Computational Reinforcement Learning

Todd M. Gureckis[a,*], Bradley C. Love[b]

[a] *Department of Psychology, New York University, 6 Washington Place, New York, NY 10003*
[b] *Cognitive, Perceptual, and Brain Sciences, University College London, 26 Bedford Way, London, UK WC1H 0AP*

Abstract

Reinforcement learning (RL) refers to the scientific study of how animals and machines adapt their behavior in order to maximize reward. The history of RL research can be traced to early work in psychology on instrumental learning behavior. However, the modern field of RL is a highly interdisciplinary area which lies that the intersection of ideas in computer science, machine learning, psychology, and neuroscience. This chapter summarizes the key mathematical ideas underlying this field including the exploration/exploitation dilemma, temporal-difference (TD) learning, Q-learning, and model-based versus model-free learning. In addition, a broad survey of open questions in psychology and neuroscience are reviewed.

Keywords: reinforcement learning, explore/exploit dilemma, dynamic decision making

1. Introduction

There are few general laws of behavior, but one may be that humans and other animals tend to repeat behaviors which have led to positive outcomes in the past and avoid those associated with punishment or pain. Such tendencies are on display in the behavior of young children who learn to avoid touching hot stoves following a painful burn, but behave in school when rewarded with toys. This basic principle exerts such a powerful influence on behavior, it manifests throughout our culture and laws. Behaviors society wants to discourage are tied to punishment (e.g., prison time, fines, consumption taxes) while behaviors society condones are tied to positive outcomes (e.g., tax credits for fuel efficient cars).

The scientific study of how animals use experience to adapt their behavior in order maximize rewards is known as *reinforcement learning* (RL). RL differs from other types

* Corresponding author.
Email addresses: todd.gureckis@nyu.edu (Todd M. Gureckis), b.love@ucl.ac.uk (Bradley C. Love)

of learning behavior of interest to psychologists (e.g., unsupervised learning, supervised learning) since it deals with learning from feedback that is largely evaluative rather than corrective. A restaurant diner doesn't necessarily learn that eating at a particular business is "wrong," simply that the experience was less than exquisite. domain for studying how people adapt their behavior based on experience.

The history of RL can be traced to early work in behavioral psychology (Thorndike, 1911; Skinner, 1938). However, the modern field of RL is a highly interdisciplinary area at the crossroads of computer science, machine learning, psychology, and neuroscience. In particular, contemporary research on RL is characterized by detailed behavioral models which make predictions across a wide range of circumstances, as well as neuroscience findings which have linked aspects of these models to particular neural substrates. In many ways, RL today stands as one of the major triumphs of cognitive science in that it offers an integrated theory of behavior at the computational, algorithmic, and implementational (i.e., neural) levels (Marr, 1982).

The purpose of this chapter is to provide a general overview of RL and to illustrate how this approach is used to understand decision making and learning behavior in humans and other animals. We begin in section 2 with a historical perspective, tracing the roots of contemporary reinforcement learning research to early work on learning behavior in animals. In section 3 we introduce a general computational formalism for understanding RL, based largely on the work of Sutton and Barto (1998). Along the way we discuss some of the critical aspects of RL including the tradeoff between exploration and exploitation, credit assignment, and error-driven learning. Section 4 focuses on the neural basis of RL. Section 5 describes a variety of contemporary research questions as well as open areas for future investigation.

2. A Historical Perspective

The early roots of RL research can be traced to the work of Thorndike (1911). Thorndike studied learning behavior in animals, most notably cats. In perhaps his most well-known experiment, he placed a cat into a specially designed "puzzle box" which could be opened from the inside via various latches, strings, or other mechanisms (see Figure 1). The cat was encouraged to escape the box by the presentation of food on the outside of the box. The key issue for Thorndike was how long it would take the cat to escape.

Once placed in this situation, the cat usually began experimenting with different ways to escape (pressing levers, pulling cords, pawing at the door). After some time and effort, they would stumble on the particular mechanism that opened the cage. This process was repeated across a number of trials, each time recording the total time until the cat escaped. Over time, the cats would learn that particular actions (e.g., pressing a lever, or pulling a cord) would lead to the desired outcome (food) and engage in this behavior more rapidly, avoiding actions which had previous been unsuccessful at opening the box. In some sense,

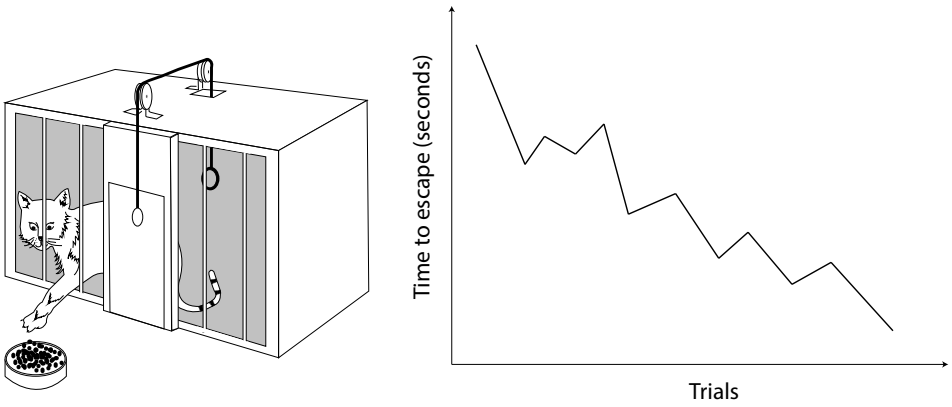

Figure 1: *Left*: An illustration of Thorndike's puzzle box experiments. *Right*: The time recorded to escape the box is reduced over repeated trials as the cat becomes more efficient at selecting the actions which lead to escape.

the correct behavior to escape was *selected* out of the full behavioral repertoire of the cat while others were eliminated.

There are several key features of Thorndike's experiments which are central to RL which we discuss throughout the chapter. The first is that successful escape from the puzzle box requires sufficient *exploration* of alternative actions. If the cat repeatedly tries the same unsuccessful action escape is hopeless. On the other hand, after the cat escapes, it becomes better to engage in less exploration and to *exploit* previously successful actions so that escape is faster. Of course, dropping off exploration too quickly could cause the cat to miss alternative means of escape that are less effortful or time consuming. Thus, effective behavior depends on a delicate balance between exploration and exploitation of options (see Section 3.7 and 5.3).

A second feature of Thorndike's experiments is that the goal (escaping the box to get the food) is a complex, multi-action sequence. When the cat first succeeds in escaping, this raises the question of *credit assignment*: which of the multiple actions the animal might have tried were responsible for the escape? This problem can be especially difficult in a sequential decision making setting because an action taken at one point in time may only have the desired effect some time later. For example, perhaps the cat pulls a string at one point in time, presses a lever, and then pulls the string a bit harder which opens the box. On the next trial should the lever be pressed or should the string be pulled? Contemporary research in computational RL provides a computational solution to how agents might solve this *credit assignment problem* (see Section 3).

Reinforcement learning is also closely related to the idea of *instrumental conditioning* and the *operant conditioning* paradigm pioneered by psychologists in the behaviorist tradition (e.g., Skinner, 1938). Operant conditioning represents a refinement of the basic

experiment design utilized by Thorndike. In a typical experiment, a rat might be placed in an isolated chamber with a lever which can be depressed. The experimenter records the frequency by which the rat presses the lever and at various points in time provides reward (e.g., food pellets) or punishment (e.g., electric shocks). The main question of interest is how the voluntary behavior of the animal is modified by various reward or punishment schedules. There continue to be a variety of theories of operant conditioning, but all share a basic view that through learning, associations are strengthened (or weakened in the case of punishment) between elements which follow one another in time.

Another major historical influence on modern RL was the work of Tolman (1948). Prior to Tolman's work, psychologists largely viewed animal behavior as a product of associative stimulus-response (S-R) learning and chaining of basic S-R behaviors. Tolman (1948) argued that many aspects of rodent behavior seemed to contradict this basic model. For example, Tolman showed that in maze tasks, rats could quickly re-route a path in a maze around a trained route which was experimentally blocked with an obstacle. This, he argued, could not be accomplished on the basis of pure stimulus-response learning since the relevant stimulus-response pairings for the new route were never directly experienced by the rat. Instead, it appeared that rats use a "cognitive map" of the maze which allowed them to flexibly plan goal-directed sequences of behavior. While Tolman's work is often seen as antithetical to basic principles of conditioning, modern RL approaches have directly explored the distinction between more reactive, associative forms of reinforcement learning and more cognitive, planning-based approaches (a distinction referred to as model-based versus model-free RL, see Section 5.1). Box 1 describes common experimental approaches to studying RL.

3. A Computational Perspective on Reinforcement Learning

The ability to adaptively make decisions which avoid punishment and maximize rewards is a core feature of intelligent behavior. Computational RL (CRL) is a theoretical framework for understanding how agents (both natural or artificial) might learn to make such decisions. CRL is not simply a description of how agents decide and learn, but offers insight into the overall *function* or *objective* of adaptive decision making. This is sometimes known as a "rational analysis" of behavior since it seeks to understand the purpose of some behavior independent of the exact mechanism by which it is accomplished (Marr, 1982; Anderson, 1990). Once the objective of learning has been made clear, CRL goes on to posit a family of related learning algorithms or mechanisms all designed to allow an embodied, autonomous agent to control their environment in effective ways. The core methods in the field were developed jointly by researchers in both computer science and psychology (Sutton and Barto, 1981; Sutton, 1988; Sutton and Barto, 1998)[1] as well as operations research (Bertsekas and Tsitsiklis, 1996).

