Estimation and Approximation Bounds for Gradient-Based Reinforcement Learning

Peter L. Bartlett and **Jonathan Baxter**
Research School of Information Sciences and Engineering
Australian National University
Canberra ACT 0200, AUSTRALIA
Peter.Bartlett@anu.edu.au, Jonathan.Baxter@anu.edu.au

Abstract

We model reinforcement learning as the problem of learning to control a Partially Observable Markov Decision Process (POMDP), and focus on gradient ascent approaches to this problem. In [3] we introduced GPOMDP, an algorithm for estimating the performance gradient of a POMDP from a single sample path, and we proved that this algorithm almost surely converges to an approximation to the gradient. In this paper, we provide a convergence *rate* for the estimates produced by GPOMDP, and give an improved bound on the approximation error of these estimates. Both of these bounds are in terms of mixing times of the POMDP.

1 INTRODUCTION

Many control, scheduling, planning and game-playing tasks can be formulated as reinforcement learning problems, in which an agent chooses actions to take in some environment, aiming to maximize a reward function. We can model the environment as a *partially observable Markov decision process* (POMDP) and formulate these reinforcement learning problems as the problem of controlling the POMDP.

Figure 1 illustrates a POMDP, controlled by a policy μ. We assume that there is a finite state space $\mathcal{S} = \{1, \ldots, N\}$, representing the distinct states that the environment can take, a finite control set $\mathcal{U}$, representing all actions that the agent can choose at each time step, and a finite observation set $\mathcal{Y}$, representing all observations that might be presented to the agent.

The evolution of the states depends on the actions. Each $u \in \mathcal{U}$ determines the state transition probability $p_{ij}(u)$, that is, the probability of transition from state i to state j, given control action u. Thus, the matrix

$$P(u) = [p_{ij}(u)]$$

is a stochastic matrix; $\sum_j p_{ij} = 1$ for $i \in \{1, \ldots, N\}$.

For each state $i \in \mathcal{S}$, an observation $y \in \mathcal{Y}$ is generated independently according to a probability distribution $\nu(i)$ over observations in $\mathcal{Y}$. We denote the probability of observation y by $\nu_y(i)$. In the special case $\nu_y(i) = \delta(i)$, the observation y is the same as the state, and the POMDP is completely observable.

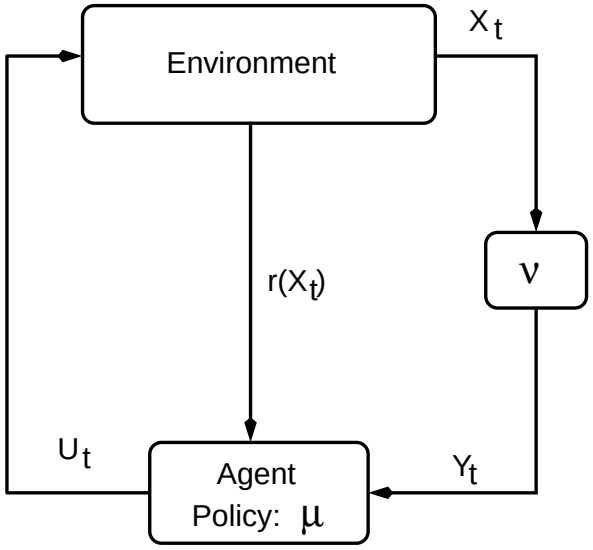

Figure 1: A partially observable Markov decision process (POMDP) controlled by the policy μ. The actions U_t determine the probabilities of transitions between different states X_t. The MDP is *partially observable* because the state X_t is not observed; the observation Y_t is conditionally independent, given X_t. The stochastic policy μ maps from observations Y_t to distributions over actions U_t. Associated with the state X_t is a reward value, $r(X_t)$. The aim is to choose a policy to maximize the long term average of the reward.

The relationship between the observations seen by the agent and the actions it chooses is defined by the policy μ. We consider randomized policies, and we assume that the policy is defined by a vector of parameters. Formally, a *parameterized randomized policy* is a function μ mapping parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ and observations $y \in \mathcal{Y}$ into probability distributions over the controls $\mathcal{U}$. That is, for each observation y and parameter vector θ, $\mu(\theta, y)$ is a distribution over the controls in $\mathcal{U}$. We denote the probability of control u under this distribution by $\mu_u(\theta, y)$.

Each state i has an associated reward $r(i)$. The aim is to choose the parameters θ of the policy so as to maximize the

long-term average reward,

$$\eta = \lim_{T \to \infty} \frac{1}{T} \mathbf{E} \sum_{t=0}^{T-1} R_t, \tag{1}$$

where $R_t = r(X_t)$ is the reward associated with the state X_t at time t. For simplicity of exposition, we will focus on policies that depend only upon the current observation Y_t. However, the results of this paper can easily be extended to policies that depend on finite histories of observations $(Y_t, Y_{t-1}, \ldots, Y_{t-k})$.

For each parameter vector θ, we have a fixed stochastic policy, so the underlying state of the POMDP evolves as a Markov chain with transition probability matrix

$$P(\theta) = [p_{ij}(\theta)]_{i,j=1\ldots n},$$

where

$$p_{ij}(\theta) = \mathbf{E}_{Y \sim \nu(i)} \mathbf{E}_{U \sim \mu(\theta, Y)} p_{ij}(U).$$

We write the parameterized class of stochastic matrices as $\mathcal{P} := \{P(\theta) : \theta \in \Theta\}$. Denote the Markov chain corresponding to $P(\theta)$ by $M(\theta)$. We will use $\{X_t, Y_t, U_t, R_t\}$ to denote the joint stochastic process where the states X_t are generated according to $P(\theta)$, observations Y_t are generated according to $\nu(X_t)$, controls U_t are generated according to $\mu(\theta, Y_t)$ and rewards R_t are generated according to $r(X_t)$.

We can view the average reward (1) as a function $\eta(\theta)$ of $\theta \in \mathbb{R}^d$, where θ are the parameters of the policy. Provided the dependence of η on θ is differentiable, we can compute $\nabla\eta(\theta)$ and use a gradient ascent method in order to increase the average reward.

