LETTER Communicated by Eric Moulines

Posterior Weighted Reinforcement Learning with State Uncertainty

Tobias Larsen
larsent@tcd.ie
Department of Computer Science, University of Bristol, Bristol, BS8 1UB, U.K.

David S. Leslie
david.leslie@bristol.ac.uk
Edmund J. Collins
e.j.collins@bristol.ac.uk
Department of Mathematics, University of Bristol, Bristol, BS8 1TW, U.K

Rafal Bogacz
R.bogacz@bristol.ac.uk
Department of Computer Science, University of Bristol, Bristol, BS8 1UB, U.K

Reinforcement learning models generally assume that a stimulus is presented that allows a learner to unambiguously identify the state of nature, and the reward received is drawn from a distribution that depends on that state. However, in any natural environment, the stimulus is noisy. When there is state uncertainty, it is no longer immediately obvious how to perform reinforcement learning, since the observed reward cannot be unambiguously allocated to a state of the environment. This letter addresses the problem of incorporating state uncertainty in reinforcement learning models. We show that simply ignoring the uncertainty and allocating the reward to the most likely state of the environment results in incorrect value estimates. Furthermore, using only the information that is available before observing the reward also results in incorrect estimates. We therefore introduce a new technique, posterior weighted reinforcement learning, in which the estimates of state probabilities are updated according to the observed rewards (e.g., if a learner observes a reward usually associated with a particular state, this state becomes more likely). We show analytically that this modified algorithm can converge to correct reward estimates and confirm this with numerical experiments. The algorithm is shown to be a variant of the expectation-maximization algorithm, allowing rigorous convergence analyses to be carried out. A possible neural implementation of the algorithm in the cortico-basal-ganglia-thalamic network is presented, and experimental predictions of our model are discussed.

Neural Computation **22, 1149–1179** (2010)

1 Introduction

Reinforcement learning is a technique that allows an individual to learn based on rewards experienced through interaction with the environment. It is inspired by animal behavior (Sutton, 1988) and can be used as a model for learning tasks such as finding food, avoiding predation, or finding a mate (Dayan & Abbott, 2001). The goal for a reinforcement learning method is to estimate the expected reward associated with each state of the environment (or each action). These estimates can then be used to inform action choice.

The most common models of reinforcement learning use the temporal difference (TD) method, in which observed rewards are compared with predicted rewards and the difference used to update the predictions for the next time step (Sutton & Barto, 1998). Montague, Dayan, and Sejnowski (1996) have proposed that during learning tasks, this algorithm is employed in neural circuits of the basal ganglia, and in particular, the TD prediction error is represented in the activity of neurons releasing neurotransmitter dopamine. This theory has since been supported by large amounts of experimental data (Schultz, 1998; Frank, Seeberger, & O'Reilly, 2004; Ungless, Magill, & Bolam, 2004; Tobler, Fiorillo, & Schultz, 2005; D'Ardenne, McClure, Nystrom, & Cohen, 2008).

In theoretical developments of reinforcement learning (see Sutton & Barto, 1998) it is usually assumed that a learner is able to identify its state unambiguously on the basis of a stimulus from the environment, and the reward received is drawn from a distribution that depends on that state. Hence, it is clear to which state of the environment a received reward should be attributed and the TD update can be calculated.

However, in any natural environment, the stimulus is noisy and might even be ambiguous. This situation is modeled in many experiments investigating the neural bases of decision making (Britten, Shadlen, Newsome, & Movshon, 1992; Shadlen & Newsome, 1996, 2001; Roitman & Shadlen, 2002). When there is state uncertainty, it is no longer immediately obvious how to perform the TD update, since the observed reward cannot be unambiguously allocated to a state of the environment.

This letter addresses the problem of incorporating the resulting state uncertainty in reinforcement learning models. We show that simply ignoring the stimulus uncertainty and allocating the reward to the most likely state of the environment results in incorrect value estimates. Furthermore, using only the state information that is available before observing the reward also results in incorrect estimates. We therefore introduce a new technique in which the estimates of state probabilities are updated according to the observed rewards (e.g., if a learner observes a reward usually associated with a particular state, this state becomes more likely). We show that this modified algorithm can converge to correct reward estimates. The technique uses similar principles to the expectation-maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), and in fact we show it to be a version of an online EM algorithm (Titterington, 1984; Wang & Zhao, 2006; Cappé & Moulines, 2009). In the appendix, we show that even if the learner uses an incorrect model of the probability distribution, the learned distributions will minimize a standard measure of distance from the true data-generating distribution. This provides robustness to the choice of probability model.

In the following section, we describe reinforcement learning and the experimental setup in which stimulus uncertainty is present. In section 3 we introduce three possible reinforcement learning schemes for such a model. Section 4 discusses a more challenging environment in which the rewards for each state are switched partway through the learning process, and it outlines a modified learning algorithm that responds more successfully to this change of reward. The proposed algorithm is shown, in section 5, to be a version of an online EM algorithm; further analysis is provided in the appendix. Section 6 suggests a possible neural implementation of the proposed algorithm, and section 7 discusses experimental predictions of the model.

2 Learning Environment

Consider a learning environment in which there is discrete set of states $\{1, 2, \ldots, N\}$. Associated with each state is a reward distribution; if the environment is in state i at time t the learner receives a random reward R_t drawn from a distribution with mean $\mu(i)$. We assume that, conditional on the state at time t, the reward R_t is independent of the rewards and states at all other times (this assumption is satisfied in typical behavioral experiments). Note that the set of states $\{1, 2, \ldots, N\}$ could actually encode a set of state-action pairs (s, a), as in many reinforcement learning algorithms (Sutton & Barto, 1998), but for the purpose of this letter we prefer not to introduce this extra level of notational complexity (the issue of online learning for action selection is discussed briefly in section 7.3).

This letter considers reinforcement learning algorithms that attempt to learn the mean parameters $\mu(i)$ for each state of the environment. In standard reinforcement learning models, the learner knows unambiguously that the state at time t is i_t. It is well known (see Sutton & Barto, 1998) that an effective scheme in this case is to maintain estimates $Q_t(i)$ for $i \in \{1, \ldots, N\}$ and, after observing reward R_t, update the estimates according to

$$Q_{t+1}(i) = \begin{cases} Q_t(i) + \alpha \left\{R_t - Q_t(i)\right\} & \text{if } i = i_t, \\ Q_t(i) & \text{otherwise,} \end{cases} \tag{2.1}$$

where $\alpha \in (0, 1)$ is a learning rate parameter. The term $R_t - Q_t(i)$ is called the temporal difference, since it is the difference between the predicted and the received reward. Throughout the letter, vector quantities are denoted in bold. Thus, $\boldsymbol{Q}_t$ denotes the vector $(Q_t(1), \ldots, Q_t(N))$ of value estimates that the learner maintains.

Note that, for proofs of the almost sure convergence of reinforcement learning, one needs the learning rate parameter α to decrease over time at a particular rate, as in the stochastic approximation literature (see, e.g., Kushner & Yin, 1997; Benaïm, 1999). However, for this letter, we retain fixed small $\alpha > 0$ and provide sketch proofs of convergence results. The main intuition we will use is that the only valid points of convergence for such a scheme are stochastic fixed points, where the expected change in $Q_t(i)$ is 0 for each i. Thus, for the simple reinforcement learning scheme, equation 2.1, we have

$$\begin{aligned}\mathbb{E}[Q_{t+1}(i) - Q_t(i) \mid \boldsymbol{Q}_t] &= \mathbb{E}\left[\mathbb{I}_{\{i_t=i\}}\alpha\left\{R_t - Q_t(i)\right\}\right]\\ &= \alpha\mathbb{P}(i_t = i)\left\{\mu(i) - Q_t(i)\right\},\end{aligned}$$

where $\mathbb{E}$ denotes expectation, $\mathbb{P}$ denotes probability, and $\mathbb{I}$ denotes an indicator function taking value 1 if the condition is true and 0 otherwise. Hence, if $\mathbb{P}(i_t = i)$ is fixed and nonzero for each i, we would expect that convergence can occur only to points $Q_\infty(i) = \mu(i)$.

In this letter, however, we consider an environment where individuals do not know the state i_t unambiguously. Such environments are often used in psychological experiments to show that humans and animals are unable to discriminate between ambiguous stimuli with 100% accuracy (Usher & McClelland, 2001). These studies show that, as the time allowed to observe the stimuli increases, the discrimination accuracy initially increases but then reaches an asymptotic level which depends on the difficulty of the discrimination (Usher & McClelland, 2001). On the basis of the analysis of behavioral data, it has been proposed that when humans are presented with ambiguous stimuli they accumulate noisy evidence until the integrated evidence reaches a fixed threshold (Ratcliff, 1988, 2006). This theory has been recently supported by neural activity recorded in monkeys (Kiani, Hanks, & Shadlen, 2008). Other recent evidence (Kepecs, Uchida, Zariwala, & Mainen, 2008) suggests that rats have a neural correlate of confidence. Thus, it is reasonable to consider models in which the learner is aware of their confidence level.

Inspired by this theory, we construct a model in which, at each trial t, the learner identifies one state s_t as the true state, and there is probability $\rho > 1/N$ that this identification is correct. Moreover, the learner knows this probability ρ, and thus ρ can be interpreted as the learner's level of confidence. For simplicity of exposition, we assume that all states other than that identified by the learner are equally likely to be the true state, so that

$$\mathbb{P}(i_t = i \mid s_t) = \rho_t(i) := \begin{cases} \rho & \text{if } i = s_t, \\ \frac{1-\rho}{N-1} & \text{otherwise.} \end{cases}$$

While this assumption will not be satisfied in most natural environments, it is not important for our results and simplifies the mathematical exposition. We furthermore assume that s_t is equally likely to take any value in $\{1, \ldots, N\}$. Again, this simplifying assumption is not important.

3 Reinforcement Learning Models

The standard model of reinforcement learning in equation 2.1 cannot be applied directly when there is uncertainty about the stimulus that has been presented since i_t is not known. In this section, we present several possible solutions to the problem.

