Effects of Decision Complexity in Goal-seeking Gridworlds: A Comparison of Instance-Based Learning and Reinforcement Learning Agents

Thuy Ngoc Nguyen (ngocnt@cmu.edu) and Cleotilde Gonzalez (coty@cmu.edu)
Dynamic Decision Making Laboratory
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA

Abstract

Decisions under uncertainty are often made by weighing the expected costs and benefits of the available options. The costs-benefits tradeoffs may make decisions easy or difficult, particularly given uncertainty of these costs and rewards. In this research, we evaluate how a cognitive model based on Instance-Based Learning Theory (IBLT) and two well-known reinforcement learning (RL) algorithms learn to make better choices in a goal-seeking gridworld task under uncertainty and on increasing degrees of decision complexity. We also use a random agent as a base level comparison. Our results suggest that IBL and RL models are comparable in their accuracy levels on simple settings, although the RL models are more efficient than the IBL model. However, as decision complexity increases, the IBL model is not only more accurate but also more efficient than the RL models. Our results suggest that the IBL model is able to pursue highly rewarding targets even when the costs increase; while the RL models seem to get "distracted" by lower costs, reaching lower reward targets.

Keywords: decision complexity; instance-based learning theory; reinforcement learning; goal-seeking task.

Introduction

Goal-seeking in gridworld navigation, has long been a classical task for developing Artificial Intelligent (AI) agents. Generally, the agent must navigate an environment (e.g., gridworld) with uncertainty about the surroundings to achieve a goal (i.e., consuming the highest rewarding object) given a number of obstacles and within a time limit. This type of task underlies a broad range of applications such as search and rescue or pickup and delivery missions.

Researchers have commonly addressed this type of task using Reinforcement Learning (RL) models, a computational method of learning from interaction (Sutton, Barto, et al., 1998; Gershman & Daw, 2017). A major challenge for research in AI is to develop systems that can replicate human behavior; and although there is much evidence of RL's ability to account for human behavior in some dynamic decision tasks (Gureckis & Love, 2009; Simon & Daw, 2011), the concern has been raised that the advance in RL paradigms is mostly centered on solving computational problems efficiently, rather than replicating or explaining in detail how humans learn (Botvinick et al., 2019).

Cognitive modeling on the other hand, is aimed at understanding and interpreting human behavior by representing the cognitive steps by which a task is performed. In particular, Instance-based Learning Theory (IBLT) was developed to provide a cognitively-plausible account for how humans make decisions from experience and under uncertainty, through interactions with dynamic environments (Gonzalez, Lerch, & Lebiere, 2003). IBLT has shown accurate representation of human choice and broad applicability in a wide number of decision making domains, from economic decision making to highly applied situations, including complex allocation of resources and cybersecurity, e.g. (Hertwig, 2015; Gonzalez, 2013; Gonzalez et al., 2003).

Nevertheless, in goal-seeking gridworld navigation tasks, cognitive models of decision making, and IBL models in particular, have been less common. Fu and Anderson (2006) proposed a RL mechanism within the ACT-R architecture, to account for repeated choice and skill learning. The study used a maze learning task, and showed that the model can fit human data fairly well to account for complex learning in this task. Also, Reitter and Lebiere (2010) proposed an ACT-R cognitive model, to address the aspects of human path-planning problems, which are relatively similar to navigation. Finally, in a prognostic foraging task, Chelian and colleagues showed that both IBL models and RL approaches can imitate human decision making well (Chelian, Paik, Pirolli, Lebiere, & Bhattacharyya, 2015). Despite all of these advances, it remains unclear how RL and IBL models compare with respect to representing human decisions under uncertainty.

