Direct Gradient-Based Reinforcement Learning: I. Gradient Estimation Algorithms

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

July 29, 1999

Abstract

Despite their many empirical successes, approximate value-function based approaches to reinforcement learning suffer from a paucity of theoretical guarantees on the performance of the policy generated by the value-function. In this paper we pursue an alternative approach: first compute the gradient of the *average reward* with respect to the parameters controlling the state transitions in a Markov chain (be they parameters of a class of approximate value functions generating a policy by some form of look-ahead, or parameters directly parameterizing a set of policies), and then use gradient ascent to generate a new set of parameters with increased average reward. We call this method "direct" reinforcement learning because we are not attempting to first find an accurate value-function from which to generate a policy, we are instead adjusting the parameters to directly improve the average reward.

We present an algorithm for computing approximations to the gradient of the average reward from a single sample path of the underlying Markov chain. We show that the accuracy of these approximations depends on the relationship between the discount factor used by the algorithm and the mixing time of the Markov chain, and that the error can be made arbitrarily small by setting the discount factor suitably close to 1. We extend this algorithm to the case of partially observable Markov decision processes controlled by stochastic policies. We prove that both algorithms converge with probability 1.

1 Introduction

Function approximation is essential to avoid the curse of dimensionality associated with large-scale dynamic programming and reinforcement learning problems. The function that is approximated is invariably some measure of the value of a state or of the values of state and action pairs (e.g. $\mathrm{TD}(\lambda)$ [26], Q-learning [31], advantage updating [2]). We will refer to these approaches generically as *value function* based.

The approximating architectures range from linear functions through to neural networks and decision trees. Once an approximate value function has been found, it is

typically used to generate a policy in a greedy fashion by choosing in each state the control (or action) with the highest value as given by the approximate value function. This approach has yielded some remarkable empirical successes in learning to play games, including checkers [21], backgammon [29, 30], and chess [4]. Successes outside of the games domain include job-shop scheduling [35], and dynamic channel allocation [24].

While there are many algorithms for training approximate value functions (see [8, 27] for comprehensive treatments), with varying degrees of convergence guarantees, all these algorithms—and indeed the approximate value function approach itself—suffer from a fundamental limitation: the error they seek to minimize does not guarantee good performance from the resulting policy.

More precisely, there exist infinite horizon Markov Decision Processes[1] (MDPs) with the following properties. For all $\epsilon > 0$ there is an approximate value function V with

$$\max_i |V(i) - V^*(i)| = \epsilon, \tag{1}$$

where the $\max$ is over all states i and $V^*(i)$ is the true value of state i under the optimal policy. However, the greedy policy based on this approximate value function has expected discounted reward

$$\eta = \eta^* - \frac{2\alpha\epsilon}{1-\alpha}, \tag{2}$$

where η^* is the expected discounted reward of the optimal policy and $\alpha \in [0, 1)$ is the discount factor[2]. Thus, even accurate approximations to the optimal value function can generate bad greedy policies if α is close to 1.

Because Equation (2) also defines the *worst* expected discounted reward of any greedy policy derived from an approximate value function satisfying (1), it has sometimes been used as a *motivation* for using approximate value function techniques. However, there are two objections to this. The first is that most existing algorithms for training approximate value functions do not minimize the maximum norm between V and V^*, but typically some ℓ_2 norm. Secondly, even if these algorithms did minimize the maximum norm directly, the smallest achievable error ϵ will be so large in many problems of practical interest that the bound (2) will be useless. Put another way, if we can choose a V to make ϵ arbitrarily small in (1), then we are not really in an approximate value function setting in the first place.

The fact that the class of approximate value functions does not contain a value function with small approximation error does not preclude it from containing a value function whose greedy policy approaches or even equals the performance of the optimal policy. All that matters for the performance of the greedy policy is the relative ordering the approximate value function assigns to the successor states in each state. This motivates an alternative approach: instead of seeking to minimize (1) or an ℓ_2 variant, one should minimize some form of relative error between state values [2, 7, 32].

[1] See Section 2 for definitions.

[2] For a proof of (2) see [8, Proposition 6.1] or [34] or [23].