3.1 Winner Takes All. The simplest algorithm we consider is simply to ignore the fact that stimulus uncertainty exists. We do this by implementing a "winner-takes-all" strategy where the state s_t with the highest confidence is assumed to be responsible for the observed reward. This results in a reinforcement learning scheme under which

$$Q_{t+1}(i) = \begin{cases} Q_t(i) + \alpha \left\{R_t - Q_t(i)\right\} & \text{if } i = s_t, \\ Q_t(i) & \text{otherwise.} \end{cases} \tag{3.1}$$

Note that this is identical to the basic scheme, equation 2.1, except that the update is applied to $Q_t(s_t)$ instead of $Q_t(i_t)$. One could ask if the incorrect reward allocations introduced as a result of this strategy will average out, resulting in correct estimates $Q_t(i)$, albeit with higher variance than when the state information is unambiguous.

To address this question, consider a situation where there are two states of the environment, so that $\mathbb{P}(s_t = 1) = \mathbb{P}(s_t = 2) = \frac{1}{2}$ independently of all other random variables. The expected change of $Q_t(1)$ is given by

$$\begin{aligned} &\mathbb{E}[Q_{t+1}(1) - Q_t(1) \mid \boldsymbol{Q}_t] \\ &\quad = \mathbb{E}\left[\mathbb{I}_{\{s_t=1\}}\alpha \left\{R_t - Q_t(1)\right\} \mid \boldsymbol{Q}_t\right] \\ &\quad = \alpha\mathbb{P}(s_t = 1)\left\{\mathbb{E}[R_t \mid s_t = 1] - Q_t(1)\right\} \\ &\quad = \tfrac{\alpha}{2}\left\{\rho\mu(1) + (1-\rho)\mu(2) - Q_t(1)\right\}. \end{aligned}$$

Hence the stochastic fixed point of $Q_t(1)$ is $Q_\infty(1) = \rho\mu(1) + (1-\rho)\mu(2)$. Similarly, the stochastic fixed point of $Q_t(2)$ is $Q_\infty(2) = \rho\mu(2) + (1-\rho)\mu(1)$. This linear dependence of the fixed point $Q_\infty(i)$ on the confidence level ρ is illustrated in Figure 1. Note that unless either $\mu(1) = \mu(2)$ or $\rho = 1$, it is not the case that $Q_\infty(i) = \mu(i)$, so if this algorithm converges, it will not be

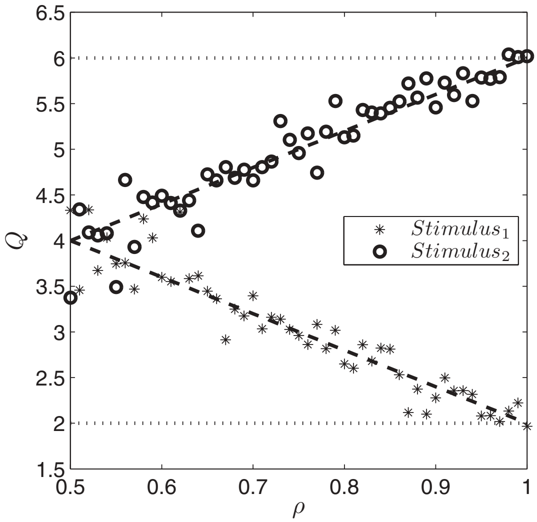


Figure 1: Final estimates found by winner-takes-all reinforcement learning in a two-state environment with different confidence levels. The circles show the final estimates for state 1 with actual expected reward 6 and the stars show the final estimates for state 2 with actual expected reward 2 (see text for simulation details). The dashed lines show the theoretically-calculated stochastic fixed points, and the dotted lines show the true state values.

to correct estimates of the action values. In the general situation with $N > 2$ states, an equivalent calculation shows that

$$Q_\infty(i) = \rho\mu(i) + \frac{1-\rho}{N-1} \sum_{j=1;\, j\neq i}^{N} \mu(j).$$

To illustrate the performance of this algorithm, consider the following simple experimental setup with $N = 2$ states. The rewards in state 1 have a normal distribution with expected value $\mu(1) = 6$ and variance 1, whereas the rewards in state 2 have a normal distribution with expected value $\mu(2) = 2$ and variance 1. A learning episode consists of 2000 iterations of the learning algorithm, with initial values $Q_1(1) = Q_1(2) = \frac{1}{2}(\mu(1) + \mu(2))$. The learning parameter α is taken to be 0.05 throughout. Figure 1 shows the final estimates in 50 learning episodes, with a different confidence level $\rho \in [0.5, 1]$ for each trial. The stochastic fixed points are also plotted. It is clear that the experimental results correspond well with the theory, but neither results nor theory match the correct estimates except when $\rho = 1$.

3.2 Confidence Weighted Reinforcement Learning. Clearly the winner-takes-all procedure is inadequate, since it simply estimates a weighted average of the rewards for the different states. However, in section 2 we noted that the learner may be able to estimate their confidence level ρ. Hence, it may be possible to alter the learning rule used in response to this confidence level.

A simple approach that could be taken to incorporate the confidence into learning is to weight the update rule for $Q_t(i)$ with the confidence in the stimulus:

$$Q_{t+1}(i) = Q_t(i) + \alpha\rho_t(i)\left\{R_t - Q_t(i)\right\} \quad \text{for } i = 1, \ldots, N. \tag{3.2}$$

Note that if $\rho < 1$, the estimate $Q_t(i)$ is updated for each state in every trial. However, this scheme is the same as that in equation 2.1 if $\rho = 1$.

Consider again the experimental setup of the previous section with only two states, each of which is sampled with equal probability independently at each trial. We see that

$$\begin{aligned}
&\mathbb{E}[Q_{t+1}(1) - Q_t(1) \mid \boldsymbol{Q}_t] \\
&= \tfrac{1}{2}\left\{\mathbb{E}[Q_{t+1}(1) - Q_t(1) \mid \boldsymbol{Q}_t, s_t = 1] + \mathbb{E}[Q_{t+1}(1) - Q_t(1) \mid \boldsymbol{Q}_t, s_t = 2]\right\} \\
&= \tfrac{1}{2}\left\{\mathbb{E}[\alpha\rho\{R_t - Q_t(1)\} \mid \boldsymbol{Q}_t, s_t = 1] + \mathbb{E}[\alpha(1-\rho)\left\{R_t - Q_t(1)\right\} \mid \boldsymbol{Q}_t, s_t = 2]\right\} \\
&= \tfrac{\alpha}{2}\left\{\rho\mathbb{E}[R_t \mid s_t = 1] + (1-\rho)\mathbb{E}[R_t \mid s_t = 2] - Q_t(1)\right\} \\
&= \tfrac{\alpha}{2}\left\{\rho[\rho\mu(1) + (1-\rho)\mu(2)] + (1-\rho)[(1-\rho)\mu(1) + \rho\mu(2)] - Q_t(1)\right\} \\
&= \tfrac{\alpha}{2}\left\{(1 - 2\rho + 2\rho^2)\mu(1) + 2\rho(1-\rho)\mu(2) - Q_t(1)\right\}.
\end{aligned}$$

Equating this to 0 shows that the stochastic fixed point of $Q_t(1)$ is

$$Q_\infty(1) = (1 - 2\rho + 2\rho^2)\mu(1) + 2\rho(1-\rho)\mu(2).$$

There is a similar solution for $Q_\infty(2)$. In the general case with $N > 2$ states, an equivalent calculation shows that the quadratic dependence on ρ is retained.

Figure 2 demonstrates this quadratic dependence on ρ, both theoretically and using the same experimental setup as for Figure 1. Perhaps surprisingly, this more sophisticated approach, which takes the uncertainty into account, results in estimates that are even further from the correct values than those achieved by simply ignoring the fact that the state information is noisy.

3.3 Posterior Weighted Reinforcement Learning. In the previous section, we saw that a simple attempt to use the confidence level in deciding allocation of reward to states resulted in worse performance than simply ignoring the uncertainty. However, at the point at which the allocation of the reward to states is made, there is more information available to the

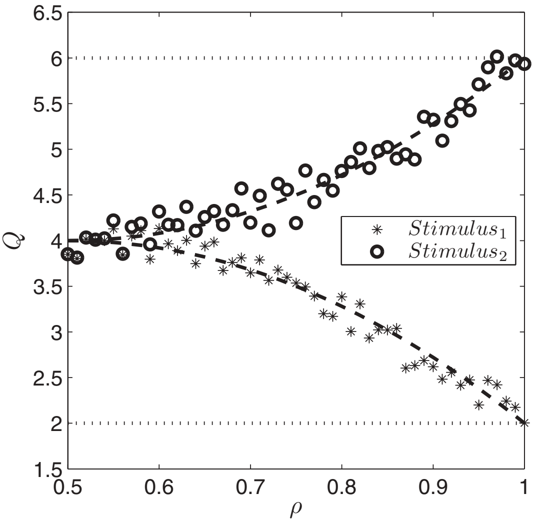


Figure 2: Final estimates found by confidence-weighted reinforcement learning in a two-state environment with different confidence levels. The circles show the final estimates for state 1 with actual expected reward 6, and the stars show the final estimates for state 2 with actual expected reward 2. The dashed lines show the theoretically calculated stochastic fixed points, and the dotted lines show the true state values.

learner than simply the probability distribution $\boldsymbol{\rho}_t = (\rho_t(1), \ldots, \rho_t(N))$. In particular the reward has also been observed, and this provides additional information about the true state. In this section, we introduce a new technique that weights the update to the estimate $Q_t(i)$ with the resulting posterior probability that the state is i.