This approach was pioneered by Williams [11], who introduced the REINFORCE algorithm for estimating the gradient in *episodic* tasks, for which there is an identified recurrent state i^*, and the agent is told when this state is entered. REINFORCE returns a gradient estimate each time i^* is entered. Williams showed that the expected value of this estimate is the gradient direction, in the case that the number of steps between visits to i^* is a constant. It is easy to prove the stronger result that the expected value of the estimate is the gradient, even when the number of steps is a random variable (see Section 3).

Other researchers have investigated algorithms that estimate the gradient of the expected reward [6, 4, 9, 8, 2, 10, 7]. With the exception of [6], these algorithms are all restricted to episodic tasks, or for tasks where the long term average reward is accurately known. The weakness of approaches that are restricted to episodic tasks arises from the reliance on the identifiable recurrent state i^*. Although the assumptions we make in this paper about the POMDP ensure that every state is recurrent, as the size of the state space increases, we can expect that the expected time between visits will increase. Furthermore, the time between visits depends on the parameters, and states that are frequently visited for the initial value of the parameters may become very rare as performance improves. In addition, in an arbitrary POMDP it may be difficult to estimate the underlying states, and therefore to determine when the gradient estimate should be updated.

In [3], we extended Williams' algorithm to avoid the need for an identifiable, frequently visited recurrent state. We introduced GPOMDP, an algorithm for estimating an approximation to the gradient (this algorithm is described in detail in Section 4). The estimates produced by REINFORCE involve products of the average reward over a sample path between visits to a recurrent state and the sum of certain gradient contributions over that sample path. In contrast, GPOMDP uses products of the instantaneous reward at each state, and a sum over the past of exponentially discounted gradient contributions. The discount factor, β, is a parameter of the algorithm. The role of this parameter depends on the *mixing time* of the POMDP. (The mixing time is the time constant in the exponential convergence of a stochastic process to its stationary distribution—see Section 2 for the definition.) We showed in [3] that, under certain assumptions on the POMDP, the estimates produced by GPOMDP converge almost surely to $\nabla_\beta \eta$, an approximation to the gradient that depends on the discount factor β used by the algorithm. The *approximation error* of the algorithm is the size of the difference between the true gradient $\nabla\eta$ and the estimate $\nabla_\beta\eta$ to which the algorithm converges. In [3], we showed that this approximation error is small provided that the time constant $\tau_{\text{alg}} = 1/(1-\beta)$ is large compared with the mixing time of the derived Markov chain $M(\theta)$ (under the assumption that the eigenvalues of the transition probability matrix are all distinct).

In this paper, we give bounds on the *estimation error* of the GPOMDP algorithm. The estimation error, which is the size of the difference between the output of the algorithm and its asymptotic output, arises because the algorithm sees only a finite data sequence. Our estimation error bounds are in terms of the algorithm's time constant $\tau_{\text{alg}} = 1/(1-\beta)$ and the mixing time of a certain stochastic process associated with the POMDP. In particular, if this mixing time is τ, the estimation error is of the order

$$\sqrt{\frac{{\tau_{\text{alg}}}^2 \tau}{n}},$$

ignoring log factors, where n is the running time of the algorithm. We also give an approximation error bound in terms of a certain mixing time τ^* of $M(\theta)$, without the restrictive assumption of [3] that the eigenvalues are distinct. We show that the approximation error of the algorithm's estimate is of the order

$$\sigma_R \frac{\tau^*}{\tau_{\text{alg}}},$$

where σ_R^2 is the variance of the reward R_t under the stationary distribution. These results show that mixing times of the controlled POMDP provide estimates for both the approximation error and the estimation error, and suggest that mixing time is crucial to the performance of the algorithm. The results also formalize a natural tradeoff: as the time constant of the algorithm gets large (when the parameter β approaches one), the approximation error decreases but the estimation error increases. This provides insight into the appropriate choice of the algorithm's parameter β.

In Section 2, we describe the assumptions we make about the controlled POMDP, and present some definitions and preliminary results. Section 3 reviews the REINFORCE algorithm, and shows that the expected value of its estimates

is correct. Section 4 presents GPOMDP, and reviews the results from [3]. Sections 5 and 6 give bounds on the convergence rate and approximation error.

2 ASSUMPTIONS, DEFINITIONS AND PRELIMINARY RESULTS

We assume that the Markov chains $M(\theta)$ satisfy several assumptions.

Assumption 1. *For each $\theta \in \Theta$, the Markov chain $M(\theta)$ is ergodic.*

A stationary distribution of a Markov chain with transition probability matrix P is a probability distribution $\pi = [\pi(1), \ldots, \pi(N)]'$ over states that satisfies

$$\pi' P = \pi'.$$

Assumption 1 implies that each $P(\theta)$ has a unique positive stationary distribution

$$\pi(\theta) := [\pi(\theta, 1), \ldots, \pi(\theta, N)]',$$

and that the sequence of states exhibits exponential convergence to this stationary distribution. We could also allow aperiodic Markov chains which have a single recurrent class, plus some transient states.

If a gradient method is to be applicable, suitable derivatives must exist. The following assumption about the parameterization of the stochastic policies suffices.

Assumption 2. *The derivatives, $\partial \mu_u(\theta, y)/\partial \theta_k$ exist for all $u \in \mathcal{U}$, $y \in Y$, $k = 1 \ldots d$ and $\theta \in \Theta$.*

This assumption implies that the derivatives $\partial p_{ij}(\theta)/\partial \theta_k$ exist for all $\theta \in \Theta$, $i, j = 1, \ldots, N$ and $k = 1, \ldots, d$.

Assumption 3. *There is a $C < \infty$ such that, for all states i, the magnitude of the reward satisfies $|r(i)| \leq C$.*

Assumption 4. *There is a $B < \infty$ such that, for all controls $u \in \mathcal{U}$, parameter vectors $\theta \in \Theta$, observations $y \in \mathcal{Y}$, and $k \in \{1, \ldots, d\}$,*

$$\frac{|\partial \mu_u(\theta, y)/\partial \theta_k|}{\mu_u(\theta, y)} \leq B.$$

The assumption that the magnitudes of the rewards are uniformly bounded is quite natural: the agent's actions can have only limited consequences. The ratios between derivatives and action probabilities are features of the class of policies that can be bounded by design.

To measure the progress of the state distribution toward the stationary distribution π, we use the *total variation distance*.