To that end, the primary goal of this work is to examine how RL and IBL agents learn in a goal-seeking gridworld task under different degrees of decision complexity. Decisions under uncertainty are often made by weighing the expected costs and benefits of the available options. Some decisions are easy (e.g., choosing between an option of low cost and high expected reward and one with high cost and low reward), while others are complex (e.g., choosing between low cost low reward, and high cost and high reward options). These decisions' complexity increases given uncertainty in the costs and rewards. Thus, we first leverage IBLT, to develop an IBL model of an agent that is able to accomplish the goal-seeking task in a gridworld environment under different levels of decision complexity. Using simulation experiments, we explore the impact of decision complexity on the performance of different types of agents, RL and IBL, including a Random that serves as a baseline comparison for the models.

Instance-Based Learning Theory

IBLT is a theory of decisions from experience, developed to explain human learning from interaction with dynamic decision environments (Gonzalez et al., 2003). IBLT provides an algorithm and a set of cognitive mechanisms that can be used to implement computational models of decision learning processes. The algorithm involves the recognition and retrieval of past experiences (i.e., instances) according to their similarity to a current decision opportunity. Instances retrieved are

used to calculate the expected utility of a potential decision in such situation. Potential decision alternatives a are evaluated sequentially, and a process of choice provides a stopping point for evaluating potential alternatives and making a choice. The choice alternative with the highest expected utility among a set of alternatives is selected. Finally, a feedback process updates the expected utility of past instances with the observed actual outcome of choices executed. Such updated instances are then reused in future decisions.

An "instance" in IBLT is a memory unit, that results from the potential alternatives evaluated. These are memory representations consisting of three elements: a situation (S) (set of attributes that give a context to the decision, or state s); a decision (D) (the action taken corresponding to an alternative in state s, or action a); and a utility (U) (expected utility u or experienced outcome x of the action taken in a state). The essential sub-symbolic mechanisms of IBLT have been discussed in multiple past publications (e.g. (Gonzalez et al., 2003; Gonzalez & Dutt, 2011; Gonzalez, Ben-Asher, Martin, & Dutt, 2015; Hertwig, 2015)), but we include these mechanisms here for completeness.

Each instance i in memory has a value of *Activation*, which represents how readily available that information is in memory (Anderson & Lebiere, 2014). The instance could be perfectly or partially matched to the attributes of a decision opportunity at the current point of time, which is determined by the partial matching mechanism (Anderson & Lebiere, 2014). But here we consider a simplified version of the Activation equation which only captures how recently and frequently the considered instances are activated:

$$A_i = \ln\left(\sum_{t' \in \{1..t-1\}} (t-t')^{-d}\right) + \sigma \ln \frac{1-\gamma_i}{\gamma_i}, \qquad (1)$$

where d and σ are respectively the decay and noise parameters; t' refers to the previous timestamp in which the outcome of instance i was observed resulting from choosing an action a at state s. The rightmost term represents the Gaussian noise for capturing individual variation in activation, and γ_i is a random number drawn from a uniform distribution $U(0,1)$.

Activation of an instance i is used to determine the probability of retrieval of such instance from memory. The probability of an instance i is a function of its activation A_i relative to the activation of all other instances corresponding to executing action a at state s:

$$p_i = \frac{e^{A_i/\tau}}{\sum_l e^{A_i/\tau}}, \qquad (2)$$

where τ is the Boltzmann constant (i.e., the "temperature") in the Boltzmann distribution (Kittel, 2004).

For simplicity, we defined τ as a function of the same σ parameter used in the activation equation $\tau = \sigma\sqrt{2}$. The parameter τ gives some variability to the probability of retrieving instances from memory.

The expected utility of taking action a in state s is calculated based on a mechanism called *Blending* (Lebiere, 1999) as specified in IBLT (Gonzalez et al., 2003), using the past experienced outcomes stored in each instance x. Here we employ the blended value that was defined and used for binary choice tasks in Lejarraga, Dutt, and Gonzalez (2012); Gonzalez and Dutt (2011):

$$V(a,s) = \sum_{i=1}^{n} p_i x_i. \qquad (3)$$

Essentially, according to (Gonzalez & Dutt, 2011), Blending (Equation 3) is the sum of all the past experienced outcomes weighted by their probability of retrieval, where x_i is the outcome stored in an instance i associated with taking action a at state s; p_i is the probability of retrieving the instance i from memory (Equation 2); and n is the number of instances stored in memory for taking action a up to the last trial.