To calculate a posterior probability, the learner must have a model for the distribution of the rewards. If the state is i and the estimate of the expected reward in state i is $Q(i)$, then the probability density of the reward is given by $f(r; i, Q(i))$. (We use the language of continuous random variables here, although the probability mass function can be substituted directly in the case of discrete reward distributions.) Since the prior probability (i.e., the probability after stimulus observation but before the reward delivery) that the state was i is $\rho_t(i)$, the posterior probability that the state is i once the reward R_t has been observed is given by Bayes' rule:

$$\mathbb{P}(i_t = i \mid R_t, \boldsymbol{Q}_t, \boldsymbol{\rho}_t) = \frac{f(R_t; i, Q_t(i))\rho_t(i)}{\sum_{j=1}^{N} f(R_t; j, Q_t(j))\rho_t(j)}. \tag{3.3}$$

Our proposed posterior weighted reinforcement learning (PWRL) scheme is similar to the confidence-weighted scheme in equation 3.2, but now the weights are simply the posterior confidence levels once the reward has been observed instead of the prior confidence levels that do not incorporate this extra information. The update equation becomes

$$Q_{t+1}(i) = Q_t(i) + \alpha \frac{f(R_t; i, Q_t(i))\rho_t(i)}{\sum_{j=1}^{N} f(R_t; j, Q_t(j))\rho_t(j)} \left\{R_t - Q_t(i)\right\} \quad \text{for } i = 1, \ldots, N. \tag{3.4}$$

Note that given $\boldsymbol{\rho}_t$, the density of the reward is $\sum_{j=1}^{N} f(r; j, \mu(j))\rho_t(j)$. Hence, the expected change in $Q_t(i)$ is given by

$$\begin{aligned}
&\mathbb{E}[Q_{t+1}(i) - Q_t(i) \mid \boldsymbol{Q}_t, \boldsymbol{\rho}_t] \\
&\quad = \alpha \int \left[\frac{f(r; i, Q_t(i))\rho_t(i)}{\sum_{j=1}^{N} f(r; j, Q_t(j))\rho_t(j)} \left\{r - Q_t(i)\right\} \right] \\
&\quad\quad \times \left[\sum_{j=1}^{N} f(r; j, \mu(j))\rho_t(j) \right] \mathrm{d}r.
\end{aligned} \tag{3.5}$$

If $Q_t(i) = \mu(i)$ for all $i = 1, \ldots, N$, we find that

$$\mathbb{E}[Q_{t+1}(i) - Q_t(i) \mid \boldsymbol{Q}_t = \boldsymbol{\mu}, \boldsymbol{\rho}_t] = \alpha\rho_t(i) \int f(r; i, \mu(i)) \left\{r - \mu(i)\right\} \mathrm{d}r = 0.$$

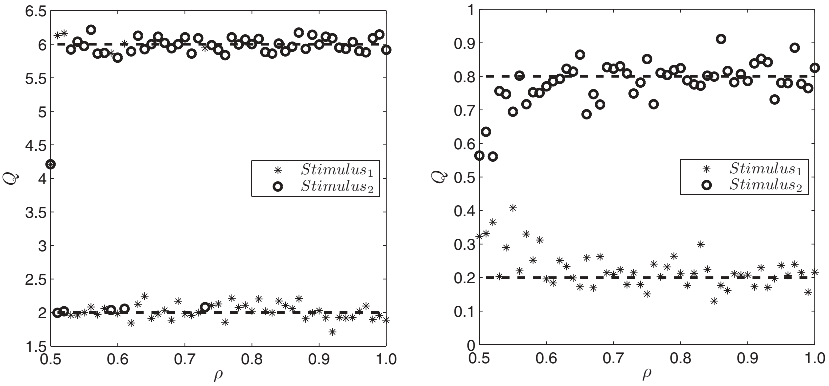
Hence $Q_\infty(i) = \mu(i)$ for $i = 1, \ldots, N$ is a stochastic fixed point of the system.

For normal random variables with unit variance, the update scheme reduces to

$$Q_{t+1}(i) = Q_t(i) + \alpha \frac{\phi(R_t - Q_t(i))\rho_t(i)}{\sum_{j=1}^{N} \phi(R_t - Q_t(j))\rho_t(j)} \left\{R_t - Q_t(i)\right\} \quad \text{for } i = 1, \ldots, N, \tag{3.6}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$ denotes the standard normal density function.

A particularly biologically relevant example is the case of Bernoulli random variables, where the learner gets a unit reward with probability $\mu(i)$ and no reward with probability $1 - \mu(i)$ (where $\mu(i) \in (0, 1)$). Bernoulli rewards provide less information than normal rewards (since an observation is merely the presence or absence of reward instead of a reward value),



(a) (b)

Figure 3: Final estimates found by posterior weighted reinforcement learning in a two-state environment with different confidence levels. (a) Rewards were sampled from normal distributions with means at 2 and 6 (b) Rewards were sampled from Bernoulli distributions with means at 0.2 and 0.8. The dashed lines show the theoretically calculated stochastic fixed points, which in this case coincide with the true state values.

so this may be considered a particularly difficult case in which to estimate values. In this case, the update for each i is

$$Q_{t+1}(i) = \begin{cases} Q_t(i) + \alpha \frac{Q_t(i)\rho_t(i)}{\sum_{j=1}^{N} Q_t(j)\rho_t(j)} \left\{1 - Q_t(i)\right\} & \text{if } R_t = 1, \\ Q_t(i) + \alpha \frac{(1-Q_t(i))\rho_t(i)}{\sum_{j=1}^{N}(1-Q_t(j))\rho_t(j)} \left\{0 - Q_t(i)\right\} & \text{if } R_t = 0. \end{cases} \tag{3.7}$$

The results for the same experiments as presented in the previous two sections are shown in Figure 3a, where it is clearly seen that the final estimates in this case are correct at most confidence levels ρ. Errors occur only for low ρ values, where the final estimates are sometimes swapped: $Q_{\text{final}}(1) \approx \mu(2)$ and $Q_{\text{final}}(2) \approx \mu(1)$ (e.g., note the stars in the top left corner of Figure 3a). This happens when the stimulus is incorrectly identified in early trials, and subsequently the likelihood terms $f(r; i, Q(i))$ dominate the prior values $\rho(i)$. We discuss this more fully in the next section and the appendix. Figure 3b shows similar results but for Bernoulli rewards, where $\mu(1) = 0.8$, $\mu(2) = 0.2$. In this experiment we take $\alpha = 0.02$ since the absolute value of the estimates is much lower than in the normal case. It is clear that the estimates also converge to the correct values in this more difficult example.

Note that to calculate the posterior probability $\mathbb{P}(i_t = i \mid R_t = r, \boldsymbol{Q}_t, \boldsymbol{\rho}_t)$ requires the choice of a parameterized reward model $f(r; i, \mu(i))$ for each state of the environment (the model will usually be the same for each source, but there is no theoretical need for this restriction). In the appendix we show that even if the learner uses an incorrect model of reward distribution, the learned distribution will be close to the true distribution, in the sense that it will minimize the expected Kullback–Leibler divergence from the true data-generating distribution. This provides robustness to the choice of probability model.

4 Switching Reward Distributions

It is important for animals to be able to respond to a change in the environment. In the case of a reinforcement learning task, this corresponds to the reward distributions changing partway through the learning process. In this section, we consider performance in a learning episode with two states, in which the reward distributions are normal with variance 1 and means 2 and 6, depending on which state is presented. For the first 500 iterations, the reward in state 1 has expected value 2, and the reward in state 2 has expected value 6. On iterations 501 through to the end of the episode at iteration 2000, these switch, so that in state 1, the expected reward is 6, whereas in state 2, the expected reward is 2. (A similar switch is carried out in the experiment with Bernoulli rewards.) Traditional reinforcement learning models, including those of sections 3.1 and 3.2, should respond to this switch and adjust their estimates to the new values by the end of the episode (although recall that with state uncertainty, these estimates will not be correct). However, we will see that the formulation of posterior weighted reinforcement learning in section 3.3 can suffer from difficulties. This is because the allocation of observations to states depends on the current estimates $\boldsymbol{Q}_t$, and the likelihood terms $f(r; i, Q_t(i))$ can dominate the prior confidence levels $\rho_t(i)$, particularly in the case of light-tailed distributions such as the normal distribution. If this occurs, the learner will continue to have high posterior probability that rewards $R_t \approx Q_t(i)$ are from state i, reqardless of the prior information $\boldsymbol{\rho}_t$.

The experimental results in this case are shown in Figure 4. It is clearly seen that the winner-takes-all and confidence-weighted schemes respond to the switch in reward distributions (although they do not converge to the correct estimate). In contrast, the PWRL scheme fails to switch when the rewards are sampled from normal distributions, although in the case of the Bernoulli distributions, the reward estimates successfully switch.

Consider the following example, which explains the failure to switch. Assume that the correct estimates have been learned in the first 500 iterations, so that $Q_{500}(1) = 2$ and $Q_{500}(2) = 6$. The means then switch. Suppose on the 501th iteration, we have $\rho_t(1) = 0.9 = 1 - \rho_t(2)$ and, further, that $R_t = 6$ (corresponding to the true state being 1, and the reward being exactly the new expected reward for that state). The posterior probability that the state

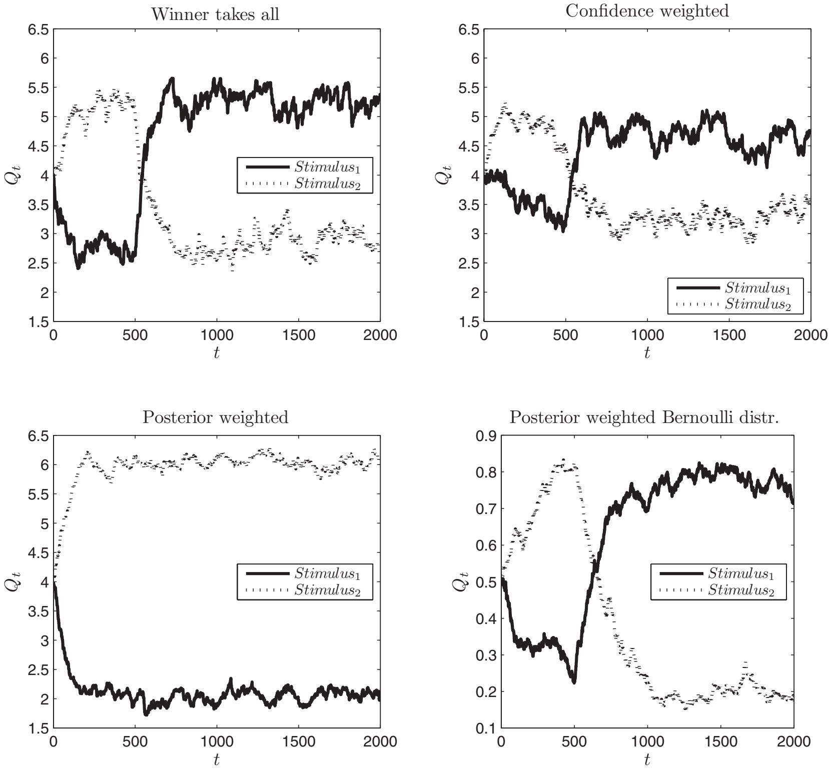


Figure 4: Estimates over trials when rewards switch after trial 500. Each panel shows estimates from a different model, labeled at the top of the panel. Each model is simulated with confidence $\rho = 0.8$. In all panels except the bottom right panel, the rewards were sampled from normal distributions with means 2 and 6 and the learning parameter $\alpha = 0.05$, while in the bottom right panel, the rewards were sampled from Bernoulli distributions with means 0.2 and 0.8 and the learning parameter $\alpha = 0.02$.

is 1 is actually given by

$$\frac{\rho_t(1)\phi(R_t - Q_t(1))}{\rho_t(1)\phi(R_t - Q_t(1)) + \rho_t(2)\phi(R_t - Q_t(2))} = \frac{0.9\times1.34\times10^{-4}}{0.9\times1.34\times10^{-4} + 0.1\times0.399}$$
$$= 0.003. \qquad (4.1)$$

The very small likelihood value for state 1 given current estimates (1.34×10^{-4}) means that the prior confidence is essentially irrelevant.