[1] Sutton & Barto (1998) in particular provides a very clear technical introduction to the area.

3.1. The goal

All else being equal, it seems likely that animals seek to avoid punishment and maximize reward through their choices and actions (e.g., Thorndike's "Law of Effect"). This basic idea is reflected in the first key assumption in CRL: that agents strive to maximize the long-term reward they experience from the environment. Formally, if the reward experienced at time t is r_t, then the goal of learning and decision making is to maximize the expectation of reward over the long run future, i.e.,

$$E[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}]. \tag{1}$$

The term γ in Equation 1 represents a discount factor which gives greater weight to rewards that are experienced sooner rather than later. This property is desirable for mathematical reasons (making the sum finite), but also because humans and other animals seem to have similar preferences for immediate over delayed rewards (Myerson and Green, 1995; Frederick et al., 2002).

3.2. Defining the decision environment: Basic definitions

Most applications of CRL, particularly applied to human and animal behavior, assume that the world can be viewed as a particular type of dynamic process know as a Markov decision process or MDP. The MDP assumption simply asserts that the world is composed of a set of finite states ($\mathcal{S}$), actions ($\mathcal{A}$), a set of transition probabilities ($\mathcal{T}$), and rewards ($\mathcal{R}$).

The *states* of a MDP refer to different distinct situations the agent might be in. The notion of a "state" in RL is sufficiently general to cover many different types of situations. For example, being in a particular location in a maze facing a particular direction might be one state. States might also involve elements internal to the agent like "being hungry" or, for a robot, "having a low battery" (see Section 5.2 for a discussion of the notion of states in the context of human learning).

Actions are the decision options that are available to the agent across all different states of the world. In some MDP problems, all actions are available in all states while in others, different subsets of actions might be available depending on the state. For example, if facing a wall at the end of a dead-end hallway, and agent's available actions are practically limited to those that allow it to turn around. Like the definition of states, actions can be internal to the agent (e.g., encode the currently attended object in memory).

Transition probabilities determine the dynamics of the environment. A transition probability can be defined as the probability of a new state s' on the next time step $(t+1)$ given the current state is s and the agent selects action a:

$$P(s_{t+1} = s' | s_t = s, a_t = a) \tag{2}$$

Note that the next state depends only on the current state and action and not on the full history of actions up to that point. This is known as the Markov assumption. While this

assumption may be violated in the real world, for many situations this assumption appears reasonable and it greatly simplifies the mathematics involved.

The $\mathcal{R}$ determines how rewards or punishments are distributed in the environment. In a psychology experiment, the experimenter might manipulate how reward is provided to the learner to alter their behavior. Similarly, roboticists who use RL to train autonomous systems often must adapt the reward function provided to the robot so the system meets certain engineering objectives (see Section 5.4 for a discussion of rewards in the context of human learning). In a MDP, rewards can be probabilistic (like the reward associated with buying a lottery ticket), thus it often makes sense to talk in terms of averages or expected values of rewards. In particular, we could summarize the expected value of rewards on trial $t + 1$ as

$$E[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s'] . \tag{3}$$

The function $\mathcal{R}$ assigns the value of reward received for each possible state transition. Together, these four elements ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{R}$) completely define the decision problem (or MDP) facing the agent. Any particular task or environment can be defined by providing particular parameters or numbers for these four quantities (usually each can be expressed as a function or a matrix). Critically, the description of a particular task as an MDP does not assume any particular agent has complete knowledge about these four quantities (in fact in most realistic settings it would be impossible for an agent to completely know these aspects of the world). What is important in an MDP is simply defining the generic nature of the decision making problem facing the agent. We later will discuss how CRL algorithms provide guidance on *how* agents to learn to make effective decisions in such an environment given various amounts of knowledge about the world and experience.

3.3. Assigning value to states and actions

Given the description of the environment as an MDP, it is clear that states and actions in the world that lead to greater expected reward have greater value to the agent. However, it is not only the states and actions *immediately* resulting in reward that may have value to an agent. For example, an action that takes the agent closer to a rewarding state can also be valuable (in the long run) by virtue of what it means for the agent's future prospects.

One clear example of this is *second-order conditioning* (Rescorla, 1980). In second-order conditioning, a stimulus such as a light is first paired with a negative outcome like a shock. Next, a second stimulus such as a tone is paired with the light (but without the shock). At test, animals show anticipation of the shock following the presentation of the tone even though this item was never directly paired with the negative outcome. In other words, the tone appears to be a proxy for the negative outcome because tones tend to beget lights, and lights tend to beget shocks.

One way to quantify this "proxy" value is as the expected sum of future rewards available from each state, denoted $V(s_t)$:

$$V(s_t = s) = E[\sum_{k=t}^{\infty} \gamma^k r_{t+k+1} | s_t = s] \tag{4}$$

In the second-order conditioning example just described, the long-term expected value at the onset of the light is largely negative, but so is the long-term expected value at the onset of the tone since they both ultimately will lead to shock. However, the state tied to the tone would have a slightly higher value since the punishment is delayed further in the future.

This notion of estimating the "proxy" value of state or actions that later lead to reward is what allows RL to offer a solution to the *credit assignment* problem described above. The value of actions that in turn lead to others actions that provide reward is captured by the evaluations on long-term rather than immediate prospects. Of course, the valuation of particular states is very sensitive to the agent's discounting parameter, γ. If γ is very small, the agent cares only about immediate reward (i.e., is myopic), and thus actions or states which result in direct reward are highly valued. If γ is larger, then future prospects are taken into account.

One of the most important and interesting aspects of CRL is that the value of each state in Equation 4 can be defined recursively in terms of the value of other states:

$$\begin{aligned} V(s_t = s) &= E[\textstyle\sum_{k=t}^{\infty} \gamma^k r_{t+k+1} | s_t = s] \\ &= E[r_{t+1} + \gamma \textstyle\sum_{k=t}^{\infty} \gamma^k r_{t+k+2} | s_t = s] \\ &= E[r_{t+1} + \gamma V(s_{t+1}) | s_t = s]. \end{aligned} \tag{5}$$

In other words, the value of a state s_t is the expectation of the reward experienced leaving that state (r_{t+1}) plus a discounted estimate of the future reward available from the possible successor states, s_{t+1}. Making the expectation in Equation 5 more explicit:

$$V(s_t = s) = \sum_a \pi(s, a) \sum_{s'} P^a_{ss'} [R^a_{ss'} + \gamma V(s')] \tag{6}$$

where $\pi(s, a)$ is the agent's current probability of choosing action a in state s, $P^a_{ss'}$ is the probability of transitioning from state s to s' given action a, and $R^a_{ss'}$ is the reward expected from that same action (using the notation developed by Sutton & Barto, 1998). Finally, $V(s')$ is the estimated value of future reward for the next state (i.e., Equation 4). This approach to averaging over possible outcomes weighted by their probability of occurrence is central to almost all work on judgement and decision making (e.g., Bernoulli, 1954).

One helpful way to think about this relationship is as a tree (see Figure 2). At the root of the tree is the agent's current state (s). The value of that state will depend first on which action the agent takes (a) which is represented by the first set of branches in the figure. Which action is selected in turn will determine which of a variety of different rewards might be experienced (r) while transitioning to the new state s'. The value of the current

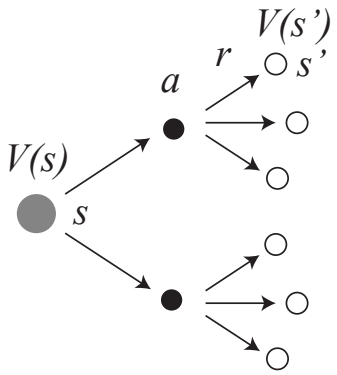

Figure 2: The value of the current state s can be estimated by "looking forward" from the current state to the expected reward r and the value of the possible successor states s', averaged over all possible actions and state transitions.

state $V(s)$ is thus a weighted average of the value of all possible successor states based on both the agent's decision making, as well as the dynamics of the environment. This is similar to analyses of games like tic-tac-toe where different actions lead to different game states and thus the value of any particular game state will depend on the available actions and states going forward. The recursive relationship between successive state is called the *Bellman equation* and is an important feature of MDPs as we will see below.

3.4. Making good decisions

One way an agent could maximize reward would be to learn a set rules about how to behave in each state. Often this set of rules is known as a *policy*, which tells the agent which action to select in each state in order to maximize the expectation of long term reward (Equation 1). We first hinted at the idea of a policy in Equation 6 when we needed to consider the agent's probability of selecting each possible action ($\pi(s, a)$). The complete policy for how to respond in each possible state of the environment is denoted simply as π.

Ideally, the agent would like to learn the *optimal policy*, which is the one that returns the most reward over the long term across all possible states of the environment. There are many different methods for learning optimal policies for MDPs including dynamic programming and Monte-Carlo methods (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998). However some of these methods are less computationally practical for biological agents with finite resources or incomplete of knowledge about the environment.

However, two particular methods of learning optimal (or near-optimal) policies have had a extremely important influence on research on learning and decision making in humans and other animals: temporal difference (TD) learning and Q-learning. In the following sections, we describe the basic idea behind each of these algorithms. Later we

will discuss how the features of these algorithms are related to the learning behavior of animals.