Definition 5. *The total variation distance between two probability distributions P, Q on a set $\mathcal{X}$ is*

$$\mathrm{d_{TV}}(P, Q) = |P - Q|(\mathcal{X}),$$

where the finite measure $|P - Q|$ is the absolute difference between the measures P and Q. (If P and Q are discrete, $|P - Q|(\mathcal{X}) = \sum_{x \in \mathcal{X}} |P(x) - Q(x)|$. If they are continuous, $|P - Q|(X) = \int_{\mathcal{X}} |p(x) - q(x)|\, dx$.)

The following lemma is folklore. (It follows, for example, from the Jordan decomposition theorem—see [5].)

Lemma 6. *For distributions P, Q on $\mathcal{X}$,*

$$\mathrm{d_{TV}}(P, Q) = 2 \sup_S (P(S) - Q(S)),$$

where the supremum is over all measurable subsets $S \subseteq \mathcal{X}$.

For a stochastic process $\{X_t\}$ and $j \leq k$, we use X_j^k to denote $(X_j, X_{j+1}, \ldots, X_k)$, and $X_{-\infty}^j$ to denote the infinite sequence $(\ldots, X_{j-1}, X_j)$.

Definition 7. *A causal stochastic process $\{X_t\}$ taking values in $\mathcal{X}$ is* mixing *if, for all sequence lengths k, there is a stationary distribution π on $\mathcal{X}^k$ such that almost surely the distribution of X_t^{t+k-1} conditioned on $X_{-\infty}^0$ converges to π as $t \to \infty$.*

Definition 8. *We say that a stochastic process $\{X_t\}$ is* exponentially mixing with time constant τ *(τ-mixing for short) if it is mixing and, for all t_0, $t \geq 0$ and $X_{-\infty}^{t_0}$, the distribution p^t of X_{t_0+t} conditioned on $X_{-\infty}^{t_0}$ satisfies*

$$\mathrm{d_{TV}}(p^t, \pi) \leq \exp\left(-\lfloor t/\tau \rfloor\right),$$

where π is the stationary distribution of X_t.

When we talk of the mixing time of a Markov chain, we mean the smallest τ such that the state sequence is τ-mixing.

Lemma 9. *If $\{X_i\}$ is τ-mixing, then for any predicate ϕ on $\mathcal{X}^n$,*

$$\begin{aligned}
&\Pr\left(\phi\left(X_{n_1}, X_{n_1+t}, \ldots, X_{n_1+nt}\right) \left| X_{-\infty}^0 \right.\right) \\
&\leq \frac{1}{2} e^{-\lfloor n_1/\tau \rfloor} + \frac{n-1}{2} e^{-\lfloor t/\tau \rfloor} \\
&\qquad + \pi^n\left\{(X_{n_1}, \ldots, X_{n_1+nt}) : \phi\left(X_{n_1}, \ldots, X_{n_1+nt}\right)\right\},
\end{aligned}$$

where π^n is the product *distribution on $\mathcal{X}^n$ generated by the stationary distribution π on $\mathcal{X}$.*

Proof. Consider distributions P_1, Q_1 on a set $\mathcal{X}_1$ and P_2, Q_2 on a set $\mathcal{X}_2$.

$$\begin{aligned}
&\mathrm{d_{TV}}(P_1 \times P_2, Q_1 \times Q_2) \\
&= \int_{\mathcal{X}_1 \times \mathcal{X}_2} d|P_1 \times P_2 - Q_1 \times Q_2| \\
&= \int_{\mathcal{X}_1 \times \mathcal{X}_2} d\left|\, (P_1 \times P_2 - Q_1 \times P_2) \right. \\
&\qquad\qquad \left. - (Q_1 \times Q_2 - Q_1 \times P_2)\right| \\
&\leq \int_{\mathcal{X}_1 \times \mathcal{X}_2} d\left(|P_1 - Q_1| \times P_2 + Q_1 \times |P_2 - Q_2|\right) \\
&= \int_{\mathcal{X}_1} d|P_1 - Q_1| + \int_{\mathcal{X}_2} d|P_2 - Q_2| \\
&= \mathrm{d_{TV}}(P_1, Q_1) + \mathrm{d_{TV}}(P_2, Q_2).
\end{aligned}$$

Lemma 6 implies that, for any $[0, 1]$-valued function f,

$$\left| \int f(x) dP(x) - \int f(x) dQ(x) \right| \leq \frac{\mathrm{d_{TV}}(P, Q)}{2}.$$

An easy inductive argument implies the result. □

Algorithm 1 The REINFORCE algorithm.

Given:

- Parameterized class of randomized policies $\{\mu(\theta,\cdot)\}$ satisfying Assumptions 2 and 4.
- POMDP which, when controlled by the randomized policies $\mu(\theta,\cdot)$, corresponds to a parameterized class of Markov chains satisfying Assumption 1.
- Start state $X_0 = i^*$.
- Observation sequence $Y_0, Y_1, \ldots$ and reward sequence $R_0, R_1, \ldots$ generated by the POMDP with controls $U_0, U_1, \ldots$ generated randomly according to $\mu(\theta, Y_t)$, with rewards R_t satisfying Assumptions 3.

Set $j = 0$, $z_0 = 0$, $t_0 = 0$, and $\Delta_0 = 0$ ($z_0, \Delta_0 \in \mathbb{R}^d$).
for each observation Y_t, control U_t **do**
  if $X_t = i^*$ **then**
    $t_{j+1} = t$
    $\Delta_{j+1} = \Delta_j + \frac{1}{j+1}\left[\frac{1}{t_{j+1}-t_j}\sum_{s=t_j+1}^{t_{j+1}} R_s z_t - \Delta_j\right]$
    $j = j + 1$
    $z_{t+1} = 0$
  else
    $z_{t+1} = z_t + \frac{\nabla\mu_{U_t}(\theta,Y_t)}{\mu_{U_t}(\theta,Y_t)}$
  end if
end for

We shall make use of Hoeffding's inequality:

Theorem 10 (Hoeffding's Inequality). *If the random variables $X_1, \ldots, X_n$ are independent and satisfy $X_i \in [a_i, b_i]$, we have*

$$\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i)\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2\epsilon^2 n}{\frac{1}{n}\sum_{i=1}^{n}(b_i-a_i)^2}\right).$$

3 WILLIAMS' REINFORCE ALGORITHM

The gradient ascent approach to reinforcement learning was pioneered by Williams [11], who introduced REINFORCE (Algorithm 1). Williams showed that the expected value of the estimates Δ_j returned by this algorithm is the gradient direction, in the case that the number of steps between visits to the identified recurrent state i^* is a constant. It is easy to prove the following stronger result.