This effect is less significant in the case of Bernoulli random variables, since the likelihood is not exponentially decreasing (as shown in Figure 4, in this case the reward estimates switch to the correct values). The problem could be somewhat alleviated in the continuous case by the use of heavy-tailed distributions, such as the t distribution, instead of normal distributions in the calculation of the posterior allocation probabilities.

Instead we focus on a different approach and additionally estimate variance terms for the normal distributions. When received rewards differ significantly from the predictions, we expect that the variance estimates will become large, thus making the likelihood less light-tailed and allowing the prior to direct the allocation of rewards to states. The update equations become:

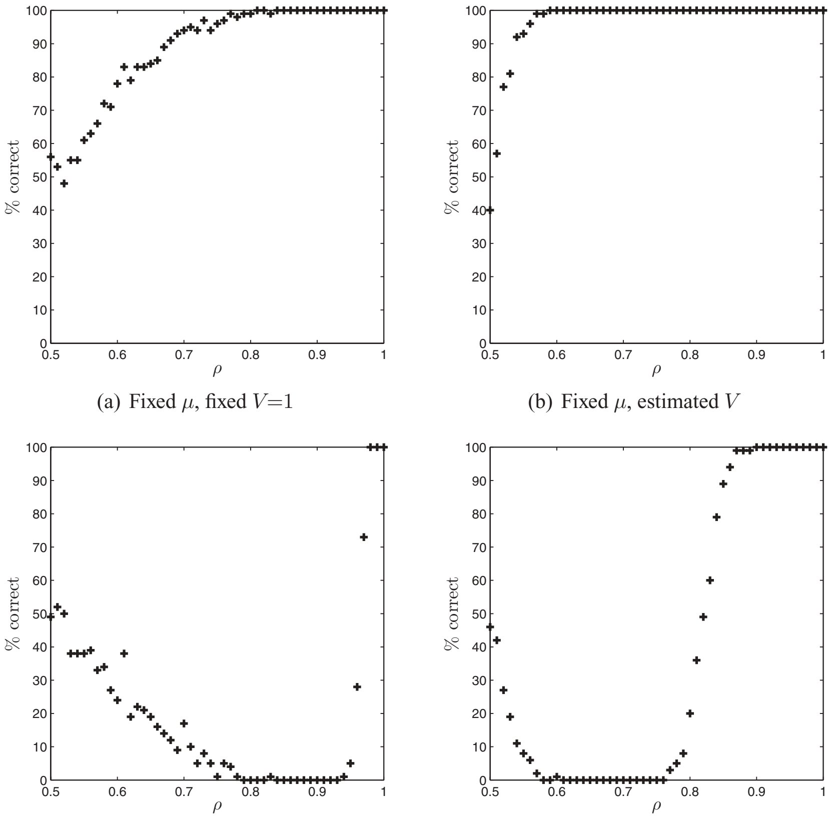
$$Q_{t+1}(i) = Q_t(i) + \alpha \mathbb{P}(i_t = i \mid R_t, \boldsymbol{\rho}_t, \boldsymbol{Q}_t, \boldsymbol{V}_t) \left\{ R_t - Q_t(i) \right\},$$

$$V_{t+1}(i) = V_t(i) + \alpha \mathbb{P}(i_t = i \mid R_t, \boldsymbol{\rho}_t, \boldsymbol{Q}_t, \boldsymbol{V}_t) \left\{ (R_t - Q_t(i))^2 - V_t(i) \right\},$$

where

$$\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\rho}_t, \boldsymbol{Q}_t, \boldsymbol{V}_t) = \frac{\rho_t(i)\phi((R_t - Q_t(i))/\sqrt{V_t(i)})/\sqrt{V_t(i)}}{\sum_{j=1}^{N} \rho_t(j)\phi((R_t - Q_t(j))/\sqrt{V_t(j)})/\sqrt{V_t(j)}}.$$

With this enhancement, the PWRL scheme is significantly less likely to make the initial allocation mistakes observed in Figure 3, since it estimates large variances early in the learning episode, and the prior confidence levels $\boldsymbol{\rho}_t$ have more influence (compare Figures 5a and 5b). It also handles switches in the rewards more successfully than the original PWRL scheme (compare Figures 5c and 5d) although it still fails for low ρ. Note that the switch of states is a particularly problematic scenario for the PWRL scheme; if one or both states changed so that their distribution was completely different to a current state reward distribution, then the likelihoods for both states will be small, and this would allow the prior to have more influence, whereas simply switching the state distributions means that the likelihood for the incorrect state is high, while the likelihood for the correct state is very low. If the $\mu(i)$ change gradually, the PWRL algorithm is able to track the changes much more easily. We note in the appendix that swapped estimates (with $Q(2) = \mu(1)$ and $Q(1) = \mu(2)$) correspond to a local minimum of a potential function for the PWRL algorithm. This explains both why convergence to this point can occur and why switching the reward distributions is a particularly difficult scenario for PWRL.



(a) Fixed μ, fixed V=1

(b) Fixed μ, estimated V

(c) Switching μ, fixed V=1

(d) Switching μ, estimated V

Figure 5: The percentage of correct final estimates in a two-state environment for different confidence levels. For each confidence level, simulations with 2000 trials were repeated 100 times. The crosses indicate the fraction of simulations in which the final estimate $Q_{2000}(1)$ was closer to $\mu(1)$ than to $\mu(2)$. In all simulations, rewards were sampled from normal distributions with means 2 and 6, and with variance 1, and the learning parameter was set to $\alpha = 0.05$. In panels *a* and *b* the means did not change over trials, while in panels *c* and *d* the means were switched after trial 500. Panels *a* and *c* show results for PWRL without variance estimation, while panels *b* and *d* show results for PWRL with variance estimation.

5 Expectation-Maximization

In this section, we return to reward distributions that are fixed through time and relate our algorithm to a standard statistical procedure. The problem posed in this letter is the estimation of some parameters (the expected reward in each state) in the presence of unobserved information (the true state when each observation is made). A standard algorithm for the general problem of parameter estimation in the presence of unobserved information is the EM algorithm (Dempster et al., 1977). We will show that the PWRL scheme is closely related to an online version of the EM algorithm (Titterington, 1984; Wang & Zhao, 2006; Cappé & Moulines, 2009) designed for estimation when data arrive incrementally, as in the reinforcement learning problem. This allows us, in the appendix, to give convergence results for the PWRL scheme. It also shows how to apply the PWRL algorithm in a principled manner to reward distributions other than those considered in this letter.

We now consider a general probability model $f(r;\theta(i))$ for the reward distribution in state i, with $\theta(i)$ a parameter vector that may have more than one entry (thus allowing us to use a unified notation for the algorithms based on the mean, the algorithm that also estimates the variance, and further generalizations). We write $p(r,i;\boldsymbol{\theta},\boldsymbol{\rho}) = \rho(i) f(r;\theta(i))$ for the joint density of state and reward given parameters $\boldsymbol{\theta} = (\theta(1),\ldots,\theta(N))$ and confidence vector $\boldsymbol{\rho}$, and write $p(r;\boldsymbol{\theta},\boldsymbol{\rho}) = \sum_{i=1}^{N} p(r,i;\boldsymbol{\theta},\boldsymbol{\rho})$ for the density of R.

Titterington (1984) suggests the following modified Fisher scoring algorithm,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\,[I_c(\boldsymbol{\theta}_t,\boldsymbol{\rho}_t)]^{-1}\,\nabla_{\boldsymbol{\theta}_t}\log(p(R_t;\boldsymbol{\theta}_t,\boldsymbol{\rho}_t)), \tag{5.1}$$

and relates it to the EM algorithm. The complete data Fisher information matrix I_c used in this update is given by

$$I_c(\boldsymbol{\theta},\boldsymbol{\rho}) = -\mathbb{E}\left[\nabla^2_{\boldsymbol{\theta}}\log(p(R,i;\boldsymbol{\theta},\boldsymbol{\rho}))\right].$$

In the case of normal random variables with unit variance, where $\theta(i)$ is the expected value in state i, we have $p(r,i;\boldsymbol{\theta},\boldsymbol{\rho}) = \rho(i)\frac{1}{\sqrt{2\pi}}e^{-(r-\theta(i))^2/2}$. Hence,

$$\begin{aligned}\frac{\partial}{\partial\theta(i)}\log(p(r;\boldsymbol{\theta},\boldsymbol{\rho})) &= \frac{p(i,r;\boldsymbol{\theta},\boldsymbol{\rho})}{p(r;\boldsymbol{\theta},\boldsymbol{\rho})}\left\{r-\theta(i)\right\}\\ &= \mathbb{P}(i_t = i \mid R_t = r, \boldsymbol{\theta}_t = \boldsymbol{\theta}, \boldsymbol{\rho}_t = \boldsymbol{\rho})\left\{r-\theta(i)\right\},\end{aligned}$$

and $I_c(\boldsymbol{\theta},\boldsymbol{\rho})$ is a diagonal matrix with diagonal equal to the confidence vector $\boldsymbol{\rho}$. Titterington's update, equation 5.1, becomes

$$\theta_{t+1}(i) = \theta_t(i) + \alpha\frac{1}{\rho_t(i)}\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}_t, \boldsymbol{\rho}_t)\left\{R_t - \theta_t(i)\right\}. \tag{5.2}$$

This update is identical to the PWRL update, equation 3.6, but with the temporal difference divided by the confidence level $\rho_t(i)$.