While this idea is promising, the approach we take in this paper is even more direct: search for a policy minimizing the expected discounted reward directly.

We can view the average reward (2) as a function $\eta(\theta)$ of $\theta \in \mathbb{R}^K$, where θ are the parameters of V. Provided the dependence of η on θ is differentiable[3], we can compute $\nabla\eta(\theta)$ and then take a small step in the gradient direction in order to increase the average reward. Under general assumptions, such an approach will converge to a local maximum of the average reward η.

Note that unlike this gradient-based approach, nearly all value function based approaches cannot guarantee policy improvement on each step. In particular, [32] analyses an MDP for which TD(λ) converges to a value function that gives a suboptimal policy, even when the initial value function gives the optimal policy. It also describes some experimental results for backgammon in which the policy was observed to degrade during training.

The main contribution of this paper is an algorithm for computing an approximation, $\nabla_\beta\eta(\theta)$, to $\nabla\eta(\theta)$, from a single sample path of the underlying Markov chain. The algorithm requires storage of only $2K$ real numbers. The accuracy of the approximation is controlled by a parameter $\beta \in [0, 1)$ of the algorithm (a discount factor) and, in particular, the relationship between β and the mixing time of the Markov chain. The approximation $\nabla_\beta\eta(\theta)$ has the property that $\lim_{\beta\to 1} \nabla_\beta\eta(\theta) = \nabla\eta(\theta)$. However, the trade-off preventing the setting of β arbitrarily close to 1 is that the variance of the algorithm's estimates increase as $\beta \to 1$.

We prove convergence with probability 1 of our algorithm and show that the same algorithm computes the gradient for both average reward and discounted reward problems. We present algorithms for both general parameterized Markov chains and partially observable Markov decision processes (POMDPs) controlled by parameterized stochastic policies. In the latter case, our algorithm needs no knowledge of the underlying Markov decision process or the observation process. It only observes the rewards and the observations.

In a companion paper [5], we present the results of several experiments in which the gradient estimates $\nabla_\beta\eta$ were used to optimize the performance of a variety of different MDPs and POMDPs, including a simple three-state Markov chain controlled by a linear function, a two-dimensional "puck" controlled by a neural network, and the call admission problem treated in [19].

1.1 Related Work

The approach we take in this paper is closely related to certain direct adaptive control schemes that are used to tune (deterministic) controllers for discrete time systems. A number of authors [14, 12, 15] have presented algorithms for the approximate computation in closed loop of derivatives of a quadratic cost function with respect to controller parameters. This information is then used to tune the controller. As for the algorithm we present for controlling POMDPs, these schemes do not require a precise model of the system being controlled. However, they are designed for rather restricted classes of

[3] In general, a *greedy* policy based on $V(\theta)$ will give a non-differentiable $\eta(\theta)$. Thus, in this paper we only consider stochastic policies.

systems and performance criteria.

For reinforcement learning problems, Williams' [33] REINFORCE algorithm is perhaps the first example of an algorithm that sought to optimize average reward (or cost) for stochastic policies by following a gradient direction. Approaches that combine both value function (or Q-function) estimation and gradient estimation include [25] and more recently [1] and [28]. These approaches attempt to combine the advantages of gradient estimation and value function approaches, although as yet there has been little empirical or theoretical investigation of their properties.

Kimura *et al.* [17] extended Williams' algorithm to the infinite horizon setting. Their algorithm is identical to the one presented here, except that it uses a discounted combination of *differential* rewards. In fact the use of differential rewards in this setting does not affect the estimates of the algorithm. While the algorithm presented in [17] provides estimates of the expectation under the stationary distribution of the gradient of the discounted reward, we show that these are biased estimates of the gradient of the expected discounted reward. This arises because the stationary distribution itself depends on the parameters. The bias can be reduced by allowing the discount factor to approach 1.

Formulae for the gradient in a Markov process setting were given in [10]. These formulae critically rely on estimates of the *differential reward* of each state, as do the algorithms given in [20]. One difficulty with estimating the differential reward is that it relies on the existence of a single recurrent state i^* for all parameter settings θ. The variance of the estimate of the differential reward of these algorithms is related to the recurrence time for i^*, which may well be very large for some parameter settings θ. (Think of the two-link "acrobot" problem [27, §11.3]: for poor parameter settings θ a recurrent state i^* with short recurrent time will be the "hanging down" position. However, for good parameter settings the pendulum will spend most of its time in the upright position, making the recurrence time to i^* very large.) It may be possible to alleviate these problems by judicially altering the recurrent state as training proceeds, but no algorithms along those lines have been presented.