Theorem 11. *Under Assumptions 1, 2, 3, and 4, for each j,*

$$\mathbf{E}\Delta_j = \nabla\mathbf{E}\left(\frac{1}{T}\sum_{t=1}^{T} R_t \,\middle|\, X_0 = i^*\right),$$

where T is the time of the first return to state i^*.

Proof. It is easy to see that the expression for Δ_{j+1} is a recursive computation of the average of the $j+1$ random variables $\xi_1, \ldots, \xi_{j+1}$, where

$$\xi_{j+1} = \frac{1}{t_{j+1}-t_j}\sum_{s=t_j+1}^{t_{j+1}} R_s z_t,$$

so we need only compute the expectation of $\xi_1 = \Delta_1$. (In fact, because of the Markov property, the random variables ξ_j are i.i.d.) Now,

$$\Delta_1 = \left(\frac{1}{T}\sum_{s=1}^{T} R_s\right)\sum_{t=0}^{T-1}\frac{\nabla\mu_{U_t}(Y_t)}{\mu_{U_t}(Y_t)},$$

where T is the time of the first return to state i^*. Define

$$\bar{R} = \frac{1}{T}\sum_{s=0}^{T} R_s.$$

We shall show by induction that

$$\begin{aligned}\mathbf{E}\Delta_1 &- \nabla\mathbf{E}\left(\bar{R}\middle| X_0 = i^*\right)\\ &= \mathbf{E}\left(\mathbf{E}\left(\bar{R}\sum_{t=s}^{T-1}\frac{\nabla\mu_{U_t}(Y_t)}{\mu_{U_t}(Y_t)}\middle| S_0^s\right)\right.\\ &\qquad\left. - \nabla\mathbf{E}\left(\bar{R}\middle| S_0^s\right)\middle| T > s\right)\Pr(T > s), \qquad (2)\end{aligned}$$

where

$$S_0^s = (X_0, Y_0, U_0, \ldots, X_{s-1}, Y_{s-1}, U_{s-1}, X_s).$$

Clearly, (2) is true for $s = 0$. Suppose it is true for $s \geq 0$. Fix any suitable S_0^s (which must have positive probability and contain no i^*s.) Then we can write

$$\begin{aligned}&\mathbf{E}\left(\bar{R}\sum_{t=s}^{T-1}\frac{\nabla\mu_t}{\mu_t}\middle| S_0^s\right)\\ &= \sum_{Y_s}\nu_s\sum_{U_s}\mu_s\left(p_{X_s,i^*}(U_s)\right.\\ &\qquad\left.\times\mathbf{E}\left(\frac{1}{s+1}\sum_{t=1}^{s+1}R_t\frac{\nabla\mu_s}{\mu_s}\middle| S_0^{s+1}\right)\right.\\ &\quad\left.+\sum_{X_{s+1}\neq i^*}p_{X_s,X_{s+1}}(U_s)\mathbf{E}\left(\bar{R}\sum_{t=s+1}^{T-1}\frac{\nabla\mu_t}{\mu_t}\middle| S_0^{s+1}\right)\right),\end{aligned}$$

where we have used the abbreviated notation $\mu_t = \mu_{U_t}(Y_t)$, $\nu_s = \nu_{Y_s}(X_s)$, and we have relied on Assumption 2. Taking the $\nabla\mu_s/\mu_s$ outside the expectations in both terms, and rearranging shows that this is equal to

$$\begin{aligned}&\mathbf{E}\left(\frac{\nabla\mu_s}{\mu_s}\mathbf{E}\left(\bar{R}\middle| S_0^{s+1}\right)\middle| S_0^s\right)\\ &\quad+\mathbf{E}\left(\mathbf{E}\left(\bar{R}\sum_{t=s+1}^{T-1}\frac{\nabla\mu_t}{\mu_t}\middle| S_0^{s+1}\right)\middle| T > s+1\right)\\ &\qquad\times\Pr\left(T > s+1 | S_0^s\right).\end{aligned}$$

Using a similar expansion, we have

$$\begin{aligned}
&\nabla \mathbf{E}\left(\bar{R}|S_0^s\right) \\
=&\sum_{Y_s}\nu_s\sum_{U_s}\nabla\mu_s\left(p_{X_s,i^*}(U_s)\mathbf{E}\left(\frac{1}{s+1}\sum_{t=1}^{s+1}R_t\middle|S_0^{s+1}\right)\right. \\
&\left.+\sum_{X_{s+1}\neq i^*}p_{X_s,X_{s+1}}(U_s)\mathbf{E}\left(\bar{R}|S_0^{s+1}\right)\right)+ \\
&\sum_{Y_s}\nu_s\sum_{U_s}\mu_s\left(\sum_{X_{s+1}\neq i^*}p_{X_s,X_{s+1}}(U_s)\nabla\mathbf{E}\left(\bar{R}|S_0^{s+1}\right)\right) \\
=&\mathbf{E}\left(\frac{\nabla\mu_s}{\mu_s}\mathbf{E}\left(\bar{R}\middle|S_0^{s+1}\right)\middle|S_0^s\right) \\
&+\mathbf{E}\left(\nabla\mathbf{E}\left(\bar{R}|S_0^{s+1}\right)\middle|T>s+1\right)\Pr\left(T>s+1|S_0^s\right).
\end{aligned}$$

Subtracting these equations and taking the expectation over S_0^s shows that (2) is true for $s+1$. By induction, it is true for all $s \geq 0$.

It remains to show that the quantity on the right hand side of (2) goes to zero as s gets large. Using Assumptions 1, 3 and 4, it is easy to verify that $\|\nabla\mathbf{E}\bar{R}\| < c$ for some constant c that depends only on θ. It follows that

$$\|\mathbf{E}\Delta_j - \nabla\mathbf{E}\bar{R}\| \leq (BR\mathbf{E}(T-s|T>s)+c)\Pr(T>s),$$

which (under Assumption 1), is no more than a constant times $\Pr(T > s)$. Since this probability approaches zero as s gets large, the result is proved. □

Notice that the proof did not rely on the fact that R_t is a function of the state X_t. Indeed, the same proof gives a similar result when $(1/T)\sum_{t=1}^{T}R_t$ is replaced by a bounded random variable $\bar{R}$ that depends only on the sequence of states $X_{t'}$ and actions $U_{t'}$ between visits to the state i^*.

