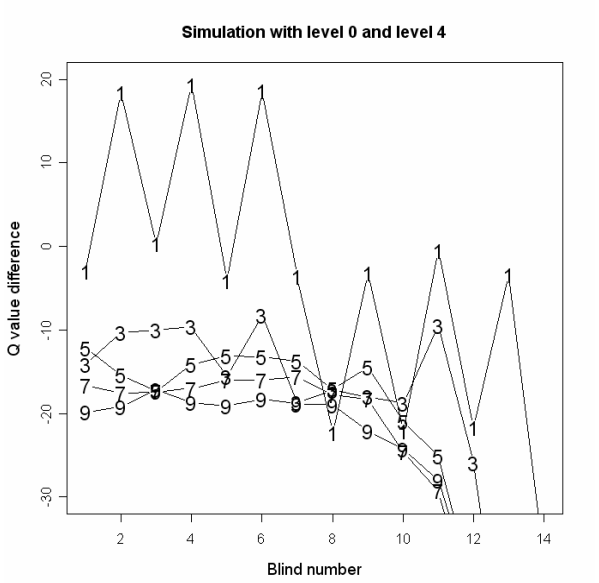


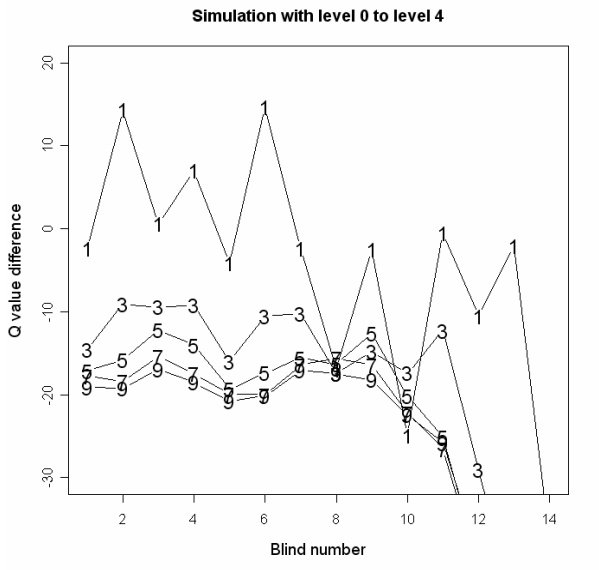
Figure 5. Effects of reinforcement learning rules with state representation at different specificity levels

Figure 5 plots the Q value difference = $Q(s, a_{wrong}) - Q(s, a_{correct})$ at each choice point for reinforcement learning rules with different specificity level (from Seq 0 to Seq 4). The Q values are learned separately and each is an average from 10 independent simulations for 17 trials. These Q value difference curves show the convergences trends for rules at different specificity level. The plot qualitatively illustrates how rules at each specificity level will affect the relative error rate shown in Figure 4. The initial error rate distribution should be similar to the curve Seq 0, but as more and more specific decisions are learned it eventually converges to the curve of Seq 4, the most specific level, as explained in the analysis presented in the previous section. The plot can be viewed approximately as a contour of Q value difference updating dynamics, since when all levels of rules are used in Soar, the total Q value difference will gradually converge following the path which is consistent with our empirical results (data not shown). One specific interpretation from Figure 5 is that initial error for point 4 is relatively higher than point 3, but it learns faster and results in lower total error rate. Qualitatively, the average Q value difference across all specificity levels, which is shown as a bold curve in Figure 5 approximates the relative total error rates for each dead-end. This can be confirmed by comparing with Figure 4.

Figure 5 only shows the qualitatively analysis based on separate simulations of each individual level. It is more informative to examine the combined Q value difference of all rules during learning.



(a)



(b)

Figure 6. Change of combined Q value difference during learning.

The numbers in Figure 6 refer to trials, with 20 trial intervals. For example, the curve with 1 represents the Q value difference after trial 1, 3 represents after 21 trials. There are totally 81 trials shown in the plot to demonstrate the Q value dynamics, although the actual rat experiment only takes 17 trials. (a) is learning with only the most general rules and the most specific rules. (b) is learning with all levels of rules. One of the main differences between (a) and (b) is point 3 is learned relatively slowly when using all levels of rules. The dynamics of learning is consistent with Figure 5 and the above analysis.

Results

Figure 7 compares observed data with prediction using all 4 levels of rules. The parameters are penalty for turning back -20, reward for reaching the goal +100, learning rate 0.1, linear discount of 10, on-policy learning with Boltzmann exploration temperature 3. Linear discount is used because it gives better results than standard exponential discount. Figure 5 and Figure 6 are generated using standard exponential discount and they are for illustration purposes. The most important parameter is the learning rate, and the results are not very sensitive to other parameters.