However, first it is worth pointing out how estimating the value of each state can help the agent make decisions. Since the values for each state represent the expected sum of future rewards available from that state (Equation 4), it means the agent is maximizing reward (given the current policy) if it chooses actions which have the highest probability of transitioning to states with high values. For example, we can talk about the long-term value of making a decision in a particular situation (more specifically a state-action pair) as:

$$Q(s_t = s, a_t = a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')] \tag{7}$$

which is just a simplification of Equation 6 which doesn't average over different possible actions. A good strategy for the agent would be to choose the action in the current state s which maximizes the value of $Q(s, a)$. This is trivially true since $Q(s, a)$ measures the expected long-run reward available from taking that action. To maximize reward over the long-term the agent just needs to choose actions which maximize this quantity in each state. However, agents must first *learn* good estimates of $V(s)$ or $Q(s, a)$, which is what TD and Q-learning methods accomplish.

3.5. Temporal difference (TD) learning

As the previous section described, an agent who has learned the values for each state in the environment can behave optimally simply by choosing actions in each state which maximize the estimated long-term reward. However, the main issue is how to efficiently learn these values. One possibly naïve way to do this would be to follow a policy, π, keeping track of any rewards experienced along the way and at some point updating the value of $V(s_t)$ based on that experience. The downside is that it will take a lot of experience before the agent knows anything about the value of any particular state (conditioned on the policy).

However, a more efficient, online, incremental learning rule that estimates $V^\pi(s_t)$ can be derived by observing the relationship described in Equation 5. The important insight of this equation is that the value of $V^\pi(s_t)$ depends on the average value of the next state $V^\pi(s_{t+1})$. This fact can be used to "bootstrap" estimates of $V^\pi(s_t)$ more quickly.

We will illustrate the basic idea with an example, then discuss the mathematics. In Figure 2, we showed how the value of a given state in an MDP is determined by the "tree" of successor outcomes available from that state. In Figure 3, we image that the agent begins in state s and has already estimated its long-term value to be 0.5, then selects action a. As a result of that selection, the agent receives a reward equal to $r = 1.0$ and transitions to a new state, s' which it has previously estimated to have value of 1.2. Assuming the discount parameter, γ is set to 0.9, this would appear to be a better outcome than the agent expected. In particular, the agent expects to on average receive 0.5 reward units

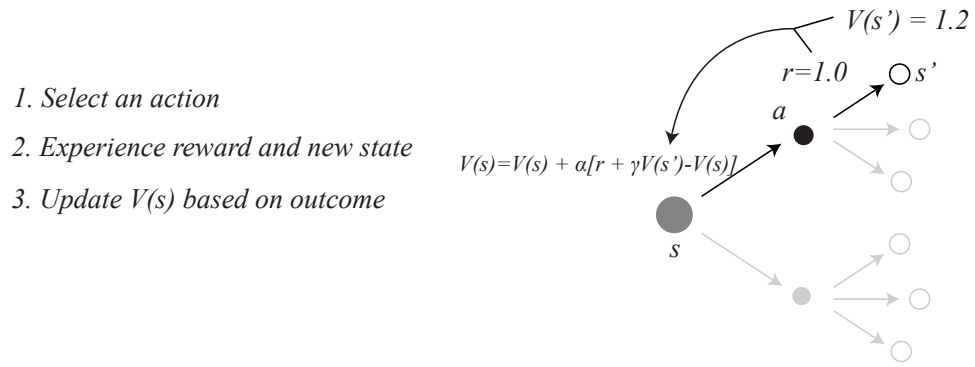

Figure 3: The key steps in the temporal-different (TD) update. First, an action is selected from the current state based on the agent's current policy. Next, the reward is experienced along with information about the successor state. The agent can lookup its estimate of the value of this state from memory or assign a default value if the successor state had never been visited. Next the value of the original state is updated in light of the experienced outcome. In this way estimates of the value of each state can be "bootstrapped." When the agent's estimate of $V(s)$ is accurate the error in expected reward and experienced reward will drop to zero on average. Learning is based on a *temporal difference* in what was expected and what was actually experienced.

going forward from state s, but on this choice estimates and receives $r + \gamma V(s') = 1.0 + 0.9 \cdot 1.2 = 2.08$.

This discrepancy suggests that agent might be wise to revise its estimate of $V(s)$ to be higher (this is a better than expected state). One approach could be to simply replace the agent's estimate of $V(s)$ so that it equals 2.08. However, remember that $V(s)$ represents the long-term reward available from a state averaged over all possible actions, rewards, and successor states. Thus, it would be to drastic to replace the old value completely with a single experienced outcome. Instead, the agent could simply move its estimate of $V(s)$ a step *in the direction* of $r + \gamma V(s')$. As long as the step size isn't too large, it can be shown that this will allow the value of $V(s)$ to eventually converge on the true long-term estimate. On some trials it might move a little too high, on some trials a little to low, but in the limit should converge towards the mean of the different experiences.

Once the value of $V(s)$ is updated, the current state is set to s' and the process continues. The important point is that the value of $V(s)$ can be *bootstrapped* based on the outcome of individual choices. After the agent makes a choice, it compares the value of the reward it experienced to the long-term estimate going forward and adjusts that estimate, $V(s)$, up or down accordingly.

Formally, the error in the agent's current estimate of $V(s)$ can be denoted using an intermediate variable called the *prediction error*, commonly denoted

$$\delta = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{8}$$

which measures the difference between the experienced and estimated value of current state. Prediction errors can be positive (when events worked out better than expected) or negative (when events work out worse then expected). When this value drops to zero it means that the current value estimates are all consistent with one another and thus represent the *true* values. In the section on the neural underpinings of RL (Section 4) we will describe how neurons encoding prediction errors have been identified in the brains of animals.

An incremental rule for adjusting $V(s_t)$ can be written using this prediction error:

$$\begin{aligned} V(s_t) &= V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \\ &= V(s_t) + \alpha\delta. \end{aligned} \tag{9}$$

in this equation α is known as the learning rate and represents the "step size" of the update to V(s). In other words, V(s) is moved slightly in the direction of the prediction error (assuming $\alpha < 1.0$)[2]. This simple, incremental method of learning the values of states is known as temporal difference learning (or TD). The name is largely descriptive: estimates of the long-term value of a state are based on differences between what was expected and what was experienced in time.

3.6. Q-learning

Q-learning is a modification of the basic temporal-difference[3] algorithm which learns the value of state-action pairs, $Q(s, a)$ directly (rather than estimating the value of individual states, $V(s)$). This is often a more useful quantity to estimate since we are often interested in the value of particular choices. As mentioned above, given direct $Q(s, a)$ estimates computing a policy is simple: choose the action in the current state associated with the largest value of $Q(s, a)$. As a result, Q-learning obviates the need for a separate representation of the policy. From the perspective of a biological agent, learning and choice are made more compatible with few demands of extra processing (e.g., to compute $V(s)$ estimates into $Q(s, a)$ estimate via Equation 7). In effect, the learned Q-values (i.e., $Q(s, a)$) tell the agent directly what choice to make.

Following Equation 9, an incremental update rule for $Q(s, a)$ can be defined:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\delta. \tag{10}$$

where δ takes a slightly different form than in standard TD learning:

[2]This discussion implicitly assumes the learning rate, α, is constant, however, the learning rate might be adjusted based on experience in the task (e.g., dropping toward zero over time) or based on the overall volatility of the environment (see Sutton & Barto, 1998, chapter 2 for a further discussion). This issue is also of interest in the psychology and neuroscience literatures (Behrens et al., 2007; Krugel et al., 2009; Nassar and Gold, 2013).

[3]In fact, both the TD algorithm described in Section 3.5 and Q-learning are considered "temporal-difference methods" although the TD algorithm shares the name with this more general class.

$$\delta = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \tag{11}$$

Q-learning is considered an "off-policy" learning algorithm because the prediction error for the current choice assumes that the agent will choose the best action on the next state (i.e., the $\max_a Q(s_{t+1}, a)$ in Equation 11) rather than follow their current policy (which might include the possibility of choosing to explore a different action). Q-learning is particularly important in computer science applications of RL because it can be proven to converge on the optimal policy for the particular MDP given that all state-action pairs are visited infinitely often (Watkins, 1989)[4]. In addition, it has shown considerable success as a model of how humans learn in dynamic decision making tasks where past actions influence future rewards (e.g., Gureckis and Love, 2009a,b; Otto et al., 2009).

3.7. Balancing exploration and exploitation - a computational view

In Thorndike's puzzle-box experiments (see Section 2), we noted that the cat needs to explore various actions to learn which allow escape from the box. However, after a bit of learning, it becomes better to "exploit" action sequences that are known to be effective.

An analogous situation arises in TD and Q-learning algorithms just described. Choosing the action with the highest value of $Q(s, a)$ in each state is will lead to optimal long-term decision making. However, this assumes that the current estimates of $Q(s, a)$ are already accurate. At the start of learning in a novel task or environment this is unlikely to be the case. Instead it will take many updates of Equation 10 in each state to ensure that the estimated values have converged.

Thus, agents learning via TD and Q-learning must also balance the need to explore (in order to learn) and exploit (in order to actually maximize reward). Early in a task, the agent shouldn't necessarily trust the current estimates of $Q(s, a)$ too much and should sometimes choose actions which actually have lower estimated values. This is because those estimates may be incorrect and, with experience, would be adjusted upwards or downwards.