In the case of Bernoulli random variables, where $\theta(i)$ is the probability of reward in state i, we have $p(r, i; \boldsymbol{\theta}, \boldsymbol{\rho}) = \rho(i)\theta(i)^r(1 - \theta(i))^{1-r}$. Hence, recalling that r is either 0 or 1,

$$\frac{\partial}{\partial\theta(i)} \log(p(r; \boldsymbol{\theta}, \boldsymbol{\rho})) = \frac{\rho(i)(-1)^{r+1}}{p(r; \boldsymbol{\theta}, \boldsymbol{\rho})},$$

and $I_c(\boldsymbol{\theta}, \boldsymbol{\rho})$ is a diagonal matrix with iith entry:

$$\frac{\rho(i)}{\theta(i)(1 - \theta(i))}$$

Plugging these into Titterington's formula, equation 5.1, gives

$$\begin{aligned}\theta_{t+1}(i) &= \theta_t(i) + \alpha \frac{1}{\rho_t(i)} \frac{\rho_t(i)(-1)^{r-1}\theta_t(i)(1 - \theta_t(i))}{p(R_t; \boldsymbol{\theta}_t, \boldsymbol{\rho}_t)} \\ &= \theta_t(i) + \alpha \frac{1}{\rho_t(i)} \mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}_t, \boldsymbol{\rho}_t) \left\{R_t - \theta_t(i)\right\}. \end{aligned} \tag{5.3}$$

This is again the PWRL update, equation 3.6, but with the temporal difference divided by $\rho_t(i)$.

In the appendix we show that the PWRL update is related to Titterington's method for a wide class of probability distributions. In particular we show that for all distributions in the exponential family (Barndoff-Nielsen, 1978) that are parameterized by the mean, Titterington's method results in the same update as equations 5.2 and 5.3.

Note that a recognized problem with Titterington's method is that division by $\rho_t(i)$ may take the estimates out of the valid parameter space (e.g., the probability of a reward in the Bernoulli case may be estimated to be negative or greater than 1). The PWRL scheme, by choosing not to divide by the prior confidence level, removes this problem. We show in the appendix that convergence proofs are still valid in the presence of this modification.

6 Possible Neural Implementation of PWRL

To implement PWRL based on the update in equation 3.4, the posterior state probabilities need to be computed on the basis of the prior probabilities (i.e., the estimates based on stimulus) and the reward value. Recently Bogacz (2009) proposed that the cortico-basal-ganglia-thalamic circuit performs an analogous computation during perceptual decisions in which the information on the identity of noisy stimuli needs to be gathered over time. In particular, he proposed that when a new piece of information on stimulus identity arrives, this circuit computes the posterior probabilities of stimuli on the basis of the prior probabilities (i.e., the estimates based on information obtained earlier within a choice trial) combined with the new piece of information.

In this section, we propose a possible neural implementation of PWRL. We first write the posterior probability, equation 3.3, in a form easier for biological implementation. Then we review the model of cortico-basal-ganglia-thalamic circuit (Bogacz, 2009) and show how it could compute these posterior probabilities. Finally, we discuss how this may allow for the updating of the $Q_t(i)$ in equation 3.4.

We start by taking logarithms of both sides in equation 3.3 and rewriting it in the equivalent form:

$$\log \mathbb{P}(i_t = i \mid R_t,\ \boldsymbol{Q}_t, \boldsymbol{\rho}_t) = Y(i) - \log \left\{ \sum_{j=1}^{N} \exp Y(j) \right\}, \tag{6.1}$$

where

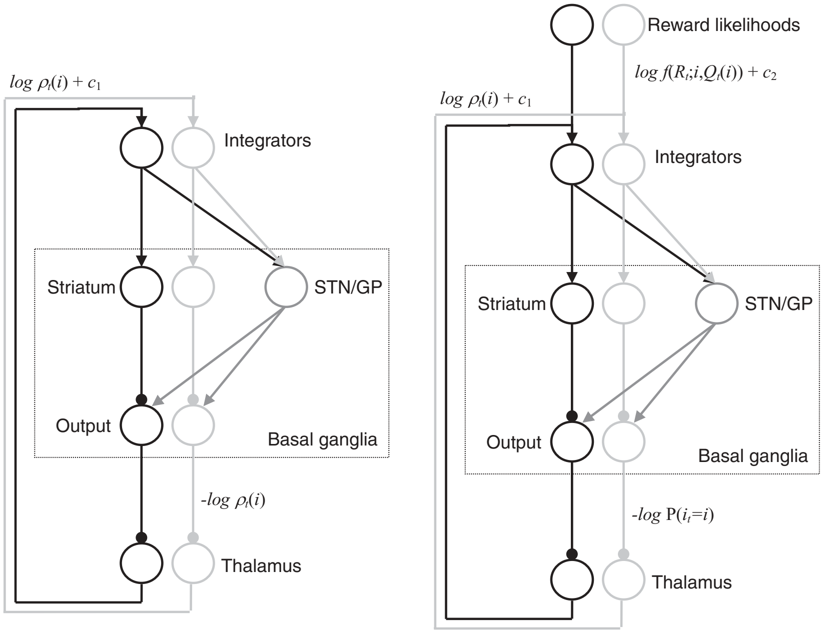
$$Y(i) = \log \rho_t(i) + \log f(R_t; i,\ Q_t(i)). \tag{6.2}$$

This makes the computation of the logarithm of the posterior probability actually quite simple: one needs to add $\log f(R_t; i,\ Q_t(i))$ to the logarithm of the corresponding prior probability and then normalize by subtracting the expression given in the second term of equation 6.1.

We will demonstrate that this computation can be performed in a model of the cortico-basal-ganglia-thalamic circuit (Bogacz, 2009). Its basic architecture, shown in Figure 6a, includes cortical integrators (that accumulate the information on stimulus identity), basal ganglia, and thalamus connected in a loop. The integrators also receive input from sensory neurons that provide information on stimuli, but these inputs are not shown in Figure 6a for simplicity; the integrators add the new input from the sensory neurons to the thalamic feedback. Within each area included in the model are neuronal populations selective for different stimuli indicated by different shades in Figure 6a. This is a system-level model that describes the activity levels of neuronal populations rather than individual neurons, and it includes only a subset of known connectivity of this circuit. We will demonstrate the computations in the circuit at three points in time: before reward delivery, at the time of the reward delivery, and after reward delivery.

Let us denote the activity of a population of cortical integrator neurons selective for stimulus i by $y(i)$. Bogacz (2009) considers a model without any rewards, which corresponds to the first time point—before a reward has been delivered. He shows that after stimulus presentation, the activities of cortical integrators are proportional to the logarithms of the estimated state probabilities. Thus, after stimulus presentation and before reward delivery, we have

$$y(i) = \log \rho_t(i) + c_1. \tag{6.3}$$



(a) (b)

Figure 6: Cortico-basal-ganglia-thalamic circuit that could compute the posterior probabilities in PWRL. Black and gray circles denote neural populations selective for the first and the second stimuli. Arrows denote excitatory connections, and lines ending with circles denote inhibitory connections. STN: subthalamic nucleus. GP: globus pallidus in rodents or globus pallidus external segment in primates. Output: output nuclei of the basal ganglia: substantia nigra pars reticulate and entopeduncular nucleus in rodents or globus pallidus internal segment in primates. (a) The state of the network between stimulus offset and reward delivery. (b) The state when reward information is provided.

(The constant c_1 is added to make this expression positive because a probability is, by definition, less than or equal to 1 so $\log \rho_t(i) \leq 0$.)

The basal ganglia are modeled as in Bogacz and Gurney (2007). In particular, the total activity of the subthalamic nucleus is proportional to

$$STN = \log\left\{\sum_{j=1}^{N} \exp y(j)\right\} \tag{6.4}$$

$$= \log\left\{\sum_{j=1}^{N} \rho_t(j) \exp c_1\right\} = c_1. \tag{6.5}$$

Bogacz and Gurney (2007) describe in detail how equation 6.4 is computed in a model of a network including the subthalamic nucleus and globus pallidus. They argue that existing neurobiological data suggest that these nuclei have suitable patterns of connectivity and input-output transfer functions to perform the calculation. The result in equation 6.5 comes from substituting equation 6.3 and noting that $\sum_{j=1}^{N} \rho_t(j) = 1$.

The output nuclei in the model receive inhibition from cortical integrators via the striatum and excitation from the subthalamic nucleus, so that

$$\begin{aligned} OUT(i) &= -y(i) + STN \\ &= -\log \rho_t(i). \end{aligned} \tag{6.6}$$

In the Bogacz (2009) model, the thalamus receives inhibition from the output nuclei and constant excitatory input c_1, so the activities of the thalamic units are $\log \rho_t(i) + c_1$ (as indicated by labels in Figure 6a). Hence, the inputs to the cortical integrators from the thalamus are equal to the integrators' original levels of activity, and so these levels are maintained.

We now show that when the reward information is provided to the integrators, the estimated state probabilities are updated as in PWRL. Labels in Figure 6b illustrate the state of the network at the second time point—the moment of reward delivery. The feedback from the thalamus is still proportional to the logarithm of the prior probabilities. We now hypothesize that certain neuronal populations are able to calculate $\log f(R_t; i, Q_t(i)) + c_2$, where the constant c_2 is added to make the values positive. (We come back to the plausibility of calculating the logarithm of the likelihoods below.) We label these neuronal populations "reward likelihoods" in Figure 6b and assume that they project to the integrators. If the integrators treat these inputs the same as inputs from sensory neurons, and therefore add these reward likelihoods to the thalamic feedback, their activities become

$$y(i) = \log \rho_t(i) + c_1 + \log f(R_t; i, Q_t(i)) + c_2 = Y(i) + c_1 + c_2 \tag{6.7}$$

where $Y(i)$ is as defined in 6.2. Substituting 6.7 into 6.4 we get

$$STN = c_1 + c_2 + \log \left\{ \sum_{j=1}^{N} \exp Y(j) \right\}. \tag{6.8}$$

Substituting 6.7 and 6.8 into 6.6 and using 6.1 we get

$$OUT(i) = -Y(i) + \log \sum_{j=1}^{N} \exp\ Y(j) = -\log\ \mathbb{P}(i_t = i \mid R_t,\ \mathbf{Q}_t, \boldsymbol{\rho}_t).$$

Hence the activities of the thalamic units become proportional to the logarithms of the posterior probabilities.