Approximate algorithms for computing the gradient were also given in [20, 19], one that sought to solve the aforementioned recurrence problem by demanding only recurrence to one of a set of recurrent states, and another that abandoned recurrence and used discounting, which is closer in spirit to our algorithm. However the latter algorithm still used estimates of the differential reward. The fundamental technical difference in this paper is that we make no use of the differential reward either in the construction or analysis of our algorithms.

Another on-line algorithm for approximating $\nabla\eta(\theta)$ was given in [11] (Algorithm 3c), again based upon estimates of the differential reward. One difficulty with the algorithm is that its memory requirements increase unboundedly with increasing accuracy of the approximate gradient.

1.2 Organisation of the paper

The rest of this paper is organised as follows. In Section 2 we introduce reinforcement learning problems as parameterized MDPs and give definitions of two performance measures: the expected discounted reward and the expected average reward. We then

prove that as far as optimizing the parameters of the MDP is concerned, we can deal with either performance measure.

Section 3 describes formally the gradient ascent approach to optimizing the performance of a parameterized Markov chain, and gives a closed-form expression for the gradient as a function of the gradient of the transition matrix. Since the expression for the gradient involves the inversion of an $n \times n$ matrix where n is the number of states of the system, it is not useful for the kind of large systems tackled by approximate reinforcement learning methods. Thus, in Section 4 we introduce the approximation $\nabla_\beta \eta$ to the true gradient $\nabla \eta$ and prove that $\nabla \eta = \lim_{\beta \to 1} \nabla_\beta \eta$. We also show that the quality of the approximation is controlled by the relationship between β and the mixing time of the Markov chain.

Section 5 introduces MCG, an algorithm for estimating $\nabla_\beta \eta$ from a sample path of a parameterized Markov chain. We prove convergence with probability one of MCG. Section 6 introduces POMDPG, an algorithm for estimating the gradient from a sample path of a POMDP that is controlled by a parameterized stochastic policy. We prove convergence of POMDPG with probabilty one, and provide extensions to control-dependent rewards and to infinite control and observation spaces. Section 7 contains some concluding remarks and suggestions for further research.

2 The Reinforcement Learning Problem

We model reinforcement learning in the standard way, as a Markov decision process (MDP) with a finite state space $\mathcal{S} = \{1, \ldots, n\}$, and a stochastic matrix[4] $P = [p_{ij}]$ giving the probability of transition from state i to state j. Each state i has an associated reward $r(i)$. The matrix P belongs to a parameterized class of stochastic matrices, $\mathcal{P} := \{P(\theta) \colon \theta \in \mathbb{R}^K\}$. Denote the Markov chain corresponding to $P(\theta)$ by $M(\theta)$. Throughout, we assume that these Markov chains satisfy the following assumptions:

Assumption 1. *Each $P(\theta) \in \mathcal{P}$ has a unique stationary distribution $\pi(\theta) := [\pi(\theta, 1), \ldots, \pi(\theta, n)]'$ satisfying the* balance equations

$$\pi'(\theta) P(\theta) = \pi'(\theta) \tag{3}$$

(throughout π' denotes the transpose of π).

Assumption 2. *The magnitudes of the rewards, $|r(i)|$, are uniformly bounded by $R < \infty$ for all states i.*

Ordinarily, a discussion of MDP's would not be complete without some mention of the actions available in each state and the space of policies available to the learner. In particular, the parameters θ would usually determine a policy (either directly or indirectly via a value function), which would then determine the transition probabilities $P(\theta)$. However, for our purposes we do not care *how* the dependence of P on θ arises, just that it satisfies Assumption 1 (and some differentiability assumptions that we shall meet in the next section).

[4] A *stochastic* matrix $P = [p_{ij}]$ has $p_{ij} \geq 0$ for all i, j and $\sum_{j=1}^{n} p_{ij} = 1$ for all i.