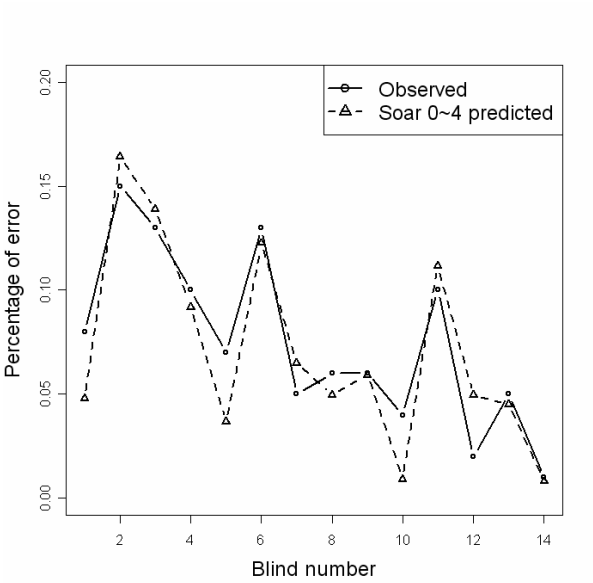
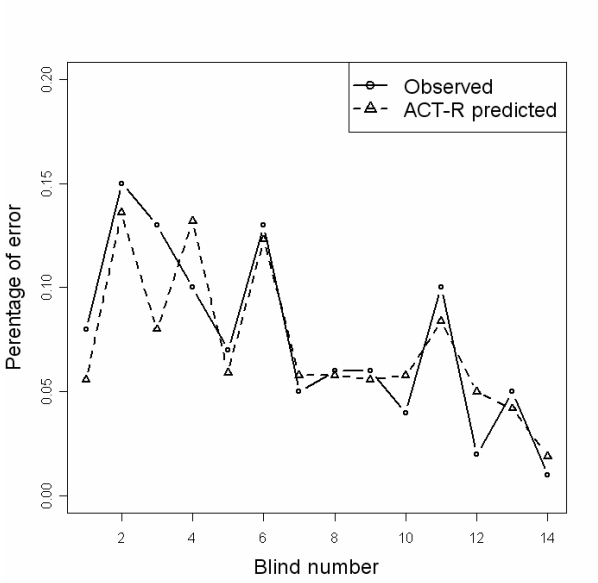


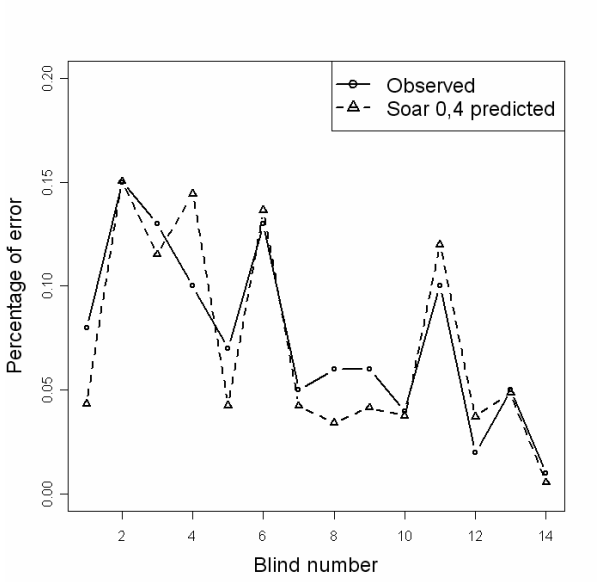
Figure 7. Soar model prediction

Comparison with ACT-R

An ACT-R model (Fu & Anderson 2006) was developed to model the Tolman and Honzik (1930) experiment, relying on ACT-R's native reinforcement learning component. In ACT-R, there are weights associated with rules. Learning adjusts those weights, which are used in selection. Each rule corresponds to one action and there is no explicit combination of values or joint updating of rules that are for the same action. The ACT-R model uses two sets of rules: a set of twenty-eight specific rules, two for each choice point; and a set of four general rules, one for each absolute direction. The specific rule set is equivalent to the level of seq. 4 (the model does not use sequences, assuming a rat knows its position in the maze), and the general rule set is equivalent to seq. 0 in Figure 3.



(a)



(b)

Figure 8. (a) Prediction using ACT-R model.
(b) Prediction using Soar model with equivalent rules.

Figure 8 (a) shows the ACT-R prediction with only the most general rules and most specific rules, which is equivalent to using only Seq 0 and Seq 4 in the Soar model. Figure 8 (b) shows the prediction using Soar 0, 4 model, which is similar to the ACT-R model especially for blind 3.