At the third time point, after the reward has been received and the new thalamic feedback arrives, we assume that the cortical integrators do not receive any other input. Hence, for the same reasons as with the prior probabilities, the logarithms of the posterior state probabilities are maintained in the circuit.

Current reinforcement learning theories usually assume that the $Q_t(i)$ are represented in synaptic weights of cortico-striatal synapses and that the weights between coactive cortical and striatal neurons are modified proportionally to the prediction error $(R_t - Q_t(i))$ represented by the concentration of dopamine released in the striatum (see Doya, 2007, for a review). Since only the weights between active cortical and striatal neurons are modified, the magnitude of the $Q_t(i)$ modification depends on the activity of cortical neurons. Furthermore since, in the model, these activities are proportional to the logarithms of the posterior probabilities, these probabilities may influence the magnitude of $Q_t(i)$ modification, as needed for the PWRL model.

The element of the above model with the least clear neural basis is the computation of $\log f(R_t; i, Q_t(i))$ by the units labeled "reward likelihoods" in Figure 6b. We do not want to speculate how such computation could be performed, except to point out that this expression is not as difficult to compute as it may seem. For example, for normally distributed rewards (with unit variance),

$$\log f(R_t; i, Q_t(i)) = \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2}(R_t - Q_t(i))^2.$$

Note that the first term is the same for all i, so it can be incorporated into constant c_2 (the precise value of this constant is unimportant, as it cancels out). Thus the "reward likelihoods" units only need to have activities that relate to the squares of the prediction errors for each of the stimuli. Furthermore, for the case of Bernoulli rewards, the calculation is even simpler:

$$\log f(R_t; i, Q_t(i)) = \begin{cases} \log(Q_t(i)) & \text{if } R_t = 1, \text{and} \\ \log(1 - Q_t(i)) & \text{if } R_t = 0. \end{cases}$$

Since we already postulate that the Q values can be stored and logarithms can be calculated, this requires no additional processing capability.

7 Discussion

We have shown that in the presence of ambiguous state signals the reinforcement learning problem is not straightforward. Simply allocating reward to the most likely state of nature results in incorrect value estimates.

Furthermore, taking account of state uncertainty in a simple way can result in even worse estimates. Weighting the allocation of reward to states using the posterior probability that a state was the true state, given both the prior confidence level and the observed reward, results in correct value estimates in the settings considered in this letter. In this section, we discuss approaches to related problems and address some areas for further research.

7.1 Alternative Approaches to Noisy and Switching Environments.

When the reward distributions switch during learning, the PWRL approach can fail to react. This is especially true if a light-tailed distribution is used to model the rewards, in which case the prior can have virtually no influence over the posterior probabilities. We introduced variance estimation to give PWRL an opportunity to react by adjusting the variance estimates. However, for low confidence levels, we observed that PWRL with variance estimation is still unable to correctly learn the new rewards after they switch. In particular, for the problematic example described in section 4, the calculation in equation 4.1 remains correct even when variances are estimated.

Other work has been carried out to investigate learning rewards under changing environments. Behrens, Woolrich, Walton, & Rushworth (2007) studied a Bayesian learner to predict how the learning rate should be modified based on the rate of change of the underlying state rewards (the volatility of the environment). Their results have the opposite requirement to that of PWRL: under larger uncertainty, the learning rate should increase, not decrease. The reasoning is that if a subject is confident that the environment is not changing, random fluctuations should not be a concern, and static behavior through low learning rate should therefore be promoted, whereas if the environment is changing, a larger learning rate is more appropriate. However, this is valid only if the uncertainty is about the underlying expected reward, and not if the uncertainty is about the underlying state, and their experiment was indeed based on choosing unambiguously colored rectangles. In spite of the differences between PWRL and the volatility model, both rely on estimating the variance or volatility and modifying their learning rate accordingly.

Alternative solutions, addressing both types of uncertainty, will be sought in future work. As well as the Behrens et al. (2007) approach, this may include some notion of state dynamics by incorporating a prior belief that, with some fixed probability, the reward distributions may change, as in Yu and Cohen (2009), or incorporating the expected and unexpected uncertainty framework of Yu and Dayan (2003). There might also be a need to extend this framework to incorporate other uncertainties such as stimulus uncertainty or reward rate uncertainty. Calculating these joint uncertainties may well be far from trivial (Dayan & Yu, 2003).

A further promising approach to dealing with switching environments is to formally test for whether the environment has switched. One technique to make this decision would be to use a version of the sequential probability ratio test. In its log-likelihood formulation, this test is hypothesized to be the method by which the prior probabilities ρ are formed (Bogacz, 2009). Use of this test to decide whether rewards have switched would result in a cumulative sum approach in which, if sufficient evidence of switching accumulates (such as frequent mis-match of prior information with estimated parameters), the estimation process is restarted.

7.2 Relationship with Partially Observable Markov Decision Processes. State uncertainty is a key feature of the large body of work on partially observable Markov decision processes (POMDPs; see Kaelbling, Littman, & Cassandra, 1998). In contrast with the objective in this letter, many reinforcement learning approaches in POMDPs generally focus on attributing reward directly to belief states (which are analogous to our confidence levels $\boldsymbol{\rho}$) instead of learning the value of the underlying states of nature (e.g., Jaakkola, Singh, & Jordan, 1995). Cao and Guo (2004) point out that observations of received reward provide information about the underlying state, which may be useful in selecting actions, but they do not demonstrate how to implement the idea; the results in this letter partially solve their problem. The same idea is used (less explicitly) by Poupart and Vlassis (2008), although their approach is restricted to rewards drawn from a discrete distribution. We note in passing that reinforcement learning and POMDPs also interact in the frameworks of Duff (2003) and Poupart, Vlassis, Hoey, and Regan (2006), where a POMDP is used to model the learning of a Markov decision process in order to optimally balance exploration and exploitation. Thus, although POMDPs have a similar basic challenge to the model studied here, much of the research in the area addresses the problem in a very different way from this letter.

7.3 Online Learning for Action Selection. Note that, in this letter, we have considered learning the values of states of the environment. However, as observed in section 2, it is easy to generalize these results to learning the values of state-action pairs (s, a) where s is the (partially observed) state of nature and the action a is selected by the learner (see Sutton & Barto, 1998). We here clarify this claim.

Suppose a learner is given prior information $\tilde{\boldsymbol{\rho}}_t$ about states $s = 1, \ldots, S$ and has a fixed action-selection policy mapping prior state information $\tilde{\boldsymbol{\rho}}$ to a distribution over actions a. Once action a_t has been selected but before the reward has been observed, the distribution over state-action pairs is given by

$$\rho(s, a) = \tilde{\rho}(s)\mathbb{I}_{a=a_t}.$$

The analysis then goes through exactly as before so long as $\mathbb{E}[\rho(s, a)] > 0$ for all state-action pairs. This is the policy evaluation problem discussed at length by Sutton and Barto (1998).

However, the main problem of interest in this context is that of online learning, where an individual's action-selection strategy depends not only on the state information, but also on current estimates of state-action values. An important question is whether the imperfect knowledge of state can lead to poor action selection, which in turn results in poor value estimation; we believe that since the states are sampled independently on different time points and are not affected by action selection, this should not occur. Other questions to address include how to specify a sensible action-selection policy to balance exploration and exploitation when true state information is not available, and how the action selection policy will influence the convergence analysis in the appendix. This analysis of online learning introduces extra complexity that goes beyond the scope of this letter. However, note that provided that $\mathbb{E}[\rho_t(s, a) \mid \boldsymbol{\theta}_t] > 0$ for all t and other technical conditions ensuring sufficient exploration, we expect standard stochastic approximation results from the machine learning literature (in particular, Singh, Jaakkola, Littman, & Szepesvari, 2000) to apply.

7.4 Experimental Validation. Finally, we address the question of whether any of the models introduced in this letter could be employed in the brain. The confidence-weighted reinforcement learning model (see section 3.2) is unlikely to be selected by evolutionary pressure because it has poorer performance and is more complicated to implement than the simple winner-takes-all model (see section 3.1). However from an evolutionary point of view, it is more difficult to choose between the winner-takes-all and the PWRL models, because the first has the virtue of simplicity and adaptability to environmental changes, while the second can achieve more accurate value estimation.

To distinguish between the winner-takes-all and PWRL models, one could perform an experiment with human participants that was simulated in Figures 1, 2 and 3a. At the end of the experiment the participants could be asked about the average reward associated with each stimulus (Budescu, Weinberg, & Wallsten, 1988). (To increase the reliability of the verbal report of their estimate, they can be told that the payment they receive will depend on how closely their estimates match the true values (Hertwig & Ortmann, 2001).) The PWRL model predicts that their estimates will be close to the true mean rewards, while the winner-takes-all model predicts that the participants will be underestimating the better option and overestimating the poorer option, with the difference between the model predictions being greater for low confidence levels ρ.

Another experiment could also be performed in which rewards for the two stimuli are swapped in the middle of the experiment (as in the simulations of Figure 4). If any participants reported the mean reward values as they were before the switch, which of course is likely only if the confidence level ρ is small, this would indicate that human learners exhibit the deficiencies of the PWRL model, and thus would provide a support for this model. Again, the power of the experiment to differentiate between the models will be greatest if the confidence level ρ is small and will decrease as ρ increases.

Appendix: Additional Material

In this appendix we consider a general formulation of the PWRL scheme and its convergence properties. In particular, we consider a general scheme that is a modification of Titterington's (1984) algorithm to remove division by the prior confidence weights:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \operatorname{diag}(\boldsymbol{\rho}_t)\, [I_c(\boldsymbol{\theta}_t, \boldsymbol{\rho}_t)]^{-1}\, \nabla_{\boldsymbol{\theta}_t} \log p(R_t; \boldsymbol{\theta}_t, \boldsymbol{\rho}_t), \tag{A.1}$$

where $\operatorname{diag}(\boldsymbol{\rho}_t)$ is the diagonal matrix with diagonal equal to $\boldsymbol{\rho}_t$. As seen in section 5, this results in the PWRL scheme in the normal and Bernoulli examples.