4 THE GPOMDP ALGORITHM

In [3], we extended Williams' algorithm to avoid the need for an identifiable, frequently visited recurrent state. Algorithm 2 shows GPOMDP, an algorithm for estimating an approximation to the gradient. In fact, Algorithm 2 is a slightly modified version of the algorithm presented in [3]. This algorithm has three distinct phases, which extend for n_1, n_2, n_3 time steps. The first phase involves waiting for the controlled POMDP to mix. The second involves gathering gradient information about actions that are taken. The third involves waiting for the long term outcomes of the actions for which the gradient information was gathered. (The algorithm in [3] did not include the first and third phase.) Introducing the first and third phases simplifies the analysis, but it is easy to extend the results to the algorithm presented in [3].

It is easy to see that the algorithm returns

$$\Delta_{n_1+n_2+n_3} = \frac{1}{n_2}\sum_{t=n_1+1}^{n_1+n_2+n_3} z_t R_t.$$

Call this value Δ. The convergence result in [3] implies that, under Assumptions 1, 2, 3 and 4, starting from any initial state, for any n_1, n_3, the limit as $n_2 \to \infty$ of the estimate Δ produced by this algorithm is almost surely

$$\nabla_\beta\eta = \pi'\nabla P J_\beta,$$

where $J_\beta = [J_\beta(1), \ldots, J_\beta(n)]$ is the vector of expected discounted future rewards,

$$J_\beta(i) = \mathbf{E}\left[\sum_{t=0}^{\infty}\beta^t R_t|X_0 = i\right]$$

The vector $\nabla_\beta\eta$ is an approximation to the gradient that depends on the parameter β of the algorithm. In the next section, we prove a (non-asymptotic) bound on the estimation error $\|\Delta - \nabla_\beta\eta\|_\infty$ of the GPOMDP algorithm, as a function of n_1, n_2, n_3.

Algorithm 2 The GPOMDP algorithm.

Given:
- Parameterized class of randomized policies $\{\mu(\theta,\cdot)\}$ satisfying Assumptions 2 and 4.
- POMDP which, when controlled by the randomized policies $\mu(\theta,\cdot)$, corresponds to a parameterized class of Markov chains satisfying Assumption 1.
- $\beta \in [0,1)$.
- Arbitrary (unknown) starting state i_0.
- Observation sequence $Y_0, Y_1, \ldots$ and reward sequence $R_0, R_1, \ldots$ generated by the POMDP with controls $U_0, U_1, \ldots$ generated randomly according to $\mu(\theta, Y_t)$, with the rewards R_t satisfying Assumption 3.

Set $z_0 = 0$ and $\Delta_0 = 0$ ($z_0, \Delta_0 \in \mathbb{R}^d$).
for $t = 0, \ldots, n_1 - 1$ **do**
  $z_{t+1} = z_t$.
  $\Delta_{t+1} = \Delta_t$.
end for
for $t = n_1, \ldots, n_1 + n_2 - 1$ **do**
  $z_{t+1} = \beta z_t + \dfrac{\nabla\mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)}$
  $\Delta_{t+1} = \Delta_t + \dfrac{1}{t - n_1 + 1}\left[R_{t+1}z_{t+1} - \Delta_t\right]$
end for
for $t = n_1 + n_2, \ldots, n_1 + n_2 + n_3 - 1$ **do**
  $z_{t+1} = \beta z_t$.
  $\Delta_{t+1} = \Delta_t + R_{t+1}z_{t+1}$.
end for

5 CONVERGENCE RATE

We can rewrite Δ, the estimate produced by the GPOMDP algorithm, progressively expanding terms involving $z_{n_1+n_2}$, then $z_{n_1+n_2-1}$, and so on up to z_{n_1+1}, and separating terms involving distinct gradients ∇_{n_1+t}. This gives

$$\Delta = \frac{1}{n_2}\sum_{t=0}^{n_2-1}\nabla_{n_1+t}\left(\sum_{s=0}^{n_2+n_3-1-t}\beta^s R_{n_1+t+1+s}\right), \quad (3)$$

where

$$\nabla_t = \frac{\nabla\mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)}.$$

This illustrates how the algorithm works: its estimate is a weighted sum of the gradients $\nabla \mu_{U_t}(\theta, Y_t)$, which are the directions in parameter space that lead to a maximal increase in the probability of the actions U_t that were chosen at each time t. These directions are weighted by an estimate of the *value* of that action (a discounted sum into the future of the rewards that followed the action U_t). They are also weighted by $1/\mu(U_t)$, which ensures that very likely or unlikely actions are represented fairly in the average.

Each term in the sum (3) depends on the complete sequence of future rewards, R_t. However, the dependence decreases exponentially quickly, so the terms can be accurately approximated by considering a finite window into the future. To this end, we introduce a modified algorithm (the *k-blocked algorithm*), which uses only k of the future reward values. This algorithm returns

$$\Delta^k = \frac{1}{n_2} \sum_{t=n_1}^{n_1+n_2-1} \nabla_t \sum_{s=0}^{k-1} \beta^s R_{t+s+1}.$$

We assume that $k \leq n_3 + 1$.

Notice that the estimate Δ^k of the k-blocked algorithm is an average of n_2 terms, each of which is a function of a vector

$$S_t^k = (\nabla_t, R_{t+1}, R_{t+2}, \ldots, R_{t+k}).$$

Define

$$\Delta_t^k = \nabla_t \sum_{s=0}^{k-1} \beta^s R_{t+s+1},$$

so that

$$\Delta^k = \frac{1}{n_2} \sum_{t=n_1}^{n_1+n_2-1} \Delta_t^k.$$

Because of Assumptions 3 and 4, we have the bound

$$\|\Delta_t^k\|_\infty \leq \frac{BC}{1-\beta}.$$

Lemma 12. *Under Assumptions 1, 2, 3 and 4, the estimate Δ returned by the* GPOMDP *algorithm and the estimate Δ^k returned by the k-blocked algorithm satisfy*