The choice rule is to select the action a that corresponds to the maximum blended value.

Goal-seeking Task in Gridworld Environment

A gridworld environment is made up of a 11×11 grid maze as illustrated in Figure 1. Each gridworld contains randomly-located obstacles (black bars). The number of obstacles varies from one to five and their size ranges from one to six 1×1 cells. There are four targets of different values, which are represented as four colored objects (blue, green, orange, and purple) of size of 1×1 in the grid and set at random locations in a way that does not overlap with the obstacles.

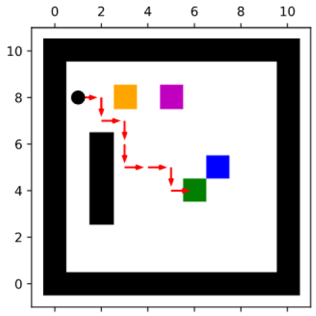


Figure 1: Illustration of the goal-seeking task in the gridworld environment. The agent's preferred goal is the "green" object.

The primary task is a goal-seeking problem in the gridworld environment, where an agent (black dot), starting in a random location (i.e., (x,y)), moves through the 11×11 grid to search for the most valuable goal among the four objects, while avoiding obstacles. The agent is tasked with consuming the object that has the highest reward (i.e. "green" in Figure 1) within a 31 step limit. Starting in its initial position, the agent makes sequential decisions about which actions to take (i.e., up, down, left, right). An episode ends when the agent decides to "consume" *any* of the four objects, or by reaching the 31 step limit without a consumption.

A sequence of moves from the initial location to the end location forms a *trajectory* (dotted red line) which is produced

by the sequence of decisions that the agent adopts. Each agent performs the task over 500 episodes for learning in the same gridworld.

Technically, each agent $\mathcal{A}_k$ is driven by a fixed reward, $r_{k,j} \in (0,1)$, for consuming an object o_j where $j = 1,\ldots,4$. Hence, the vector $r_k = (r_{k,1},\ldots,r_{k,4})$ has four components (one for each of the four objects), and it was drawn from a Dirichlet distribution ($\sum_{j=1,\ldots,4} r_{k,j} = 1$ and $r_{k,j} > 0$) with concentration parameter $\alpha = 0.01$, which signified that the agent $\mathcal{A}_k$ was favourably attracted to one of the four objects. In other words, if the agent successfully consumes the most preferred object, it will receive the highest reward while consuming any of the other 3 objects (i.e. the distractors) results in receiving much smaller rewards. Besides, the agent is penalized 0.01 for each step, which is a movement cost, and 0.05 for walking into a wall.

Agents in the Gridworld

IBL Agent

In the gridworld task, an *instance* is defined by a triplet (s,a,x), where x is the outcome or expected utility resulting from taking action a (i.e., up, down, left right) in state s (i.e., the state is the location of the agent, defined by the x-y coordinates) in a grid (Nguyen & Gonzalez, 2020). When making a prediction about which action a the agent $\mathcal{A}_k$ will take at state s, the IBL agent selects the action with the highest expected utility using the *blended* value (Equation 3).

Importantly, the agent only gets a positive outcome when consuming an object after a sequence of decisions. Thus, the IBL agent must learn to update the expected utility from the outcome received after consuming an object, so that different instances created by the trajectory are reinforced accordingly. The delayed feedback mechanism proposed in IBLT (Gonzalez et al., 2003) is underdeveloped, and most of the tasks that IBLT has been applied to, include immediate feedback. Thus, a mechanism to deal with delayed feedback is required in the gridworld task. Unlike prior work that focused on how humans learn from delayed feedback (Walsh & Anderson, 2011; Kelly & West, 2013), we simply use the final outcome and distribute it equally to all actions taken in a trajectory. That is, considering the trajectory $\mathcal{T}_k = \{(s_t,a_t)\}_{t=0}^{T}$ if the $\mathcal{A}_k$ gets the outcome x' at the end of the episode ($t = T$) then the expected utility of executing $\{(s_t,a_t)\}_{t=0}^{T-1}$ is all updated to x'. We leave the alignment to human judgements of delayed feedback for future research.