One way to balance these competing concerns would be to choose the action associated with the highest value of $Q(s, a)$ most of the time (exploit), but some fraction of the time to choose an action randomly (explore). As long as the probability of exploring is non-zero but not too large, this strategy can help the agent learn the true Q-values for any particular environment. This explore/exploit strategy is often known as the ϵ-greedy algorithm and entails choosing the option associated with the highest value of $Q(s, a)$, but choosing randomly from the available alternatives with probability ϵ. This policy also ensures that each state action pair will be visited infinitely (assuming infinite time).

[4]This might sound like an extreme demand to make on optimal learning behavior for a biological agent, but is a general requirement of all optimal convergence algorithms. In practice, algorithms such as Q-learning can converge on effective policies even in complex tasks given reasonable amounts of experience with the environment.

The downside of ϵ-greedy is that it continues to explore with probability ϵ even after the agent has lots of experience with the environment (and the Q-values may have converged to their accurate values). Another approach, known as the "softmax" strategy, explores probabilistically, but allows the current estimates of $Q(s, a)$ to influence the probability of exploration (Sutton and Barto, 1998). According to the softmax rule, the probability of choosing action a_i in any state s_t is given by:

$$P(a_i, s_t) = \frac{e^{Q(a_i, s_t) \cdot \tau}}{\sum_{j=1}^{N} e^{Q(a_j, s_t) \cdot \tau}} \tag{12}$$

where τ is a parameter which determines how closely the choice probabilities are biased in favor of the value of $Q(a_i, s_t)$ and N is the number of available actions in state s_t. In general, the probability of choosing option a_i is an increasing function of the estimated value of that action, $Q(a_i)$, relative to the other action (see also Luce, 1959). However, the τ parameter controls how deterministic responding is. When $\tau \to 0$ each option is chosen randomly (the impact of learned values is effectively eliminated). Alternatively, as $\tau \to \infty$ the model will always select the highest valued option (also known as "greedy" action selection).

Interestingly, the probability of exploration in the softmax model is sensitive to the degree of "competition" between choices. Assuming $\tau > 0$, when all possible actions have similar values of $Q(s, a)$, exploration becomes more likely. However, when one action is greatly superior, it will tend to be selected more often. In addition, in the softmax rule the value of τ might be adjusted as the experience in the task accumulates (e.g. Busemeyer and Stout, 2002) to favor exploitation over exploration (similar to simulated annealing). Section 5.3 below discusses in more detail open issues surrounding how people balance exploration and exploitation.

4. Neural correlates of RL

Interest in computational RL stems not only from its utility in computer science applications (e.g., Tesauro, 1994; Bagnell and Schneider, 2001), but the fact that human and animal brains appear to use similar types of learning algorithms. Indeed, modern RL represents a powerful theoretical framework for thinking about systems-level neuroscience. On one hand, this outcome not surprising since the pioneering work in computer science on RL was directly inspired by psychological research on basic learning processes (Sutton, 1988; Sutton and Barto, 1998). However, the discovery of extremely close correspondences between the predictions of RL algorithms and the operation of particular neural systems in mammal brains represents a major scientific advance (see Niv, 2009 for an excellent review and history of these developments).

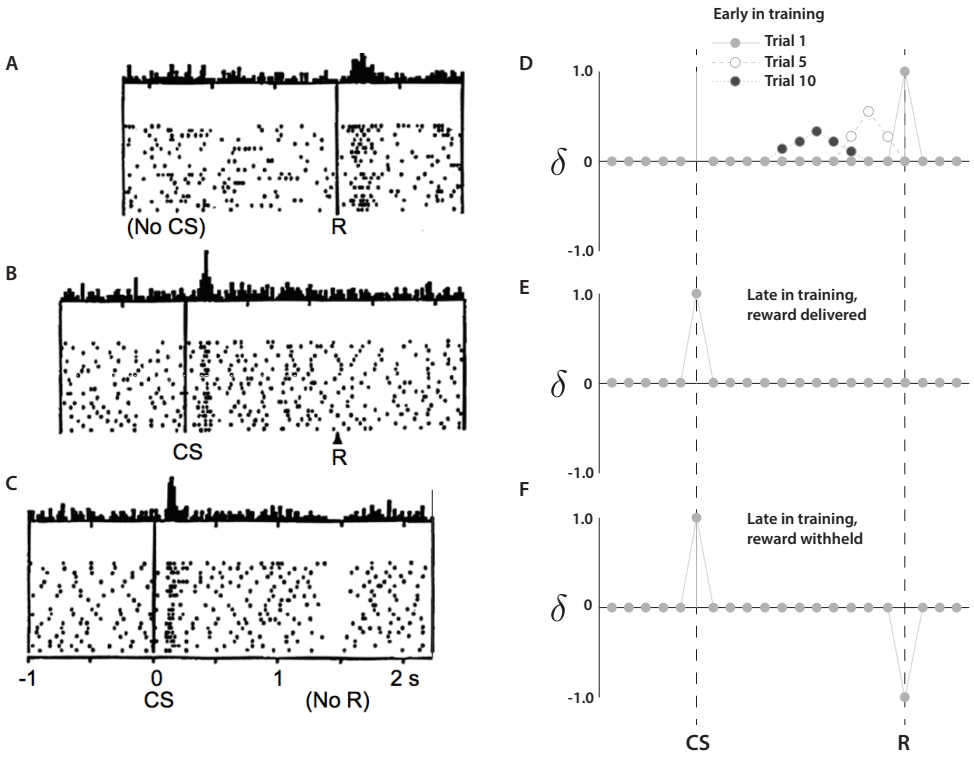

Figure 4: Figure adapted from Niv (2009). Panels A-C are taken from Schultz et al. (1997) and shows raster plots of recorded dopaminergic neurons at various stages of a classical conditioning experiment. The rows along the bottom of each panel represent individual neurons. Time within the current trial flows from left to right across the page. Black dots in a row reflect a recorded neural spike at that point in time. Along the top is a histogram of the total firing rate summed across all recorded cells (essentially the marginal distribution of the points below). Panels D-F plot the predictions of the temporal difference algorithm (Section 3.5), in particular, the prediction error term from Equation 8 at various point within a trial and as training progresses. Refer to the text in Section 4.1 for a full description.

4.1. The reward prediction error hypothesis

Perhaps the most famous discovery related to RL and the brain is the contribution RL has made to understanding the computational role of dopamine in learning. As Niv (2009) describes, early theories of dopamine suggested it encoded the reward value of particular stimuli in the environment. For example, dopamine neurons recorded within the midbrains of monkeys while they performed simple conditioning experiments showed increased firing rates immediately following the delivery of rewarding stimuli such as food (e.g., Schultz et al., 1993). However, if this rewarding stimulus was consistently preceded by a conditioned stimulus (CS, e.g., a light or a tone) then the reward related firing pattern would extinguish across trials. Instead, the neurons would begin firing upon the presentation of the CS.

For example, Figure 4, Panel A shows an example trial prior to any learning in the experiment. Shortly after the delivery of the unexpected reward (R) there is a large phasic spike in neural firing. However, panel B shows the same neurons later in a trial after a CS is repeatedly paired (following a fixed delay) to the reward. After many trial, the neurons no longer respond vigorously to the onset of the reward (R) but instead show a phasic burst of activity shortly after the presentation of the CS. Finally, panel C shows the firing pattern of the neurons late in training when the reward is unexpectedly withheld after the presentation of the CS (i.e., "No R"). Here, there is a strong phasic burst of activity following the CS, but a noticeable drop in firing when the reward is withheld.

This puzzling pattern of neural firing was ultimately deciphered using to the concept of a temporal-difference based prediction error which we introduced in Equation 8 (Montague et al., 1995, 1996; Schultz et al., 1997). A critical assumption in the neurocomputational model is that different points in time represent different states in the MDP (see Daw et al., 2006a; Ludvig et al., 2012, for a contemporary discussions of the representation of time in RL models). Note that in a given state (i.e., time point), s, prediction errors (δ) will be positive when experienced outcomes are better than expected (i.e., $\gamma V(s') + r > V(s)$) and are negative when experienced outcomes are worse than expected. Early in training, the unexpected delivery of a rewarding stimulus leads to a positive prediction error (see Figure 4, panel D). As training trials in a condition experiment are repeated, the learned values for such states increase until the prediction error drops to zero. At the same time, TD algorithms "pass back" the proxy value of one state to preceding states (assuming $\gamma > 0$). Eventually, the prediction error associated with the unexpected presentation of positively valued CS (i.e., the start of the trial) prompts a new positive prediction error (the start of the trial is unexpected, but generally positive by proxy since it will always lead to later reward). Likewise, withholding an expected delivery of reward following the CS results in a negative prediction error (see Figure 4, panel F, closely matching the drop in firing observed in panel C).

The importance of this finding is hard to overstate. Research in computer science and behavioral psychology had identified prediction error as a key signal for enabling incremental learning from experience (i.e., Section 3). The discovery that particular neurons

in the brains of monkey reliably code a similar signal suggested an understanding not only of the detailed empirical phenomena, but of the overall function that dopamine plays in learning from experience.