$$\left\|\Delta^k - \Delta\right\|_\infty \leq \frac{BC}{1-\beta}\beta^k.$$

Proof. Using Equation (3), we have

$$\begin{aligned}
&\left\|\Delta^k - \Delta\right\| \\
&= \frac{1}{n_2} \left\| \sum_{t=n_1}^{n_1+n_2-1} \nabla_t \left(\sum_{s=0}^{k-1} \beta^s R_{t+s+1} \right. \right. \\
&\qquad \left. \left. - \sum_{s=0}^{n_1+n_2+n_3-(t+1)} \beta^s R_{t+s+1} \right) \right\| \\
&\leq \frac{1}{n_2} \sum_{t=n_1}^{n_1+n_2-1} \|\nabla_t\| \left| \sum_{s=0}^{k-1} \beta^s R_{t+s+1} \right. \\
&\qquad \left. - \sum_{s=0}^{n_1+n_2+n_3-(t+1)} \beta^s R_{t+s+1} \right| \\
&\leq \frac{1}{n_2} \sum_{t=n_1}^{n_1+n_2-1} \|\nabla_t\| \sum_{s=k}^{n_1+n_2+n_3-(t+1)} \beta^s |R_{t+s-1}| \\
&\qquad \text{(for } k \leq n_3 + 1\text{)} \\
&\leq \sup_t \|\nabla_t\| \sup_t |R_t| \sum_{s=k}^{n_1+n_2+n_3-(t+1)} \beta^s,
\end{aligned}$$

which implies the result. □

A similar proof, plus the ergodic theorem and the asymptotic convergence result in [3], give the following result.

Lemma 13.

$$\left\|\mathbf{E}_\pi \Delta_t^k - \nabla_\beta \eta\right\| \leq \frac{BC}{1-\beta}\beta^k.$$

We can now obtain the main result of this section. Recall that d is the number of policy parameters.

Theorem 14. *If the process*

$$S_t^k = (\nabla_t, R_{t+1}, R_{t+2}, \ldots, R_{t+k})$$

is τ-mixing, $s \leq n_2$, and $k \leq n_3 + 1$, then

$$\begin{aligned}
&\Pr\left(\|\Delta - \nabla_\beta \eta\|_\infty \geq \epsilon + \frac{2BC}{1-\beta}\beta^k \,\middle|\, X_{-\infty}^0 \right) \\
&\leq \frac{sd}{2} e^{-\lfloor n_1/\tau \rfloor} + \frac{n_2 d}{2} e^{-\lfloor s/\tau \rfloor} \\
&\qquad + 2sd \exp\left(\frac{-\epsilon^2 n_2 (1-\beta)^2}{4B^2C^2 s} \right).
\end{aligned}$$

The theorem is an easy consequence of the following theorem, applied to the function Δ_t^k of the vector S_t^k, together with Lemmas 12 and 13.

Theorem 15. *If $\{X_t\}$ is τ-mixing and $f : \mathcal{X} \to [a,b]^d$, and $s \leq n_2$, then*

$$\begin{aligned}
&\Pr\left(\left\| \frac{1}{n_2} \sum_{i=n_1}^{n_1+n_2-1} f(X_i) - E_\pi f \right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0 \right) \\
&\leq \frac{d}{2} \left(s e^{-\lfloor n_1/\tau \rfloor} + n_2 e^{-\lfloor s/\tau \rfloor} + 4s \exp\left(\frac{-\epsilon^2 n_2}{4(b-a)^2 s} \right) \right).
\end{aligned}$$

Proof. Combining Hoeffding's inequality (Theorem 10) and Lemma 9 shows that, for any τ-mixing stochastic process $\{X_i\}$ and any $f : \mathcal{X} \to [a, b]$,

$$\Pr\left(\left|\frac{1}{n}\sum_{i=0}^{n-1} f(X_{n_1+it}) - E_\pi f\right| \geq \epsilon \,\middle|\, X_{-\infty}^0\right)$$
$$\leq \frac{1}{2}e^{-\lfloor n_1/\tau\rfloor} + \frac{n-1}{2}e^{-\lfloor t/\tau\rfloor} + 2\exp\left(\frac{-2\epsilon^2 n}{(b-a)^2}\right). \quad (4)$$

The idea of the rest of the proof is to split the sequence from n_1 to $n_1 + n_2 - 1$ into m interleaved subsequences, so that each consecutive element of each of these subsequences is separated by s time steps. Rapid mixing ensures that these subsequences are approximately i.i.d. Suppose at first that $n_2 = ms$ for some positive integer m. Then

$$\Pr\left(\left\|\frac{1}{n_2}\sum_{i=n_1}^{n_1+n_2-1} f(X_i) - E_\pi f\right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right)$$
$$\leq \Pr\left(\exists n_1 \leq j \leq n_1 + s - 1 : \left\|\frac{1}{m}\sum_{i=0}^{m-1} f(X_{is+j}) - E_\pi f\right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right)$$
$$\leq s \max_j \Pr\left(\left\|\frac{1}{m}\sum_{i=0}^{m-1} f(X_{is+j}) - E_\pi f\right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right), \quad (5)$$

where the max is over $n_1 \leq j \leq n_1 + s - 1$. Now, the union bound and Inequality 4 imply that

$$\Pr\left(\left\|\frac{1}{m}\sum_{i=0}^{m-1} f(X_{is+j}) - E_\pi f\right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right)$$
$$\leq d\left(\frac{1}{2}e^{-\lfloor j/\tau\rfloor} + \frac{m-1}{2}e^{-\lfloor s/\tau\rfloor} + 2\exp\left(\frac{-2\epsilon^2 m}{(b-a)^2}\right)\right).$$

Thus, the right hand side of (5) is no more than

$$sd\left(\frac{1}{2}e^{-\lfloor n_1/\tau\rfloor} + \frac{m-1}{2}e^{-\lfloor s/\tau\rfloor} + 2\exp\left(\frac{-2\epsilon^2 m}{(b-a)^2}\right)\right).$$

Now, for any positive integer s, if s does not divide n_2, we can use a similar argument, but some of the subsequences in (5) will be of length $m_j = \lfloor n_2/s\rfloor$ and some of length $m_j = \lceil n_2/s\rceil$. But for $s \leq n_2$,

$$n_2/(2s) \leq \lfloor n_2/s\rfloor \leq n_2/s \leq \lceil n_2/s\rceil \leq n_2/s + 1 \leq 2n_2/s.$$

So the same argument shows that

$$\Pr\left(\left\|\frac{1}{n_2}\sum_{i=n_1}^{n_1+n_2-1} f(X_i) - E_\pi f\right\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right)$$
$$\leq s \max_j \Pr\left(\left\|\frac{1}{m_j}\sum_{i=0}^{m_j-1} f(X_{is+j}) - E_\pi f\right\|_\infty \geq \frac{\epsilon}{2} \,\middle|\, X_{-\infty}^0\right)$$
$$\leq \frac{sd}{2}e^{-\lfloor n_1/\tau\rfloor} + \frac{n_2 d}{2}e^{-\lfloor s/\tau\rfloor} + 2sd\exp\left(\frac{-\epsilon^2 n_2}{4(b-a)^2 s}\right),$$

where the max is over $n_1 \leq j \leq n_1 + s - 1$. □

Simple manipulations and logarithmic inequalities (see, for example, the appendix of [1]) give the following corollary.