A.1 Exponential Families. Consider now algorithms that estimate the (scalar) mean parameter θ of a probability distribution from the exponential family (Barndoff-Nielsen, 1978). Distributions in the exponential family have a density function (or mass function)

$$f(r; \theta) = h(r) \exp\left\{r\eta(\theta) - b(\theta)\right\},$$

where h, η, and b are functions. This family of distributions includes the Bernoulli and normal random variables previously considered, as well as exponential and Poisson random variables. Since we assume that the parameterization (i.e., choice of η and b) is such that $\mathbb{E}[R; \theta] = \theta$, the standard formula for exponential family distributions tells us that

$$\theta = \mathbb{E}[R\,;\, \theta] = \frac{b'(\theta)}{\eta'(\theta)}. \tag{A.2}$$

As in section 5, we calculate the partial derivatives of the likelihood and the complete data information matrix:

$$\begin{aligned}\frac{\partial}{\partial\theta(i)} \log p(r; \boldsymbol{\theta}, \boldsymbol{\rho}) &= \mathbb{P}(i_t = i \mid R_t = r, \boldsymbol{\theta}_t = \boldsymbol{\theta}, \boldsymbol{\rho}_t = \boldsymbol{\rho}) \\ &\qquad \times \left\{r\eta'(\theta(i)) \,-\, b'(\theta(i))\right\} \\ \frac{\partial^2}{\partial\boldsymbol{\theta}(i)^2} \log p(j, r; \boldsymbol{\theta}, \boldsymbol{\rho}) &= \mathbb{I}_{\{i=j\}} \left\{r\eta''(\theta(j)) - b''(\theta(j))\right\}\end{aligned}$$

$$\begin{aligned}
(I_c(\boldsymbol{\theta}, \boldsymbol{\rho}))_{ii} &= -\mathbb{E}\left[\frac{\partial^2}{\partial\theta(i)^2}\log p(i_t, R_t; \boldsymbol{\theta}, \boldsymbol{\rho}) \,\middle|\, \boldsymbol{\theta}_t = \boldsymbol{\theta}, \boldsymbol{\rho}_t = \boldsymbol{\rho}\right] \\
&= -\rho(i)\left\{\mathbb{E}\big[R_t \,\big|\, i_t = i, \boldsymbol{\theta}_t = \boldsymbol{\theta}, \boldsymbol{\rho}_t = \boldsymbol{\rho}\big]\eta''(\theta(i)) - b''(\theta(i))\right\} \\
&= -\rho(i)\left\{\left[\frac{b'(\theta(i))}{\eta'(\theta(i))}\right]\eta''(\theta(i)) - b''(\theta(i))\right\} \\
&= \rho(i)\eta'(\theta(i))\frac{\partial}{\partial\theta(i)}\left(\frac{b'(\theta(i))}{\eta'(\theta(i))}\right) \\
&= \rho(i)\eta'(\theta(i))\frac{\partial}{\partial\theta(i)}\theta(i) \\
&= \rho(i)\eta'(\theta(i)),
\end{aligned}$$

where we have twice used equation A.2. Hence, for all exponential family distributions parameterized by the mean, we have an update

$$\begin{aligned}
\theta_{t+1}(i) &= \theta_t(i) + \alpha\rho_t(i)\frac{1}{\rho_t(i)\eta'(\theta_t(i))}\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}_t, \boldsymbol{\rho}_t) \\
&\quad \times \left\{R_t\eta'(\theta_t(i)) - b'(\theta_t(i))\right\} \\
&= \theta_t(i) + \alpha\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}_t, \boldsymbol{\rho}_t)\left\{R_t - \theta_t(i)\right\}. \qquad \text{(A.3)}
\end{aligned}$$

A.2 Convergence. The Kullback-Liebler (KL) divergence is a natural and commonly used measure of similarity of probability distributions. Along similar lines to the results of Titterington (1984), Wang and Zhao (2006), and Cappé and Moulines (2009), we will show that the stochastic fixed points of the PWRL algorithm are minima of the expected KL divergence from the true data-generating model to the model space used to estimate the rewards. This shows that parameter estimates converge to points that make the fitted reward distributions as close as possible to the true reward distributions. Note that this result not only provides a justification for the use of the scheme when the correct model of reward distributions is used. It also provides a robustness property, showing that when an incorrect model is used, the resulting estimated parameters correspond to fitted models that are as close a fit to the truth as is possible in the selected model class.

Consider the model of section 5 where $p(r, i; \boldsymbol{\theta}, \boldsymbol{\rho}) = \rho(i)f(r; \theta(i))$, and we do not assume that $\theta(i)$ is a scalar. The complete data information matrix I_c is now block diagonal (since the $\theta(i)$ is a vector instead of a scalar) with ith block,

$$\begin{aligned}
-\mathbb{E}\left[\nabla^2_{\theta(i)}\log p(i_t, R_t; \boldsymbol{\theta}, \boldsymbol{\rho})\right] &= -\rho(i)\mathbb{E}\left[\nabla^2_{\theta(i)}f(R_t; \theta(i)) \,|\, i_t = i\right] \\
&= \rho(i)I_f(\theta(i))
\end{aligned}$$

where I_f is the positive definite information matrix corresponding to the non-mixture density f. Similarly

$$\begin{aligned}\nabla_{\theta(i)} \log p(r; \boldsymbol{\theta}, \boldsymbol{\rho}) &= \frac{\rho(i)\nabla_{\theta(i)} f(r; \theta(i))}{p(r; \boldsymbol{\theta}, \boldsymbol{\rho})} \\ &= \frac{\rho(i) f(r; \theta(i))}{p(r; \boldsymbol{\theta}, \boldsymbol{\rho})} \frac{\nabla_{\theta(i)} f(r; \theta(i))}{f(r; \theta(i))} \\ &= \mathbb{P}(i_t = i \mid R_t = r, \boldsymbol{\theta}_t = \boldsymbol{\theta}, \boldsymbol{\rho}_t = \boldsymbol{\rho}) \nabla_{\theta(i)} \log f(r; \theta(i)).\end{aligned} \tag{A.4}$$

Hence the PWRL update (equation A.1) is given by

$$\theta_{t+1}(i) = \theta_t(i) + \alpha \mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}_t, \boldsymbol{\rho}_t) \left[I_f(\theta_t(i))\right]^{-1} \nabla_{\theta_t(i)} \log f(R_t; \theta_t(i)).$$

We formalize our learning environment to allow more rigorous study. Assume that at each time instant, a probability vector $\boldsymbol{\rho}_t$ is sampled, then a state is sampled according to $\boldsymbol{\rho}_t$, and finally a reward is sampled from a distribution that depends on only the state. The sampling at time t is identical to and independent of the sampling at any other time point. We denote by π the joint distribution of $\boldsymbol{\rho}_t$ and R_t, and $\pi(\cdot \mid \boldsymbol{\rho})$ the distribution of R_t conditional on the event $\boldsymbol{\rho}_t = \boldsymbol{\rho}$. Note that the (non-switching) experimental framework described in the main body of the letter is included in this formal model.

Stochastic approximation theory tells us to consider the mean field $F(\boldsymbol{\theta})$ with ith component

$$\begin{aligned}F(\boldsymbol{\theta})(i) &= \mathbb{E}\left[\alpha^{-1}(\theta_{t+1}(i) - \theta_t(i)) \mid \boldsymbol{\theta}_t = \boldsymbol{\theta}\right] \\ &= \mathbb{E}_\pi\left[\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}, \boldsymbol{\rho}_t) \left[I_f(\boldsymbol{\theta}(i))\right]^{-1} \nabla_{\theta(i)} \log f(R_t; \theta(i))\right] \\ &= \left[I_f(\theta(i))\right]^{-1} \mathbb{E}_\pi\left[\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}, \boldsymbol{\rho}_t) \nabla_{\theta(i)} \log f(R_t; \theta(i))\right].\end{aligned} \tag{A.5}$$

A Lyapunov function for the system is a function $V(\boldsymbol{\theta})$ such that the scalar product

$$\langle F(\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta})\rangle \leq 0$$

with equality only when $\nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) = 0$. If such a function exists then when conditions are placed on the learning parameters α the convergence of $\boldsymbol{\theta}_t$ to stationary points of V can be proved (Kushner & Yin, 1997; Benaïm, 1999). We do not provide the technical details in this letter since they are closely related to the proof by Wang and Zhao (2006). Similar to the approach of Wang and Zhao, we will consider the KL divergence, conditional on $\boldsymbol{\rho}_t$, from the reward distribution under π to the fitted reward distributions, then take an expectation over $\boldsymbol{\rho}_t$. The resulting function will be shown to be a Lyapunov function.

The conditional KL divergence is defined as

$$\begin{aligned} v(\boldsymbol{\theta}, \boldsymbol{\rho}) &:= KL(p(\cdot \mid \boldsymbol{\theta}, \boldsymbol{\rho}) \parallel \pi(\cdot \mid \boldsymbol{\rho})) \\ &= \mathbb{E}_{\pi(\cdot \mid \boldsymbol{\rho})} \left[\log \left(\frac{\pi(R_t \mid \boldsymbol{\rho}_t)}{p(R_t \mid \boldsymbol{\rho}_t, \boldsymbol{\theta})} \right) \middle| \boldsymbol{\rho}_t = \boldsymbol{\rho} \right] \\ &= \mathbb{E}_{\pi(\cdot \mid \boldsymbol{\rho})} \left[\log \pi(R_t \mid \boldsymbol{\rho}_t) \mid \boldsymbol{\rho}_t = \boldsymbol{\rho} \right] \\ &\quad - \mathbb{E}_{\pi(\cdot \mid \boldsymbol{\rho})} \left[\log p(R_t \mid \boldsymbol{\rho}_t, \boldsymbol{\theta}) \mid \boldsymbol{\rho}_t = \boldsymbol{\rho} \right]. \end{aligned}$$

Taking expectation over $\boldsymbol{\rho}$ gives the expected KL divergence

$$V(\boldsymbol{\theta}) := \text{Constant} - \mathbb{E}_{\pi} \left[\log p(R_t \mid \boldsymbol{\rho}_t, \boldsymbol{\theta}) \right].$$

To show that this is a Lyapunov function for the PWRL model, take the derivative with respect to $\theta(i)$ to give