4.2. Model-based analysis of fMRI

Another, very general, way in which RL has informed our understanding of the brain is in the analysis of data collected from humans using fMRI (functional magnetic resonance imaging). Traditional studies using fMRI often contrasted the patterns of brain activity recorded during particular task-relevant states with a suitably defined baseline (e.g., passively looking at a fixation cross). However, RL researchers have pioneered the use of computational models to help structure fMRI data in a trial-by-trial fashion (Daw, 2011; Ashby, 2011). The idea is that computational models inspired by RL algorithms posit the existence of certain latent variables. The trial-by-trial changes in prediction error described above are one example, but so are other latent variables assumed by RL algorithms such as the values of particular state-action pairs (i.e., $Q(s, a)$). During learning, these variables fluctuate dynamically based on the experience and decisions of the learning agent. When first fit to human behavior (i.e., patterns of choices), these models provide excellent targets for structuring analyses of fMRI data. For example, regressors can be constructed representing the trial-by-trial fluctuations in prediction error (δ) which are then correlated with the fluctuations in measured blood oxygen level dependent (BOLD) signal (Platt and Glimcher, 1999; Sugrue et al., 2004; Daw et al., 2006b; Ahn et al., 2011).

Analyses of this type have revealed regions in the human brain which appear reliably responsive to prediction errors (O'Doherty et al., 2003; McClure et al., 2003), and has given insight into how people trade off between exploration and exploitation (Daw et al., 2006b). Even outside of the field of RL, such model-based analysis approaches are changing the way fMRI data are analyzed (e.g., see Davis et al., 2012a,b, in the domain of category learning or Anderson et al, 2008 in problem solving and reasoning).

5. Contemporary Issues in RL Research

5.1. Model-based versus Model-free Learning

Temporal difference learning methods (such as TD and Q-learning described above) depend on direct experience to estimate the value of particular actions. For example, an agent learning via Q-learning will not understand that hot stoves should not be touched until it tries grasping a hot stove and experiences the large negative reward associated with that state-action pair. In this sense, these algorithms are much like early associative theories of conditioning described above: they learn stimulus-response associations between states, actions, and rewards based on direct experience.

However, as the work by Tolman (1948) and others showed, animals exhibit a much richer and more flexible class of learning behaviors. For example, the ability of a rat to

re-route it's path around a novel obstacle would seem to depend on a richer knowledge about the structure of the maze than is assumed by temporal-difference learning methods. In fact, TD and Q-learning are are classified by computer scientists as "model-free" learning algorithms because they do not require the agent to know about the underlying structure of the environment (e.g., the set of transition probabilities, $\mathcal{T}$, and rewards, $\mathcal{R}$ defined above). In model-free RL, estimates of $V(s)$ or $Q(s, a)$ suffice to enable adaptive behavior. In this case, the model refers to something akin to a relatively rich mental model of the task environment.

In contrast, other types of RL algorithms (called model-based RL) emphasize the learning of the transition probabilities and rewards. Once the agent has a representation of these quantities it becomes possible compute "on the fly" the values of $V(s)$ or $Q(s, a)$ at any point in time (using the Bellman insight in Figure 2). In addition, representing the transition probabilities and rewards explicitly allows the agent to plan into the future, forecasting the outcome of possible action sequences. This is exactly the type of behavior exhibited by Tolman's clever maze-running rats and it likely an important part of human decision making as well (e.g., Sloman, 1996).

The distinction between model-based and model-free RL is not purely theoretical. In fact, a growing body of work has explored the idea that multiple learning systems in the brain specialize in these respective types of learning and decision making (Daw et al., 2005, 2011; Otto et al., 2013). In particular, the idea is that habitual behaviors (i.e., those that have been repeated many time and are executed without much though) may be akin, computationally, to model-free learning. Such behaviors are acquired slowly and depend heavily on direct experience. In contrast, model-based RL is more akin to more cognitively effortful forms of planning and reasoning. Research in animal conditioning has pointed to a similar distinction between goal-directed and habitual behaviors (Dickinson, 1985; Dickinson and Balleine, 2004).

The precise computational definition given to these two forms of learning allow model-based and model-free RL to make dissociable behavioral predictions in a variety of tasks. For example, Simon and Daw (2011) present fMRI evidence for dissociable neural systems that separately represented model-based and model-free RL in a realistic spatial navigation task. Similarly, Daw et al. (2011) found evidence for the contribution for differentiable model-based and model-free learning systems in a simple sequential decision making task which become more or less volatile over time. Consistent with the idea that model-based system utilizes more cognitive resources such as working memory and executive control, Otto et al. (2013) found that participants' learning behavior was better fit by model-free algorithms when they performed a sequential decision making task under working memory load.

Computational RL may also provide insight into why these two systems exist. For example, Daw et al. (2005) argue the model-free system takes longer to learn (since getting good estimates for $Q(s, a)$ can take considerable time and experience). On the other hand, model-based systems are able to guide behavior more quickly. However,

the model-based system is more error-prone since it requires an accurate representation of the transition probabilities in the environment. Thus, there are times where the accuracy of the model-free system will exceed the performance of the model-based system. In effect, the two systems are specialized for learning at different time scales. The full details of various model-based RL algorithms is beyond the scope of the present chapter (see Daw, 2012, for a high level summary).

5.2. The influence of state representations on learning

A critical component of almost all existing RL algorithms is the notion of a state (owing the to close relationship between these algorithms and the mathematics of Markov decision processes). In these models, a state is essentially context or situation in which a certain set of choices are available. For example, the idea of a "temporal difference" error signal implicitly assumes that the deviation between the expected and experienced outcomes is conditioned on the current state or context (Schultz et al., 1997). However, a rigorous account of what exactly constitutes a state is often left aside in neurobiological models. To the degree that such models can eventually be extended to explain behavior in more complex learning situations, it is critical that the field adopt a better understanding of how the state representation that the learner adopts influences learning and how such representations are acquired.

The importance of state representation are particularly apparent in sequential decision making contexts where agents make a series of decisions in order to maximize reward. For example, a robot navigating a building could be in a particular state (e.g., a certain position and orientation in a particular hallway) and could make a decision (e.g., turn left), causing a change in state. According to the RL framework, the overall goal of the agent to to make the decision in each state that would maximize its long-term reward. However the way that the agent represents the structure of the environment can strongly influence its ability to achieve this goal. For example, an agent with very limited sensors might have difficulty differentiating between various hallways and intersections and thus would have trouble deciding which action to take in any given case.

A similar problem faces human learners. For example, Gureckis and Love (2009b) studied a sequential decision making task where human participants had to learn to avoid an immediate, short-term gain in order to maximize long-term reward (see also Herrnstein et al., 1993) (see Box 1 for an overview of these types of tasks). Critically, the task was structured such that the reward received on any trial depended on the history of the participant's response on previous trials. Finding the optimal long-term strategy in such environments is difficult because the relationship between distinct task states (in this case, different patterns of prior responses) and the reward on any trial is unclear. Gureckis and Love suggested one reason people show difficulty in such tasks may not stem entirely from an impulsive bias towards immediate gains, but instead because they adopt the wrong state representation of the task. A series of experiments found that providing simple perceptual cues which help to disambiguate successive tasks states greatly improved

participants' ability to find the optimal solution (Gureckis and Love, 2009a,b; Otto et al., 2009). Corresponding simulations with an artificial RL agent based on Q-learning (see Section 3.6) showed that, like humans in the experiment, enriching the discriminability of distinct task states greatly improves learning in the task.

In Gureckis and Love's (2009b) experiment, state cues were simple binary lights which unambiguously mapped to distinct task states. However, in the real world, such state cues are unlikely to be so direct. Consider a case where you must decide which area of a lake is best for fishing. Your options might be the shallows by the shore, or the deep sections in the center. However, a number of factors such as season, time of day, presence of long grass for breeding, or the turbidity of the water may also be relevant. Successful decision making thus requires the integration of a variety of cues in order to identify the current state and enable effective behavior. Note that the question of how people combine multiple cues in order to make predictions about the environment has been extensively studied in the categorization literature. However, the close relationship between work in categorization and RL has only recently been acknowledged (Shohamy et al., 2008; McDonnell and Gureckis, 2009; Gureckis and Love, 2009b; Redish et al., 2007; Gershman et al., 2010).

5.3. Varieties of exploration in humans

Effective RL often requires a delicate balance of exploratory and exploitative behavior (Sutton and Barto, 1998; Steyvers et al., 2009). For example, consider the problem of choosing where to dine out from a set of competing options. The quality of restaurants often changes over time such that one cannot be certain which restaurant is currently best. In this type of "non-stationary" environment, one either chooses the best-experienced restaurant so far (i.e., exploit) or visits a restaurant that was inferior in the past but now may have improved (i.e., explore). Even in stationary environments, knowing when and how much to explore is a difficult and important problem. For example, when outcomes are stationary in time but noisy uncertainty about which option is best can complicate decision making. Closely related problems occur in the foraging literature (Kamil et al., 1987; Stephens and Krebs, 1986). Animals must decide when to abandon a resource patch (e.g., a lake) and seek out a new resource. Optimal foraging requires keeping an estimate of the current reward rate and the expected reward rate for moving to a new resource patch (Kamil et al., 1987).

Exploring when one should exploit and, conversely, exploiting when one should explore both incur costs. For example, an actor who excessively exploits will fail to notice when another action becomes superior. Conversely, an actor who excessively explores incurs an opportunity cost by frequently forgoing the high payoff option. How often one should explores should vary as a function of the environment. For environments that are volatile and undergo frequent change, one should explore more often (Daw et al., 2011; Knox et al., 2011). In contrast, there is little reason to explore a well-understood environment that never changes. Other factors that affect how often one should explore include

the task horizon (i.e., how many more opportunities there are to make a choice) and how rewarding the environment is in general (Rich and Gureckis, in review; Steyvers et al., 2009). To return to the restaurant scenario, there is little reason on the last day of vacation to try a new restaurant when all the restaurants besides one's favorite have proven to be horrible. On the other hand, if it's early in the vacation and most restaurants are generally good, then it makes sense to explore.