Corollary 16. *Suppose that the process*

$$S_t^k = (\nabla_t, R_{t+1}, R_{t+2}, \ldots, R_{t+k})$$

is τ-mixing, $\tau \leq n_2$, and $k \leq n_3 + 1$. Then for

$$n_1 \geq 2\tau \ln(3dn_2/\delta),$$

with probability at least $1 - \delta$ (conditioned on $X_{-\infty}^0$),

$$\|\Delta - \nabla_\beta \eta\|_\infty = O\left(\frac{BC}{1-\beta}\left(\beta^k + \sqrt{\frac{\tau}{n_2}}\ln\left(\frac{n_2 d}{\delta}\right)\right)\right).$$

Equivalently, if

$$k \geq \frac{1}{1-\beta}\ln\left(\frac{4BC}{(1-\beta)\epsilon}\right),$$
$$n_1 \geq 2\tau\ln\left(\frac{3dn_2}{\delta}\right), \text{ and}$$
$$n_2 = \Omega\left(\frac{B^2C^2\tau}{\epsilon^2(1-\beta)^2}\ln^2\left(\frac{d\tau B^2C^2}{\epsilon^2(1-\beta)^2\delta}\right)\right),$$

then

$$\Pr\left(\|\Delta - \nabla_\beta\eta\|_\infty \geq \epsilon \,\middle|\, X_{-\infty}^0\right) \leq \delta.$$

When is S_t^k τ-mixing? Since it is composed of ∇_t and k subsequent reward values, we expect that if the underlying state is rapidly mixing, then so is S_t^k. The following result shows that the mixing time of S_t^k is not much worse than that of the underlying Markov chain.

Lemma 17. *If a Markov chain $\{X_t\}$ is τ-mixing, then the Markov chain $(X_t, X_{t+1}, \ldots, X_{t+k})$ is τ'-mixing, where*

$$\tau' \leq \tau\ln(e(k+1)).$$

Proof. The same argument as in the proof of Lemma 9 shows that if the Markov chain $\{X_t\}$ is τ-mixing, then the conditional distribution p^t of $(X_t, X_{t+1}, \ldots, X_{t+k})$, given $X_{-\infty}^{t_0}$, has

$$\mathrm{d_{TV}}(p^t, \pi) \leq (k+1)\exp\left(-\left\lfloor\frac{t}{\tau}\right\rfloor\right)$$
$$= \exp\left(-\left\lfloor\frac{t}{\tau} - \ln(k+1)\right\rfloor\right)$$
$$\leq \exp\left(-\left\lfloor\frac{t}{\tau(1+\ln(k+1))}\right\rfloor\right).$$

□

Since the stochastic process

$$S_t^k = (\nabla_t, R_{t+1}, \ldots, R_{t+k}),$$

conditioned on $(X_t, \ldots, X_{t+k})$, is i.i.d., this implies that the mixing time of S_t^k is never more than the mixing time of the Markov process $(X_t, X_{t+1}, \ldots, X_{t+k})$. Together with Corollary 16, this gives the following result.

Corollary 18. *If the Markov chain $M(\theta)$ is τ-mixing, then for*

$$n_1 \geq 2\tau \ln(e(n_3+2)) \ln(3dn_2/\delta),$$

for any start state X_0, with probability at least $1-\delta$

$$\|\Delta - \nabla_\beta \eta\|_\infty = O\left(\frac{BC}{1-\beta}\left(\beta^{n_3} + \sqrt{\frac{\tau \ln n_3}{n_2}} \ln\left(\frac{n_2 d}{\delta}\right)\right)\right).$$

Notice that this corollary is weaker than Corollary 16, since the mixing time of $M(\theta)$ provides only a loose upper bound on the mixing time of S_t^k. In particular, suppose the state X_t decomposes into (V_t, W_t), where V_t is rapidly mixing, but W_t is slowly mixing (and the evolution of each is independent of the other). Then if ∇_t and R_t depend only on V_t, they will mix rapidly, but the bound implied by Lemma 17 will be poor. A similar example shows that we cannot obtain a bound on the mixing time of S_t^k in terms of that of ∇_t (or that of R_t): consider what happens if ∇_t depends only on V_t, but R_t depends only on W_t.

6 APPROXIMATION ERROR

The estimate Δ produced by the GPOMDP algorithm converges to $\nabla_\beta \eta$, an approximation to the gradient $\nabla\eta$. In [3], we showed that this approximation is accurate, provided that the time constant $1/(1-\beta)$ is large compared with the mixing time τ^* of the derived Markov chain $M(\theta)$. But the proof in [3] required the assumption that the eigenvalues of the state transition probability matrix of $M(\theta)$ are all distinct. In this section, we present a similar result, but without the restriction on the eigenvalues of the state transition probability matrix. The result is in terms of a slightly different mixing time, based on the χ^2 distance. (Despite the name, the χ^2 distance is not symmetric).