RL Agents

We compare the performance of the IBL agent against two RL agents called Q-learning and SARSA which are well-known temporal difference techniques in RL (Sutton et al., 1998). The basic difference between these two RL algorithms is in the way of updating a value of current state-action pair. In SARSA (*on-policy* method), the update takes into account the value of the actual action taken at one state ahead of the current state whereas in Q-learning (*off-policy* method), it simply considers the highest possible action that can be taken at the current state.

Q-learning Agent. A Q-learning agent was implemented with a tabular form of Q-learning algorithm (Sutton et al., 1998). In general, the goal of the RL agent $\mathcal{A}_k$ is to estimate the optimal state-action values referred to as Q-values, where $Q(s,a)$ returns the expected future reward of action a at state s. Initially, all the Q-values are set to zero and then are iteratively updated. Given enough iterations, the agent can learn the optimal Q-values denoted by $Q^*(s,a)$, and for each state s the agent selects the action having the highest Q-value, $\pi_k^*(s) = \text{argmax}_a\, Q^*(s,a)$.

SARSA Agent. A SARSA agent was designed based on the SARSA algorithm. The name SARSA comes from the fact that the updates depend on a quintuple of events (s,a,r,s',a'), where s and a are the current state and action of the agent, r is the observed reward for choosing the action a, and s' and a' are the new state-action pair. Essentially, SARSA, in contrast to Q-learning, learns the value of each state–action pair (i.e. the Q-value) by looking ahead to the next action to see what the agent will perform at the next step and then update the Q-value of its current state-action pair accordingly.

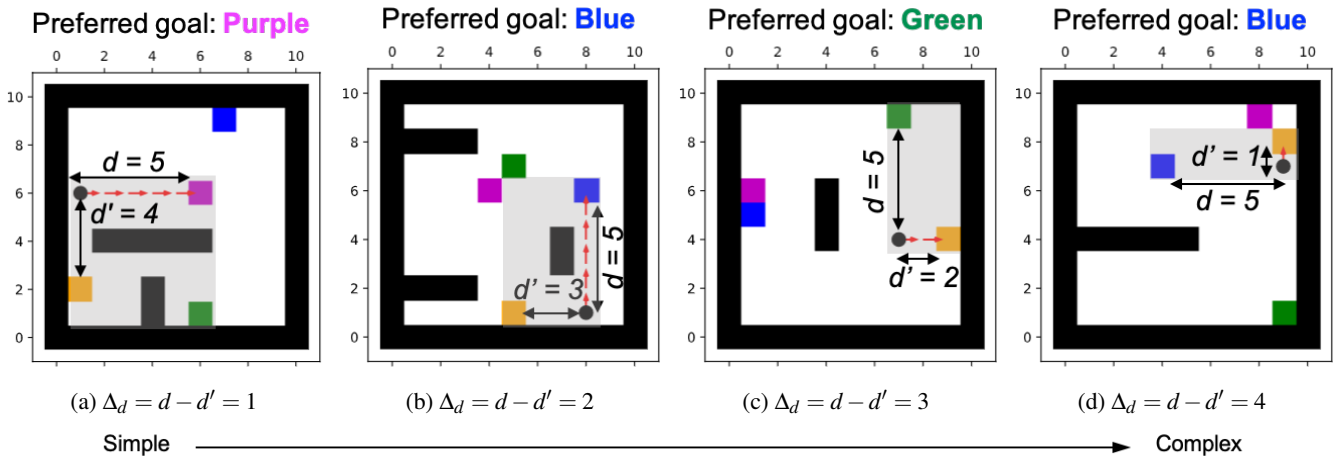
Random Agent

A random agent $\mathcal{A}_k$ selects an action a in state s based on the probability $\pi_k(a|s)$. Precisely, the policy of $\mathcal{A}_k$ is drawn from a Dirichlet distribution $\pi_k \sim Dir(\alpha)$ with concentration parameter α, so that $\sum_{a\in A} \pi_k(a|s) = 1$ and $\pi_k(a|s) > 0$. If α is close to 0 then the policy of an agent is characterized to be near deterministic. Conversely, the action of the agent is far more stochastic if α is much greater than 0.