The exploration methods considered above, such as softmax (Eq. 12) and ϵ-greedy (see Section 3.7) do not explicitly consider these issues concerning exploration. However, other methods for regulating uncertainty do. Below, a number of methods of exploring in dynamic decision environments are considered. The methods are arranged from least to most sophisticated. Each method has its place in both engineering applications and for modeling human behavior.

5.3.1. Trial Independent "Random" Exploration

This form of exploration is the simplest and most commonly used in psychology, neuroscience, and computer science (e.g., Daw et al., 2006c; Sutton and Barto, 1998). On each trial, a value estimate is gathered for each possible action. Exploiting corresponds to choosing the highest value option, whereas exploring corresponds to choosing some other action. This form of exploration is referred to as trial independent because the probability of choosing an action is not influenced by what was chosen on past trials or what will be chosen on future trials. The only factor determining the probability of selecting an action on the current trial is its current value. Examples of choice procedures using this scheme are softmax (Eq. 12) and ϵ-greedy. The strengths of these forms of exploration include simplicity. For example, these methods do not require a complex analysis of the environment. Because every action has some chance of being sampled on every trial, a well-designed learning rule (e.g., Q-learning) will eventually discover the optimal policy. Weakness include that there is no guarantee that one is exploring when one should.

5.3.2. Systematic Exploration

Trial independent random exploration does not conform to popular intuition about what exploration is. Consider a historic explorer, such as Christopher Columbus, sailing toward the new world. His ships did not move east one day, west the next, and then south on the third day as they might according to a softmax exploration procedure where a choice is made on every trial. Instead, a series of linked and similar (i.e., repeated) actions were taken to truly move into unchartered territory. In certain tasks, exploration that reaches novel states requires consistent action across trials, which is not readily achievable with trial independent random exploration. Several consistent choices might move an agent to a novel state that several thousand "random" choices would never reach.

There are certain tasks where people likely explore in a similar fashion. Otto et al. (Otto et al., 2010) conducted a dynamic decision task that had a local optimum that people could only escape by repeatedly taking actions that resulted in lower immediate reward,

but higher subsequent reward (cf. Bogacz et al., 2007). These authors found that when people were in a motivational state that fosters cognitive flexibility (cf. Higgins, 2000) that subjects tended to be streaky in their choices, repeating the same option for several trials in a row. In particular, these subjects were best modeled by a choice procedure which repeated the previous choice with probability p and with probability $1 - p$ the softmax procedure determined the choice. This generally leads to systematic, streaky patterns of exploration.

5.3.3. Optimal Planning

The above methods for balancing exploration and exploitation are heuristic in nature. More ideally, one would calculate an overall plan in which exploration was integral to maximizing total reward. For example, for an ideal actor that uses its beliefs and plans ahead there is no distinction between exploration and exploitation as all actions are following a policy that maximizes expected value Gittins (1979).

This may seem like a philosophical distinction but it is conceptually and practically important. The above methods acknowledge there is information value in taking actions that currently have lower-expected reward. These methods are blind to what the true best policy is, so need to explore in hopes of discovering it. In contrast, an ideal actor that performs optimal planning chooses to "explore" **because** it is the next move in the plan that has the highest expected value. In this sense, it is incorrect to say exploration ever occurs in an ideal actor model because every move is exploitative of long-term reward.

One issue is that computing the ideal actor's policy is computationally challenging and often can only be done for certain problems (cf., Kaelbling et al., 1998). Optimal solutions can be calculated for relatively simple problems, such a finite horizon n-armed bandit task (Gittins, 1979; Steyvers et al., 2009). Interestingly, recent work in psychology suggests that people actually can adopt such optimal decision policies in certain situations (Rich and Gureckis, in review; Knox et al., 2011, e.g.). Another example includes modeling navigation in mazes where people begin from a random starting position (Stankiewicz et al., 2006).

5.4. Varieties of reward

In RL, rewards are fundamental in that they define the task for the agent. Rewards would seem to be a straightforward, transparent, and immutable – the reward is simply the signal the environment provides and the agent receives. In reality, the concept of reward in RL is much richer, complex, and subtle. Although rewards can follow in a straightforward manner from the environment, in many RL domains and applications rewards are another aspect of the overall system that is manipulated by the agent designer or human experimenter.

For example, in Gureckis and Love (2009a) human and artificial agents both performed better in a difficult RL problem when low levels of noise were added to the reward signal. In other words, corrupting the reward signal with noise actually increased agent

performance. The reason for this surprising outcome was that agents tended to under-explore initially and the added noise in the reward signal had the effect of increasing agent exploration, which benefitted the agent in the long-run. Although not presented this way in the original paper, this is a case where a limitation in an agent can be addressed by the design of a reward signal.

This notion has been explored formally in machine learning (Singh et al., 2010). When an agent is computationally bounded in some sense, there can be an alternative reward signal that will lead to the agent performing better than the external reward signal. Likewise, agents are often endowed with additional internal reward signals to create basic drives or motivations, such as curiosity (Schmidhuber, 1990; Oudeyer and Kaplan, 2007) or the desire for information Gureckis and Markant (2012). From a psychological perspective, models of human behavior have been formulated in which actions (and corresponding rewards) include internal actions, such as storing information into a working memory store (Dayan, 2012; Gray et al., 2006; Todd et al., 2008). The costs associated with these internal operations can be optimized by learning algorithms, such as Q-learning, to capture human performance given task constraints.

Finally, alternative ways of training agents sidestep many of the demands of traditional RL schemes that require extensive exploration of undesirable states to converge on a policy. One approach is learning by demonstration (Abbeel et al., 2013). In this line of work, a difficult task is demonstrated to an agent, which dramatically speeds learning. For example, an agent can learn to control a helicopter much faster by having certain maneuvers demonstrated (i.e., observe the actions and corresponding outcomes) than by self-exploring the huge space of action sequences and subsequent rewards. Another approach is interactive shaping in which an outside teacher provides a reward signal to the agent rather than the environment providing the reward signal. Agents can master difficult tasks, such as playing Tetris, much more quickly by receiving evaluative feedback from an observing human teacher than by learning to play by exploring and experiencing subsequent rewards (Knox et al., 2012; Knox and Stone, 2009). The human supplied feedback effectively transforms an RL task into a supervised learning task in which only the immediate action is rewarded. Many of the inherent difficulties of RL problems, such as delayed rewards and learning the proper sequence of actions, are sidestepped by this approach.

6. Concluding remarks

Reinforcement learning represents an dynamic confluence of ideas from psychology, neuroscience, computer science, and machine learning. The power and generality of the framework is made clear by the diverse range of theoretical and practical ideas which it has help to articulate. That said there are many important open issues in the field which have been hinted at above.

7. Glossary

- **Credit Assignment** - refers to the problem of assigning blame to actions when rewards are delayed in time. For example, if you leave your cup on the edge of a table, then three days later nudge the table with your hip and break the cup, which action is most responsible for the negative outcome? In one way it is the earlier action of leaving the cup in a dangerous place. However, traditional theories of conditioning might assume that more immediate actions are to blame. RL deals with this problem having the value of action depend not only on their immediate outcomes but on the sum of future rewards available following that action.

- **Explore/Exploit Dillemma** - In environments where the reward associated with different actions are unknown, agents face a decision dilemma to either return to options previously known to be positive (exploit), or to explore relatively unknown options. The optimal balanced between exploration and exploitation is computable in some environments but generally can be computationally intractable.

- **Temporal Difference (TD) learning** - is a method of computational RL which learns the value of actions through experience. In particular, in TD, learning is based on a deviation between what is expected at one point in time and what is actually experienced. TD is explain in detail in Section 3.5.

- **Policy** - a set of rule that determine which action an agent should selected in each possible state in order to maximize the expectation of long-term reward. A everyday example of a policy would be a list of directions from getting from location A to location B. At each interaction the directions (policy) tell the agent which decision to make.

- **State** - refer to distinct situations an agent may be in. For example, being stopped at a red light on the corner of Houston St. and Broadway in NYC might be a state. States derive their important in RL from the dependence of most RL methods on the underlying mathematics of Markov Decision Processes (MDPs).

- **Reward** - is typically a scalar value that determines the quality or desirability of an outcome or situation. Typically rewards are designed by society, experimenters, or roboticists to guide the behavior of agents in particular ways. The goal of an RL agent is to maximize the reward received over the long term.

- **Myopic behavior** - the behavior of RL agents often depends on the degree to which they value long-term versus short-term outcome. A parameters (γ) in most RL models controls this tradeoff. When γ is zero, RL agents make choices that only value immediate rewards. This is often described as "myopic" behavior because it disregards future consequences of actions.

8. Text Box 1: Experimental approaches

Given the description in Section 3 of common RL algorithms, it is useful to consider the range of behavioral phenomena these theories have been used to explain. Overall, the diversity of tasks which have been modeling using RL is a testament to the very general framework it provides for thinking about human and animal behavior.