Definition 19. *Given two probability distributions P, π on $\{1, 2, \ldots, N\}$, with $\pi_i > 0$ for all i, the χ^2 distance between P and π is given by*

$$d_{\chi^2}(P, \pi) = \left(\sum_{i=1}^N \frac{(P_i - \pi_i)^2}{\pi_i}\right)^{1/2}.$$

Lemma 20. *For any two probability distributions P, π on $\{1, 2, \ldots, N\}$, with $\pi_i > 0$ for all i,*

$$d_{TV}(P, \pi) \leq d_{\chi^2}(P, \pi).$$

Proof. We can define a vector v with $v_i = |p_i - \pi_i|/\sqrt{\pi_i}$, so that $d_{\chi^2}(P, \pi) = \|v\|$. But $d_T V(P, \pi) = \sqrt{\pi}' v$, where $\sqrt{\pi} = \left[\sqrt{\pi_1}, \ldots, \sqrt{\pi_N}\right]$. Since $\|\sqrt{\pi}\| = 1$, the Cauchy-Schwartz inequality implies the result. □

Theorem 21. *Partition the state transition probability matrix P as*

$$P^t = \begin{bmatrix} {p_1^t}' \\ \vdots \\ {p_N^t}' \end{bmatrix}.$$

Suppose there are constants c, τ^* for which

$$\left(\mathbf{E}_{X\sim\pi} d^2_{\chi^2}(p_X^t, \pi)\right)^{1/2} \leq c \exp\left(-\frac{t}{\tau^*}\right).$$

Then for all $\beta \in [0, 1)$,

$$\|\nabla\eta(\theta) - \beta\nabla_\beta\eta(\theta)\| \leq c\|\nabla\sqrt{\pi}'\|\|\Pi^{1/2} r\|(1-\beta)\tau^*,$$

where $\Pi = \mathrm{diag}(\pi)$.

Notice that $\|\Pi^{-1/2} r\|^2$ is the expectation of R_t^2 under the stationary distribution. This result improves on the corresponding result in [3] by removing the restriction on the distinctness of the eigenvalues of the transition probability matrix. Unfortunately, the constants in this result are not as small as we might like. In particular, it is easy to show that

$$\mathbf{E}_{X\sim\pi} d_{\chi^2}\left(p_X^t, \pi\right)^2 \leq N - 1,$$

and the case $t = 0$ illustrates that this bound is tight. Thus, the constant c in the condition of Theorem 21 must be linear in the size N of the state space, and hence to get a useful bound, $(1-\beta)$ needs to be linear in N. The result in [3] suggests that Theorem 21 can be improved.

The proof of Theorem 21 uses the following lemma.

Lemma 22.

$$\left\|\Pi^{1/2}\left(P^t - e'\pi\right)\Pi^{-1/2}\right\| \leq \sqrt{\mathbf{E}_{X\sim\pi} d_{\chi^2}(p_X^t, \pi)^2}.$$

Proof. Write $p_j^t = (p_{j,1}^t, \ldots, p_{j,N}^t)'$. Then for any $v = (v_1, \ldots, v_N)'$, we have

$$\begin{aligned} & v'\left(\Pi^{1/2}\left(P^t - e'\pi\right)\Pi^{-1/2}\right)v \\ &= \sum_{i,j}\left(p_{j,i}^t - \pi_i\right)\sqrt{\frac{\pi_j}{\pi_i}} v_i v_j \\ &= \sum_j \sqrt{\pi_j} v_j \sum_i \left(p_{j,i}^t - \pi_i\right)\frac{v_i}{\sqrt{\pi_i}} \\ &\leq \|v\| \sum_j \sqrt{\pi_j} v_j d_{\chi^2}(p_j^t, \pi) \\ &\leq \|v\|^2 \left(\sum_j \pi_j d_{\chi^2}(p_j^t, \pi)^2\right)^{1/2}, \end{aligned}$$

where both inequalities follow from the Cauchy-Schwartz inequality. □

Proof. (of Theorem 21) Theorem 5 in [3] shows that

$$
\begin{aligned}
&\|\nabla\eta(\theta) - \beta\nabla_\beta\eta(\theta)\| \\
&= \nabla\pi'(1-\beta)\sum_{t=0}^{\infty}\beta^t P^t r \\
&= \nabla\pi'(1-\beta)\sum_{t=0}^{\infty}\beta^t \left(P^t - e\pi'\right) r \\
&\qquad \text{(because } \nabla\pi' e = \nabla(\pi' e) = 0\text{)} \\
&= \nabla\sqrt{\pi}'(1-\beta)\sum_{t=0}^{\infty}\beta^t \left(\Pi^{1/2}\left(P^t - e\pi'\right)\Pi^{-1/2}\right)\Pi^{1/2} r \\
&\leq \|\nabla\sqrt{\pi}'\|(1-\beta)\sum_{t=0}^{\infty}\beta^t \sqrt{\mathbf{E}_{X\sim\pi} d_{\chi^2}(p_X^t,\pi)^2}\,\|\Pi^{1/2} r\| \\
&\qquad \text{(by Lemma 22)} \\
&\leq \|\nabla\sqrt{\pi}'\|\frac{(1-\beta)c}{1-\beta e^{-1/\tau^*}}\|\Pi^{1/2} r\| \\
&< \|\nabla\sqrt{\pi}'\|(1-\beta)c\tau^*\|\Pi^{1/2} r\|,
\end{aligned}
$$

since $1/(1-e^{-1/\tau^*}) < \tau^*$. □

ACKNOWLEDGEMENTS

This research was partially supported by the Australian Research Council.

References

[1] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, UK, 1999.

[2] L. Baird and A. Moore. Gradient Descent for General Reinforcement Learning. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[3] J. Baxter and P. L. Bartlett. Direct Gradient-Based Reinforcement Learning: I. Gradient Estimation Algorithms. Technical report, Research School of Information Sciences and Engineering, Australian National University, July 1999.

[4] X.-R. Cao and Y.-W. Wan. Algorithms for Sensitivity Analysis of Markov Chains Through Potentials and Perturbation Realization. *IEEE Transactions on Control Systems Technology*, 6:482–492, 1998.

[5] P. R. Halmos. *Measure Theory*. Springer-Verlag, New York, 1974.

[6] H. Kimura, K. Miyazaki, and S. Kobayashi. Reinforcement learning in POMDPs with function approximation. In D. H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 152–160, 1997.

[7] V. R. Konda and J. N. Tsitsiklis. Actor-Critic Algorithms. In *Neural Information Processing Systems 1999*. MIT Press, 2000. To Appear.

[8] P. Marbach. *Simulation-Based Methods for Markov Decision Processes*. PhD thesis, Laboratory for Information and Decision Systems, MIT, 1998.

[9] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. Technical report, MIT, 1998.

[10] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems 1999*. MIT Press, 2000. To Appear.

[11] R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992.