Experiments

To investigate how different agents perform under different levels of the decision complexity, we designed experimental manipulations in which we control cost-benefit tradeoffs of choices made in a gridworld task.

Inspired by general decision processes and animal behavior, we designed levels of complexity. In animal foraging the complexity involves a tradeoff between the quality of food and the effort of obtaining it, and this tradeoff also applies to human decision processes (Mehlhorn et al., 2015). A gridworld can be more complex when its arrangement of goals and obstacles creates a high conflict between benefits (i.e., the object's reward) and associated costs (i.e., the distance, or number of steps needed to consume that object). For instance, a setting in which an agent must decide whether to consume a close (e.g., one step distant from the current agent's location) but low-reward object or to search for a far-away but higher reward object is more challenging than a decision between a close and high-reward object and a far and low-reward object. We refer to low-reward objects as "distractors" and to the highest reward object as the "preferred object". Agents



(a) $\Delta_d = d - d' = 1$ (b) $\Delta_d = d - d' = 2$ (c) $\Delta_d = d - d' = 3$ (d) $\Delta_d = d - d' = 4$

Figure 2: Illustration of the designed gridworlds with the level of complexity increasing from left to right.

generally prefer high value objects but they need to explore the environment to learn the value of the four objects, since this is only known after they consume an object.

In our experiment, complexity is characterized by the difference between the distance from an agent to the preferred object (d) and the distance from the agent to the closest distractor (d'), i.e., $\Delta_d = d - d'$. Intuitively, the larger the value of Δ_d, the more complex the decision is, given the temptation to consume a closer distractor than to consume the highest value distant goal. Simply put, the high value of Δ_d signifies the high conflict between consuming the preferred object with the longer distance d or the distractor with the shorter distance d'.

The experiment design is illustrated in Figure 2. It is worth noting that we only examine the cases when $\Delta_d > 0$ as when $d > d'$. Take Figure 2a as an example of how the setup works. Here the distractor is the "orange" object while the highest value goal is "purple". The distance from the agent's location to its goal is $d = 5$ and to the distractor object is $d' = 4$, and hence $\Delta_d = (d - d') = 1$. This is a simple environment since the cost to reach the highest value goal and the distractor is nearly equal (and thus, preferring the highest value goal over the distractor is a simple choice). In contrast, as exemplified in Figure 2d, with $\Delta_d = 4$, the choice is more complex, since preferring the highest value goal ("blue" object) is more costly than consuming the distractor ("yellow" object) and as a result, the agent may be attracted to the closer object (even if the reward is lower).

Model Parameters

The IBL agent's parameters are $\sigma = 0.25$ and $d = 0.5$, default parameters that come from the ACT-R architecture (Anderson & Lebiere, 2014). For the Random agent, we consider $\pi_k \sim Dir(\alpha = 3)$. Regarding the parameters of Q and SARSA, we set the discount factor $\gamma = 0.99$ and the learning rate $\alpha = 0.1$.