8.1. Classical and instrumental conditioning

As mentioned earlier, the contemporary field of RL was anticipated by early work in classical and instrumental conditioning. Such early studies identified notions of contingency, reward, and prediction error as possible determinants of learning (e.g., Rescorla and Wagner, 1972; Wagner and Rescorla, 1972). It is not surprising then that RL algorithms have continued to be influential as theories of basic conditioning phenomena. For example, the temporal difference methods described above were largely developed by theorists to explain continuous-time effects on conditioning as well as second-order conditioning, two classical (i.e., Pavlovian) phenomena which are difficult to account for using standard models such as the Rescorla-Wagner model (Niv and Schoenbaum, 2008).

When modeling classical conditioning, the standard TD learning algorithm described in Section 3.5 is often used. This is natural since in classical conditioning paradigms, the participant does not have any control over the task (e.g., there are no choices to be made). For example, in an eye-blink conditioning experiment a tone might sound moments before a puff of air is delivered to the participant's eye. When applied to classical conditioning phenomena, particular assumptions are made about the representations of "states" in the system. For example, TD models of classical conditioning (see Section 4.1) assume that continuous time is divided into arbitrarily small discrete units, each of which constitutes a different state in an MDP (Sutton, 1995; Daw et al., 2006a; Ludvig et al., 2012). The value of being in the state of hearing the tone might be lower since it means a puff of air is coming soon.

RL algorithms have also been applied to instrumental conditioning paradigms explaining issues like the effects of reward rate on responding and response vigor (e.g., Niv et al., 2007). Here is is natural to consider learning algorithms like Q-learning or actor-critic architectures (not reviewed here but see Konda and Tsitsiklis (1999)) where agents explicitly are estimating the value of particular actions in different states and adjusting decision policies based on these experiences.

8.2. Bandit tasks

A particular kind of instrumental conditioning paradigm that has strongly influenced research on RL in humans is the multi-armed bandit task. The basic experiment procedure is named after the slot machines in casinos which deliver payouts probabilistically when the "arm" of the machine is pulled (in fact, these machines are sometimes known as "one-armed bandits"). In the multi-armed bandit task, a number of choice options are

presented to the subject. On each trial the subject can select one of the bandits. Selected bandits pay out a random reward from a distribution specified by the experimenter. The goal of the subject is to maximize the total reward they earn across a finite number of trials (see Steyvers et al., 2009; Lee et al., 2011, for discussion of human experimentation and modeling).

Bandit tasks are simple experiments which expose many of the core aspects of RL described above. For example, the learner has to balance exploration and exploitation between different bandits in order to ensure they are consistently choosing the best one. In addition, bandit tasks involve incremental learning of the valuation of different options across multiple trials. In some variants of a bandit task the payout probability of the different options drifts randomly over time (sometimes called "restless bandits") further encouraging continuous learning and exploration (e.g., Daw et al., 2006b; Yi et al., 2009).

8.3. Sequential decision making tasks

A key feature of RL models is the ability to learn to make sequences of decisions to achieve some goal. Experiments assessing this behavior tend to use tasks with greater structure than a traditional bandit task. For example, researchers have considered variants on bandit tasks where the reward rates of different options are linked together in particular ways. For example, Knox et al. (2011) explored a "leap-frog" type tasks where the overall reward rates of two bandits take turns increasing in value at random points during the experiment. As discussed below, this additional structure favors different types of exploration strategies which are more structured in time (see Section 5.3). Relatedly, some researchers have explore dynamic decision making tasks where the payoffs available from different actions depends on the past history of choices made by the agent (Neth et al., 2006; Gureckis and Love, 2009a,b; Otto et al., 2009, 2010; Otto and Love, 2010). In these cases, optimal, reward-maximizing strategies may require more complex sequential decisions similar to playing a game like tic-tac-toe or chess (see Section 5.2). Researchers have even used complex video games that involve spatial navigation to interrogate basic learning and decision processes (e.g., Simon and Daw, 2011).

9. Acknowledgments

The authors wish to thank Julie Hollifield for help with the figures and Nathaniel Daw, David Halpern, John McDonnell, Yael Niv, Alex Rich, and A. Ross Otto for helpful discussions in the preparation of the manuscript. TMG was supposed by grant number BCS-1255538 from the National Science Foundation and contract D10PC20023 from Intelligence Advanced Research Projects Activity (IARPA) via Department of the Interior (DOI).

10. References

Abbeel, P., Coates, A., Ng, A. Y., 2013. Autonomous helicopter aerobatics through apprenticeship learning. The International Journal of Robotics Research 32, 458–482.

Ahn, W.-Y., Krawitz, A., Kim, W., Busemeyer, J., Brown, J., 2011. A model-based fmri analysis with hierarchical bayesian parameter estimation. Journal of Neuroscience, Psychology, and Economics 4 (2), 95–110.

Anderson, J., 1990. The Adaptive Character of Thought. Lawrence Erlbaum Associates.

Anderson, J., Carter, C., Fincham, J., Qin, Y., Ravizza, S., Rosenberg-Lee, M., 2008. Using fmri to test models of complex cognition. Cognitive Science 32, 1323–1348.

Ashby, F., 2011. Statistical analysis of fMRI data. MIT Press, Cambridge, MA.

Bagnell, J., Schneider, J., 2001. Autonomous helicopter control using reinforcement learning policy search methods. In: International Conference on Robotics and Automation. IEEE, pp. 1615–1620.

Behrens, T., Woolrich, M., Walton, M., Rushworth, M., 2007. Learning the value of information in an uncertain world. Nature Neuroscience 10 (9), 1214–1221.

Bernoulli, D., 1954. Exposition of a new thoeyr on the measurement of risk. Econometrica 22, 23–36.

Bertsekas, D., Tsitsiklis, J., 1996. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA.

Bogacz, R., McClure, S., Li, J., Cohen, J., Montague, P., 2007. Short-term memory traces for action bias in human reinforcement learning. Brain Research 1153, 111–121.

Busemeyer, J., Stout, J., 2002. A contribution of cognitive decision models to clinical assessment: Decomposing performance on the bechara gambling task. Psychological Assessment 14 (3), 253–262.

Davis, T., Love, B., Preston, A., 2012a. Learning the exception to the rule: Model-based fmri reveals specialized representations for surprising category members. Cerebral Cortex 22, 260–273.

Davis, T., Love, B., Preston, A., 2012b. Striatal and hippocampal entropy and recognition signals in category learning: Simultaneous processes revealed by model-based fmri. Journal of Experimental Psychology: Learning, Memory, & Cognition 38, 821–839.

Daw, N., 2011. Trial-by-trial data analysis using computational models. In: E.A., P., T.W., R., M., D. (Eds.), Affect, Learning and Decision Making, Attention and Performance XXIII. Oxford University Press.

Daw, N., 2012. Model-based reinforcement learning as cognitive search: Neurocomputational theories. In: Todd, P., Hills, T., Robbins, T. (Eds.), Cognitive search: Evolution, algorithms, and the brain. MIT Press, Cambridge, MA.

Daw, N., Courville, A., Touretzky, D., 2006a. Representation and timing in theories of the dopamine system. Neural Computation 18, 1637–1677.

Daw, N., Gershman, S., Seymour, B., Dayan, P., Dolan, R., 2011. model-based influences on humans' choices and striatal prediction errors. Neuron 69, 1204–1215.

Daw, N., Niv, Y., Dayan, P., 2005. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nature Neuroscience 8 (12), 1704–1711.

Daw, N., O'Doherty, J., Seymour, B., Dayan, P., Dolan, R., 2006b. Cortical substrates for exploratory decision in humans. Nature 441, 876–879.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., Dolan, R. J., 2006c. Cortical substrates for exploratory decisions in humans. Nature 441 (7095), 876–9.

Dayan, P., 2012. How to set the switches on this thing. Curr Opin Neurobiol 22 (6), 1068–74, dayan, Peter England Curr Opin Neurobiol. 2012 Dec;22(6):1068-74. doi: 10.1016/j.conb.2012.05.011. Epub 2012 Jun 15.

Dickinson, A., 1985. Actions and habits: The development of behavioural autonomy. Philosophical Transactions of the Royal Society B: Biological Sciences 308, 67–78.

Dickinson, A., Balleine, B., 2004. The role of learning in the operation of motivational systems. In: Gallistel, R. (Ed.), Stevens' handbook of experimental psychology: Vol. 3. Learning, motivation, and emotion, 3rd Edition. Wiley, Hoboken, NJ.

Frederick, S., Loewenstein, G., O'Donoghue, T., 2002. Time discounting and time preference: A critical review. Journal of Economic Literature XL, 351–401.

Gershman, S., Blei, D., Niv, Y., 2010. Context, learning, and extinction. Psychological Review 117, 197–209.

Gittins, J. C., 1979. Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society, Series B 41, 148–177.

Gray, W. D., Sims, C. R., Fu, W. T., Schoelles, M. J., 2006. The soft constraints hypothesis: a rational anal-

ysis approach to resource allocation for interactive behavior. Psychological review 113 (3), 461–82, gray, Wayne D Sims, Chris R Fu, Wai-Tat Schoelles, Michael J Psychol Rev. 2006 Jul;113(3):461-82.

Gureckis, T., Love, B. C., 2009a. Learning in noise: Dynamic decision-making in a variable environment. Journal of Mathematical Psychology 53, 180–193.

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

Gureckis, T., Markant, D., 2012. A cognitive and computational perspective on self-directed learning. Perspectives in Psychological Science.