Table 1. Correlation Matrix comparing all models

	Observed	Soar0~4	Soar 0,4	ACT-R
Observed	-	0.92	0.90	0.86
Soar 0~4	-	-	-	0.82
Soar 0,4	-	-	-	0.94
ACT-R	-	-	-	-

Table 1 compares the correlations of the ACT-R model and Soar model. The Soar 0, 4 model predicts the ACT-R model very well (correlation 0.95), while Soar 0~4 model predicts the experimental data better (correlation 0.91) than the other models. The differences between the correlation coefficients are statistically significant. For 0.86 versus 0.91, the p value is < 0.001 assuming the original rat data and ACT-R data both have the same variance as our simulation data. This (weakly) suggests that the rats learn to make decisions using a history of prior decisions.

Table 2. Correlation with partial observed data

	Soar0~4	Soar 0,4	ACT-R
Blinds 1~6	0.98	0.82	0.71
Blinds 10~14	0.86	0.98	0.87

Taking a closer look at the results, Soar 0~4 matches the blinds closer to the beginning much better while the Soar 0,4 model matches well for those closer to the end. Table 2 compares the correlation with partial experimental data. One hypothesis could be that when the rat is at the choice points close to the end, the general rules give good results, so that it less depends on lower level rules. While at the beginning, where the general rules give bad results, it also uses the more specific rules. This hypothesis suggests adjusting the weights of rules at different levels for different choice points, but that introduces more parameters and is probably beyond what can be confirmed from the available data.

Table 3. Comparison between the models

	Model Level	Architectural Level
Soar	Use action history	Parallel rule firing
ACT-R	No action history	Single rule firing

Table 3 compares the two levels of difference between the ACT-R model and our Soar model. Clearly the most important difference is our model's learning over multiple sequences of past actions (the model level difference). It is reasonable to assume that representing that information in the state and increasing the number of rules in ACT-R would improve the ACT-R model's match to the observed data, especially for the early choice points.

A detailed comparison between reinforcement learning in ACT-R and Soar has already been made by Nason (Nason & Laird 2004). However, this task model highlights an important difference between the two approaches. In Soar, for a single decision, multiple reinforcement learning rules are allowed to contribute to the decision making and then are updated by learning. In ACT-R, although multiple rules contribute to making a decision through competition, only one is picked and updated. Soar speeds learning with multiple reinforcement learning rules in terms of requiring fewer external actions, although the asymptotic behavior of the two approaches should be similar. This architectural level difference is secondary for the results presented here – it is the action history representation (model level difference) that makes the qualitatively different predictions in our hypothesis. However, it may be worthwhile to explore the importance of this architectural level difference in other applications.

Another difference is the reward discount functions used in Soar and ACT-R. The default option in Soar is to multiply future expected reward with a discount factor γ ($0 < \gamma < 1$) in the step-wise update function, which results in exponential decay of rewards. We experimented with linear discount (constant discount between steps) which generates slightly better results. The compared ACT-R model uses a hyperbolic discount function, which might help our model to make better predictions especially for those later choice points. In general, it's flexible to experiment with different options in the Soar architecture.

Discussions

The major contributions of this paper are to examine the contribution of sequences of action histories, to decision-making and learning. The second major contribution was to evaluate the approach to representation and updating of expected values in Soar-RL and discovering that they provide an accurate model of learning dynamics by having overlapping rules at different specificity levels. Functionally, learning progresses from generalize to specific. The ACT-R model provided a useful benchmark for comparison.

We can also ask where our model falls short. Our model does not make a good prediction at blind 12. This could be due to experimental data noise, but it's more likely that there is more structure in the task that is not captured by our model. One possibility can be that instead of always using absolute directions, the rats may actually use combinations of absolute and relative directions, such as turning left and right as the state encoding strategy.

References

Fu, Wai-Tat & Anderson, J. R (2006). From Recurrent Choice to Skill Learning: A Reinforcement-Learning Model. *Journal of Experimental Psychology: General, 135(2) 184-206*

Nason, S., Laird, J. E., (2005) Soar-RL: Integrating Reinforcement Learning with Soar, Cognitive Systems, Volume 6, Issue 1, pp. 51-59.

Nuxoll A.& Laird J.E. (2004). A Cognitive Model of Episodic Memory Integrated with a General Cognitive Architecture, *International Conference on Cognitive Modeling*.

Tolman E. C. & Honzik, C. H. (1930). Degrees of hunger, reward and non-reward, and maze learning in rats, *University of California Publications in Psuchology, 4(16), 241-256*.