Independent Variables

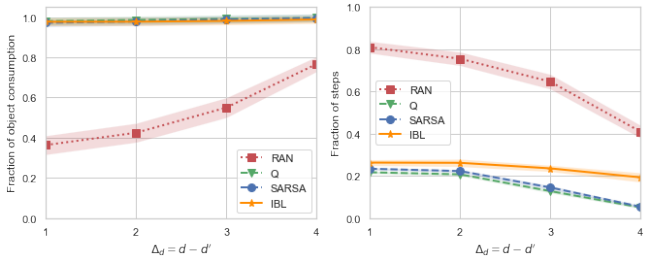
For simplicity, in this experiment we deal only with the trade-off between one preferred goal and one distractor, where the distance between an agent and its preferred goal is fixed to $d = 5$. Hence, to manipulate decision complexity, we only vary the distance from the agent to the distractor ($d' = 1 \ldots 4$). We examined four levels of decision complexity ($\Delta_d = 1..4$) and four types of agents (IBL, Q, SARSA, and Random). For each of the four levels of complexity, we ran 100 agents of each type, that is, for each value of Δ_d, 100 different gridworlds were generated. In each gridworld, the agents had 500 learning episodes.

Evaluation Metrics

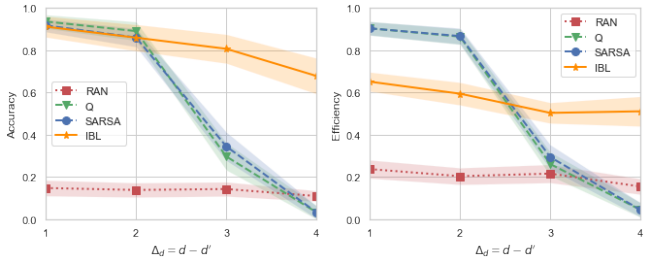
For each model we calculated the following measures: (1) *Fraction of object consumption*: the proportion of episodes (out of 500) in which the agent reaches one of the four objects (i.e., rather than wandering around and reaching the limit of steps without consuming any object); (2) *Fraction of steps*: the average ratio (across 500 episodes) between the number of steps for consuming any of the four objects and the maximum number of steps; (3) *Accuracy*: the proportion of episodes (out of 500) wherein the agent accomplishes the task (i.e. successfully consumes the highest value goal); and (4) *Efficiency*: the ratio between the reward from consuming an object and the movement cost (i.e., the multiplication of the penalty for each step and the number of steps taken) across the 500 episodes. The efficient values were normalized to the range in $[0, 1]$ using min-max normalization, i.e. $(value - min)/(max - min)$.

Results

We have analyzed the performance of the four types of agents, namely IBL, Q, SARSA, and Random, with respect to complexity ($\Delta_d = 1 \ldots 4$).



(a) Fraction of obj. consumption (b) Fraction of steps



(c) Accuracy (d) Efficiency

Figure 3: Performance of the agents in the task when varying the degree of complexity $\Delta_d = 1 \ldots 4$ (X-axis).

Fraction of Object Consumption

Figure 3a shows that the Fraction of object consumption for IBL and the two RL agents (Q and SARSA) reaching *any* of the objects is approximately equal to 1 regardless of complexity. Unsurprisingly, the Random agent performed considerably worse than the other agents, but their capability to consume an object in less than 31 steps increased with complexity. This can be explained by the environment design that the more challenging the environment is, the closer the agent is to a distractor. Hence, it is evidently easier for a Random agent to bump into an object when Δ_d increases.

Fraction of Steps

Figure 3b shows the Fraction of steps, suggesting that the IBL agent took about the same steps to get an object regardless of the level of complexity, while the Q and SARSA agents took slightly less steps with larger complexity. We also observe that the Random agent required the most steps to find an object, but the fraction decreases as complexity increases. Again, the most likely explanation is that the agents tend to consume the closer distractors.

Accuracy

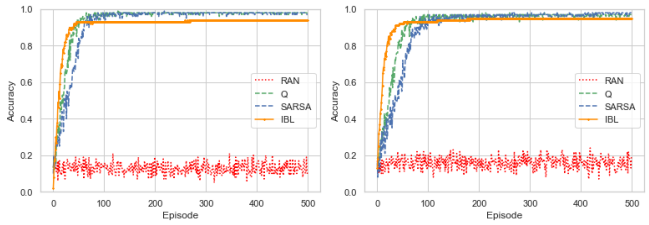
Figure 3c demonstrates that Accuracy of the RL agents and the IBL agent are comparable in simple environments ($\Delta_d = 1$ and 2). However, when complexity increases ($\Delta_d = 3$ and 4), IBL exhibits only a light drop in accuracy, while the Q and SARSA agents dropped accuracy significantly, reaching close to random accuracy with the highest complexity. More concretely, the IBL agents with an approximate 70% overall success rate by far surpassed the Q and SARSA agents whose the fraction of successful episodes was less than 5% over all 500 episodes. With respect to the Random agent, its curve is flat and nearly constant at about 0.18 over the values of Δ_d, signifying that its performance is independent of complexity due to its random characteristic.