Herrnstein, R., Loewenstein, G., Prelec, D., Vaughan, W., 1993. Utility maximization and melioration: Internalities in individual choice. Journal of Behavioral Decision Making 6, 149–185.

Higgins, E. T., 2000. Making a good decision: Value from fit. American Psychologist 55, 217–1230.

Kaelbling, L. P., Littman, M. L., Cassandra, A. R., 1998. Planning and acting in partially observable stochastic domains. Artificial Intelligence 101, 99–134.

Kamil, A. C., Krebs, J. R., Pulliam, H. R., 1987. Foraging Behavior. Plenum Press, London.

Knox, W., Glass, B., Love, B., Maddox, W., Stone, P., 2012. How humans teach agents. International Journal of Social Robotics 4 (4), 409–421.

Knox, W. B., Otto, A. R., Stone, P., Love, B. C., 2011. The nature of belief-directed exploratory choice in human decision-making. Frontiers in psychology 2, 398, knox, W Bradley Otto, A Ross Stone, Peter Love, Bradley C Switzerland Front Psychol. 2011;2:398. doi: 10.3389/fpsyg.2011.00398.

Knox, W. B., Stone, P., 2009. Interactively shaping agents via human reinforcement: The tamer framework. In: Proceedings of The Fifth International Conference on Knowledge Capture (K-CAP 2009). pp. 9–16.

Konda, V., Tsitsiklis, J., 1999. Actor-critic algorithms. In: Neural Information Processing Systems. MIT Press.

Krugel, L., Biele, G., Mohr, P., Li, S.-C., Heekeren, H., 2009. Genetic variation in dopaminergic neuromodulation influences the ability to rapidly and flexibly adapt decisions. Proceedings of the National Academy of Sciences 106, 17951–17956.

Lee, M., Zhang, S., Munro, M., Steyvers, M., 2011. Psychological models of human and optimal performance in bandit problems. Cognitive Systems Research 12, 164–174.

Luce, R. D., 1959. Individual choice behavior: A theoretical analysis. Greenwood Press, Westport, Conn.

Ludvig, E., Sutton, R., Kehoe, E., 2012. Evaluating the td model of classical conditioning. Learning and Behavior 40 (3), 305–319.

Marr, D., 1982. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Freeman.

McClure, S., Berns, G., Montague, P., 2003. Temporal prediction errors in a passive learning task activate human striatum. Neuron 38 (2), 339–346.

McDonnell, J., Gureckis, T., 2009. How perceptual categories influence trial and error learning in humans. In: Multidisciplinary Symposium on Reinforcement Learning. Montreal, Canada.

Montague, P., Dayan, P., Person, C., Sejnowski, T., 1995. Bee foraging in uncertain environments using predictive hebbian learning. Nature 377 (6551), 725–728.

Montague, P., Dayan, P., Sejnowski, T., 1996. A framework for mesencephalic dopmaine system based on predictive hebbian learning. Journal of Neuroscience 16 (5), 1936–1947.

Myerson, J., Green, L., 1995. Discounting of delayed rewards: Models of individual choice. Journal of the Experimental Analysis of Behavior 64, 263–276.

Nassar, M., Gold, J., 2013. A healthy fear of the unknown: Perspectives on the interpretation of parameter fits from computational models in neuroscience. PLOS Computational Biology 9 (4), e1003015.

Neth, H., Sims, C., Gray, W., 2006. Melioration dominates maximization: Stable suboptimal performance despite global feedback. In: Sun, R., Miyake, N. (Eds.), Proceedings of the 28th Annual Meeting of the Cognitive Science Society. Lawrence Erlbaum Associates, Hillsdale, NJ.

Niv, Y., 2009. Reinforcement learning in the brain. Journal of Mathematical Psychology 53 (3), 139–154.

Niv, Y., Daw, N., Joel, D., Dayan, P., 2007. Tonic dopamine: Opportunity costs and the control of response vigor. Psychopharamcology 191 (3), 507–520.

Niv, Y., Schoenbaum, G., 2008. Dialogues on prediction errors. Trends in Cognitive Sciences 12 (7), 265–272.

O'Doherty, J., Dayan, P., Friston, K., Critchley, H., Dolan, R., 2003. Temporal difference learning model accounts for responses in human ventral striatum and orbitofrontal cortex during pavlovian appetitive learning. Neuron 38, 329–337.

Otto, A., Gershman, S., Markman, A., Daw, N., 2013. The curse of planning: Dissecting multiple reinforcement learning systems by taxing the central executive. Psychological Science 24 (5), 751–761.

Otto, A., Gureckis, T., Love, B., Markman, A., 2009. Navigating through abstract decision spaces: Evaluating the role of state knowledge in a dynamic decision making task. Psychonomic Bulletin and Review 16 (5), 957–963.

Otto, A., Love, B., 2010. You don't want to know what you're missing: When information about forgone rewards impedes dynamic decision making. Judgment and Decision Making 5, 1–10.

Otto, A., Markman, A., Gureckis, T., Love, B., 2010. Regulatory fit in a dynamic decision-making environment. Journal of Experimental Psychology: Learning, Memory, and Cognition.

Oudeyer, P. Y., Kaplan, F., 2007. What is intrinsic motivation? a typology of computational approaches. Frontiers in Neurorobotics 1 (6).

Platt, M., Glimcher, P., 1999. Neural correlates of decision variables in parietal cortex. Nature 400 (6741), 233–238.

Redish, A., Jensen, S., Johnson, A., Kurth-Nelson, Z., 2007. Reconciling reinforcement learning models with behavioral extinction and renewal: Implications for addition, relapse, and problem gamling. Psychological Review 114 (3), 784–805.

Rescorla, R., 1980. Pavlovian second-order conditioning: Studies in associative learning. Lawrence Erlbaum Associates, Hillsdale, N.J.

Rescorla, R., Wagner, A., 1972. A theory of pavolvian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In: Black, A., Prokasy, W. (Eds.), Classical Conditioning II: Current Research and Theory. Appleton-Century-Crofts, New York, pp. 64–99.

Rich, A., Gureckis, T., in review. The value of approaching bad things. In: Proceedings of the 36th Annual Conference of the Cognitive Science Society. Cognitive Science Society, Austin, TX.

Schmidhuber, J., 1990. A possibility for implementing curiosity and boredom in model-building neural controllers.

Schultz, W., Apicella, P., Ljungberg, T., 1993. Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task. Journal of Neuroscience 13 (3), 900–913.

Schultz, W., Dayan, P., Montague, P. R., 1997. A neural substrate of predicion and reward. Science 275, 1593–1598.

Shohamy, D., Myers, C., Kalanithi, J., Gluck, M., 2008. Basal ganglia and dopamine contributions to probabilistic category learning. Neuroscience and biobehavioral reviews 32 (2), 219–236.

Simon, D., Daw, N., 2011. Neural correlates of forward planning in a spatial decision task in humans. Journal of Neuroscience 31, 5526–5539.

Singh, S., Lewis, R. L., Barto, A. G., Sorg, J., 2010. Intrinsically motivated reinforcement learning: An evolutionary perspective. IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT 2 (2), 70–82.

Skinner, B., 1938. The Behavior of Organisms: An Experimental Analysis. Appleton-Century, Oxford, England.

Sloman, S., 1996. The empirical case for two systems of reasoning. Psychological Bulletin 119 (1), 3–22.

Stankiewicz, B., Legge, G., Mansfield, J., Schlicht, E., 2006. Lost in virtual space: Studies in human and ideal spatial navigation. Journal of Experimental Psychology: Human Perception and Performance 32, 688–704.

Stephens, D., Krebs, J., 1986. Foraging Theory. Princeton University Press, Princeton, NJ.

Steyvers, M., Lee, M., Wagenmakers, E., 2009. A bayesian analysis of human decision-making on bandit problems. Journal of mathematical psychology 53, 168–179.

Sugrue, L., Corrado, G., Newsome, W., 2004. Matching behavior and the representation of value in the parietal cortex. Science 304, 1782–1787.

Sutton, R., 1988. Learning to predict by the method of temporal difference. Machine Learning 3, 9–44.
Sutton, R., 1995. Td models: Modeling the world at a mixture of time scales. In: Proceedings of the 12th International Conference on Machine Learning. Morgan-Kaufmann, pp. 531–539.
Sutton, R., Barto, A., 1981. Toward a modern theory of adaptive networks: Expectation and prediction. Psychological Review 88, 135–170.
Sutton, R., Barto, A., 1998. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.
Tesauro, G., 1994. Td-gammon, a self-teaching backgammon program, achieves master-level play. Neural Computation 6 (2), 215–219.
Thorndike, E., 1911. Animal Intelligence: Experimental Studies. Macmillan, New York.
Todd, M., Niv, Y., Cohen, J., 2008. Learning to use working memory in partially observable environments through dopaminergic reinforcement. Advances in Neural Information Processing Systems.
Tolman, E., 1948. Cognitive maps in rats and men. Psychological Review 55 (4), 189–208.
Wagner, A., Rescorla, R., 1972. Inhibition in pavlovian conditioning: Application of a theory. In: Boake, R., Halliday, M. (Eds.), Inhibition and Learning. Academic Press, London, pp. 301–336.
Watkins, C., 1989. Learning from delayed rewards. Ph.D. thesis, Cambridge University, Cambridge, England.
Yi, S., Steyvers, M., Lee, M., 2009. Modeling human performance on restless bandit problems using particle filters. Journal of Problem Solving 2 (2).