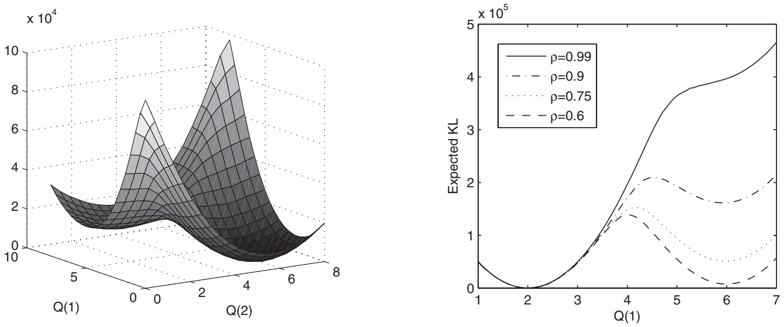
$$\begin{aligned} \nabla_{\theta(i)} V(\boldsymbol{\theta}) &= -\mathbb{E}_{\pi} \left[\nabla_{\theta(i)} \log p(R_t \mid \boldsymbol{\rho}_t, \boldsymbol{\theta}) \right] \\ &= -\mathbb{E}_{\pi} \left[\mathbb{P}(i_t = i \mid R_t, \boldsymbol{\theta}, \boldsymbol{\rho}) \nabla_{\theta(i)} \log f(R_t; \theta(i)) \right], \end{aligned}$$

as in equation A.4. Comparing with (equation A.5) we see that $\nabla_{\theta(i)} V(\boldsymbol{\theta}) = -I_f(\theta(i)) F(\boldsymbol{\theta})(i)$. Hence, taking the scalar product gives

$$\begin{aligned} \langle F(\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) \rangle &= -\sum_{i=1}^{N} \left\langle F(\boldsymbol{\theta})(i), I_f(\theta(i)) F(\boldsymbol{\theta})(i) \right\rangle \\ &\leq 0 \end{aligned}$$

since each $I_f(\theta(i))$ is positive definite. Equality holds only when each $F(\boldsymbol{\theta}(i)) = \nabla_{\boldsymbol{\theta}(i)} V(\boldsymbol{\theta}) = 0$, which is at stationary points of the expected KL divergence from the true data-generating distribution π to the fitted models.

We now consider the Lyapunov function corresponding to the experiments of sections 3 and 4, with the original formulation of PWRL (i.e., no variance estimation). Figure 7a shows the Lyapunov function for pairs of Q values when the confidence level $\rho = 0.9$. The saddle shape of this potential surface indicates that there is actually a local minimum where $Q(1) = \mu(2)$ and $Q(2) = \mu(1)$, as well as the previously calculated global minimum at $Q(1) = \mu(1)$ and $Q(2) = \mu(2)$. This local minimum corresponds to the situation where the estimates are "swapped" (see section 3.3). To investigate this phenomenon further, we plot, for different ρ values, the value of the expected KL divergence along the line $Q(2) = \mu(1) + \mu(2) - Q(1)$ (which joins



(a) Lyapunov function surface for $\rho = 0.9$.

(b) Transects of the Lyapunov surface for various ρ values.

Figure 7: Plots of the Lyapunov function for normal rewards with $\mu(1) = 2$ and $\mu(2) = 6$, with variance fixed to 1 throughout.

the two minima). In Figure 7b we see that for $\rho = 0.99$, the local minimum is not present. On the other hand, for $\rho = 0.6$, the local minimum is very pronounced, and indeed the Lyapunov function at the swapped estimates is very nearly as low as at the correct estimates.

These plots also give us further insight into the difficulties the algorithm suffered in section 4 when the reward distributions were switched partway through the experiment. If the estimates are approximately correct at the time of switching, this effectively corresponds to placing the estimates at the (incorrect) local minimum of this Lyapunov function. Hence, to learn the correct estimates, the Q values must climb out of the local minimum before converging to the global minimum corresponding to correct estimates. From Figure 7b it is clear that if ρ is small, the two potential wells are very similar, and moving the estimates from one to the other is highly unlikely. However, when ρ is large, the local minimum is in a shallow potential well, and it is easy for the estimates to escape and converge to the correct values at the global minimum.

Note that this analysis also provides clear justification for our claim that switching is the hardest kind of distributional change that can occur for this algorithm. When the switch occurs, we are essentially placing the estimates at the incorrect local minimum. Any other change in the distributions would place the estimates somewhere else on the potential surface, from which it would be easier to converge to the new global minimum.

Acknowledgments

This work was supported by EPSRC grant EP/C516303/1. We thank Guido Biele for discussion of experimental predictions of the PWRL model and

Christophe Andrieu for directing us to the literature on online EM. This research was initiated at a Bridging the Gaps event at the University of Bristol, funded by grant EP/E022685/1 from the Engineering and Physical Sciences Research Council.

References

Barndoff-Nielsen, O. (1978). *Information and exponential families in statistical theory*. New York: Wiley.

Behrens, T. E. J., Woolrich, M. W., Walton, M. E., & Rushworth, M. F. S. Learning the value of information in an uncertain world. *Nature Neuroscience*, *10*, 1214–1221.

Benaïm, M. (1999). Dynamics of stochastic approximation algorithms. In J. Azima, M. Emery, M. Ledoux, & M. Yor (Eds.), *Le Séminaire de probabilités*. New York: Springer-Verlag.

Bogacz, R. (2009). Optimal decision making theories. In J.-C. Dreher & L. Tremblay (Eds.), *Handbook of reward and decision making*. Orlando, FL: Academic Press, pp. 375–397.

Bogacz, R., & Gurney K. (2007). The basal ganglia and cortex implement optimal decision making between alternative actions. *Neural Computation*, *19*, 442–477.

Britten, K., Shadlen, M., Newsome, W., & Movshon J. A. (1992). The analysis of visual motion: A comparison of neuronal and psychophysical performance. *Journal of Neuroscience*, *12*, 4745–4765.

Budescu, D. V., Weinberg, S., & Wallsten, T. S. (1988). Decisions based on numerically and verbally expressed uncertainties. *Journal of Experimental Psychology: Human Perception and Performance*, *14*, 281–284.

Cao, X.-R., & Guo, X. (2004). Partially observable Markov decision processes with reward information. In *Proceedings of the 43rd IEEE Conference on Decision and Control* (Vol. 4, pp. 4393–4398).

Cappé, O., & Moulines, E. (2009). Online expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society*, *B 71*, 593–615.

D'Ardenne, K., McClure, S., Nystrom, L., & Cohen, J. (2008). Bold responses reflecting dopaminergic signals in the human ventral tegmental area. *Science*, *319*, 1264–1267.

Dayan, P., & Abbott, L. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Dayan, P., & Yu, A. (2003). Uncertainty and learning. *IETE Journal of Research*, *49*, 171–182.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society*, *B 39*, 1–38.

Doya, K. (2007). Reinforcement learning: Computational theory and biological mechanisms. *HFSP Journal*, *1*, 30–40.

Duff, M. (2003). Design for an optimal probe. In *Proceedings of the 20th International Conference on Machine Learning*. Menlo Park, CA: AAAI Press.

Frank, M., Seeberger, L., & O'Reilly, C. (2004). By carrot or by stick: Cognitive reinforcement learning in Parkinsonism. *Science*, *306*, 1940–1943.

Hertwig, R., & Ortmann, A. (2001). Experimental practices in economics: A methodological challenge for psychologists? *Behavioral and Brain Sciences*, *24*, 383-451.

Jaakkola, T., Singh, S. P., & Jordan, M. I. (1995). Reinforcement learning algorithm for partially observable Markov decision problems. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems, 7*. Cambridge, MA: MIT Press.

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

Kepecs, A., Uchida, N., Zariwala, H.A., & Mainen, Z.F. (2008). Neural correlates, computation and behavioural impact of decision confidence. *Nature*, *455*, 227–231.

Kiani, R., Hanks, T., & Shadlen, M. (2008). Bounded integration in parietal cortex underlies decisions even when viewing duration is dictated by the environment. *Journal of Neuroscience*, *28*, 3017–3029.

Kushner, H. J., & Yin, G. G. (1997). *Stochastic approximation algorithms and applications*. New York: Springer-Verlag.

Montague, P., Dayan, P., & Sejnowski, T. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of Neuroscience*, *16*, 1936–1947.

Poupart, P., & Vlassis, N. (2008). Model-based Bayesian reinforcement learning in partially observable domains. In *Proceedings of the Tenth International Symposium on Artificial Intelligence and Mathematics*. New York: ACM Press.

Poupart, P., Vlassis, N., Hoey, J., & Regan, K. (2006). An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*. New York: ACM Press.

Ratcliff, R. (1988). Continuous versus discrete information processing: Modeling accumulation of partial information. *Psychological Review*, *95*, 238–255.

Ratcliff, R. (2006). Modeling response signal and response time data. *Cognitive Psychology*, *53*, 195–237.

Roitman, J., & Shadlen, M. (2002). Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of Neuroscience*, *22*, 9475–9489.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, *80*, 1–27.

Shadlen, M., & Newsome, W. (1996). Motion perception: Seeing and deciding. *Proceedings of the National Academy of Sciences*, *93*, 628–633.

Shadlen, M., & Newsome, W. (2001). Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. *Journal of Neurophysiology*, *86*, 1916–1936.

Singh, S., Jaakkola, T., Littman, M. L., & Szepesvari, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, *38*, 287–308.

Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Titterington, D. (1984). Recursive parameter-estimation using incomplete data. *Journal of the Royal Statistical Society*, *B 46*, 257–267.

Tobler, P., Fiorillo, C., & Schultz, W. (2005). Adaptive coding of reward value by dopamine neurons. *Science*, *307*, 1642–1645.

Ungless, M., Magill, P., & Bolam, J. (2004). Uniform inhibition of dopamine neurons in the ventral tegmental area by aversive stimuli. *Science*, *303*, 2040–2042.

Usher, M., & McClelland, J. (2001). On the time course of perceptual choice: The leaky competing accumulator model. *Psychological Review*, *108*, 550–592.

Wang, S., & Zhao, Y. (2006). Almost sure convergence of Titterington's recursive estimator for mixture models. *Statistics and Probability Letters*, *76*, 2001–2006.

Yu, A., & Cohen, J. D. (2009). Sequential effects: Superstition or rational behavior. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*, *21*. Cambridge, MA: MIT Press.

Yu, A., & Dayan, P. (2003). Expected and unexpected uncertainty: ACh and NE in the neocortex. In S. Becker, S. Thrïn, & K. Obermayer (Eds.), *Advances in neural information processing Systems*, *15*. Cambridge, MA: MIT Press.

Received January 20, 2009; accepted September 16, 2009.