Efficiency

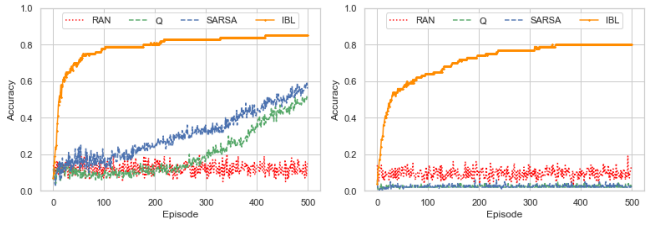
Figure 3d reveals that the Q and SARSA agents are the most efficient agents in the simple decisions ($\Delta_d \leq 2$), followed by IBL and then by Random. The higher value of the ratio between the benefit (i.e. the consumption reward) and the movement cost (i.e. the penalty for each step that the agent takes $\times$ the number of steps) indicates that the RL agents are able to obtain an object having the highest reward within a limited number of steps, when the decision is simple. Conversely, in complex decisions ($\Delta_d > 2$) the results show that the IBL agent is the most efficient, followed by the RL agents and the Random agent. The Efficiency together with the Accuracy results suggest that the RL agents are "distracted" by the closer objects and end up consuming the closer objects rather than affording the costs of searching for the highest value object. As a result, they got a significantly small amount of reward.

Learning Curves of Accuracy

To start to explain the observations above regarding the accuracy of the models, we analyzed the average Accuracy for each type of agent over the course of 500 episodes. This analysis would help observe how the accuracy developed within each level of complexity. Figure 4 demonstrates that the IBL agent learned slightly faster than the RL agents even in lower levels of complexity. The learning speed of the IBL models decrease with increased complexity, but the difference between IBL and the RL agents is larger in the complex settings ($\Delta_d > 2$). The Random agent does not learn.



(a) $\Delta_d = d - d' = 1$ (b) $\Delta_d = d - d' = 2$



(c) $\Delta_d = d - d' = 3$ (d) $\Delta_d = d - d' = 4$

Figure 4: Learning curves of the agents over 500 episodes for each level of complexity Δ_d.

Specifically, in the most complex decision environment ($\Delta_d = 4$), the average Accuracy achieved by the IBL agent

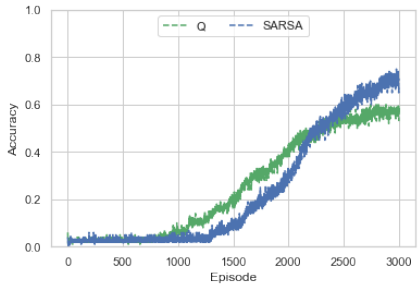


Figure 5: Learning curves of the Q and SARSA agents over 3000 episodes when $\Delta_d = 4$.

was over 0.6 just after 100 episodes, while the average Accuracy of the RL agents was nearly zero. To investigate this further, we ran the RL models for 3000 episodes under the highest level of complexity ($\Delta_d = 4$). The results shown in Figure 5 demonstrate that the Q and SARSA agents have a low start but learn to be more accurate in the highest complexity levels, after extended practice. We speculate that the one-step update of state-action values in the RL algorithms may prevent them from learning faster and determining the value of the various objects within 500 attempts. In contrast, the IBL model uses all the past instances in the blending mechanism (but these instances are decayed to different degrees as in Equation 1). This aggregation of more experiences may help to evaluate the decision tradeoffs more accurately, resulting in faster and more successful weigh of the costs and benefits in the decisions.

Conclusions

We investigated the performance of an IBL agent, two RL agents (Q-learning and SARSA), and a Random agent, while performing a navigation task under uncertainty and under increasing decision complexity. The decision complexity is formalized as the tradeoff between the objects' rewards and the associated movement costs. To select the object to consume in the presence of uncertainty, the agents must evaluate the expected reward of the object and the steps needed to reach it (costs), from experiential learning.

Experimental results revealed that the Accuracy and Efficiency of the two RL agents were not robust to increased levels of decision complexity, while the IBL cognitive model was more resilient to higher levels of complexity. The explanation is that the one-step update of state-action values in the RL agents results in these agents getting "distracted" by near objects, which are consumed even when they are of lower value. Thus, as the difficulty of the decisions increases the Accuracy and Efficiency of the RL agents decrease. The IBL agent is less efficient than the RL agents under low levels of complexity but under higher complexity levels it learns to consume the higher value objects even when it takes more steps to reach those objects.

Acknowledgments

This research is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), award number: FP00002636. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

Anderson, J. R., & Lebiere, C. J. (2014). *The atomic components of thought*. Psychology Press.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in cognitive sciences*.

Chelian, S. E., Paik, J., Pirolli, P., Lebiere, C., & Bhattacharyya, R. (2015). Reinforcement learning and instance-based learning approaches to modeling human decision making in a prognostic foraging task. In *2015 joint ieee international conference on development and learning and epigenetic robotics* (pp. 116–122).

Fu, W.-T., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of experimental psychology: General*, *135*(2), 184.

Gershman, S. J., & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual review of psychology*, *68*, 101–128.

Gonzalez, C. (2013). The boundaries of instance-based learning theory for explaining decisions from experience. In *Progress in brain research* (Vol. 202, pp. 73–98). Elsevier.

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., & Dutt, V. (2011). Instance-based learning: Integrating decisions from experience in sampling and repeated choice paradigms. *Psychological Review*, *118*(4), 523–51.

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

Gureckis, T. M., & Love, B. C. (2009). Short-term gains, long-term pains: How cues about state aid learning in dynamic environments. *Cognition*, *113*(3), 293–313.

Hertwig, R. (2015). Decisions from experience. *The Wiley Blackwell handbook of judgment and decision making*, *1*, 240–267.

Kelly, M. A., & West, R. L. (2013). Decision making in a dynamically structured holographic memory model: Learning from delayed feedback. In *Proceedings of the international conference on cognitive modelling.*

Kittel, C. (2004). *Elementary statistical physics*. Courier Corporation.

Lebiere, C. (1999). Blending: An act-r mechanism for aggregate retrievals. In *Proceedings of the sixth annual act-r workshop.*

Lejarraga, T., Dutt, V., & Gonzalez, C. (2012). Instance-based learning: A general model of repeated binary choice. *Journal of Behavioral Decision Making*, *25*(2), 143–153.

Mehlhorn, K., Newell, B. R., Todd, P. M., Lee, M. D., Morgan, K., Braithwaite, V. A., ... Gonzalez, C. (2015). Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision*, *2*(3), 191.

Nguyen, T. N., & Gonzalez, C. (2020). Cognitive machine theory of mind. In *Proceedings of the 42nd annual meeting of the cognitive science society (cogsci 2020).*

Reitter, D., & Lebiere, C. (2010). A cognitive model of spatial path-planning. *Computational and Mathematical Organization Theory*, *16*(3), 220–245.

Simon, D. A., & Daw, N. D. (2011). Environmental statistics and the trade-off between model-based and td learning in humans. In *Advances in neural information processing systems* (pp. 127–135).

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning* (Vol. 2) (No. 4). MIT press Cambridge.

Walsh, M. M., & Anderson, J. R. (2011). Learning from delayed feedback: neural responses in temporal credit assignment. *Cognitive, Affective, & Behavioral Neuroscience*, *11*(2), 131–143.