Note that it is easy to extend these definitions to the case where the rewards also depend on the parameters θ or on the transitions $i \to j$. It is straightforward to extend our algorithms and results to these cases. See Section 6.1 for an illustration.

We first consider *discounted reward* problems. For $\alpha \in [0,1)$ and $\theta \in \mathbb{R}^K$, define the value of each state $i \in \mathcal{S}$ by

$$J_\alpha(\theta, i) := \lim_{N\to\infty} \mathbf{E}_\theta \left[\sum_{t=0}^{N} \alpha^t r(i_t) | i_0 = i \right], \tag{4}$$

where $\mathbf{E}_\theta$ denotes the expectation over all sequences $i_0, i_1, \ldots$, with transitions generated according to $P(\theta)$. Write $J_\alpha(\theta) = [J_\alpha(\theta,1), \ldots, J_\alpha(\theta,n)]'$ or simply $J_\alpha = [J_\alpha(1), \ldots, J_\alpha(n)]'$ when the dependence on θ is obvious.

The goal is to find a $\theta \in \mathbb{R}^K$ maximizing the *expected discounted reward*:

$$\eta_\alpha(\theta) := \sum_{i=1}^{n} \pi(\theta, i) J_\alpha(\theta, i) = \pi' J_\alpha. \tag{5}$$

We also consider *average reward* problems. Define the *average reward* by:

$$\eta(\theta) := \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{n} \pi(\theta, i) \mathbf{E}_\theta \left[\sum_{t=1}^{N} r(i_t) | i_0 = i \right].$$

It can be shown (see [6]) that

$$\begin{aligned} \eta(\theta) &= \sum_{i=1}^{n} \pi(\theta, i) r(i) \\ &= \pi'(\theta) r, \end{aligned} \tag{6}$$

where $r = [r(1), \ldots, r(n)]'$.

Somewhat surprisingly, for any $\alpha \in [0,1)$, optimizing the discounted reward (5) is equivalent to optimizing the average reward (6), as the following theorem demonstrates.

Theorem 1. *For all $\theta \in \mathbb{R}^K$ and $\alpha \in [0,1)$,*

$$\eta_\alpha(\theta) = \frac{\eta(\theta)}{1-\alpha}. \tag{7}$$

Proof. Let $e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]'$ where the "1" is in the ith position. Then,

suppressing θ dependence, we have:

$$\begin{aligned}
\eta_\alpha &= \pi' J_\alpha \\
&= \lim_{N\to\infty} \sum_{i=1}^{n} \pi(i) \left[r(i) + \sum_{i_1=1}^{n} p_{ii_1} \left[\alpha r(i_1) + \sum_{i_2=1}^{n} p_{i_1 i_2} \left[\alpha^2 r(i_2) + \dots \right.\right.\right. \\
&\qquad \left.\left.\left. + \sum_{i_N=1}^{n} p_{i_{N-1} i_N} \alpha^N r(i_N) \right]\right] \dots \right] \\
&= \lim_{N\to\infty} \sum_{i=1}^{n} \pi(i) \left[r(i) + \alpha e_i' P r + \alpha^2 e_i' P^2 r + \dots + \alpha^N e_i' P^N r \right] \\
&= \lim_{N\to\infty} \left[\pi' r + \alpha \pi' P r + \alpha^2 \pi' P^2 r + \dots + \alpha^N \pi' P^N r \right] \\
&= \lim_{N\to\infty} \sum_{t=0}^{N} \alpha^t \pi' r \\
&= \frac{\eta}{1-\alpha},
\end{aligned}$$

where the third-last line follows from $\sum_i \pi(i) e_i' = \pi' I$ and the penultimate line follows from the balance equations (3). □

3 Gradient Ascent for Parameterized Markov Chains

The approach taken to optimization of $\eta(\theta)$ in this paper is *gradient ascent*. That is, repeatedly compute $\nabla\eta(\theta)$ with respect to the parameters θ, and then take a step in the uphill direction: $\theta \leftarrow \theta + \gamma\nabla\eta(\theta)$, for some suitable step-size γ. From (7),

$$\nabla\eta_\alpha(\theta) = \frac{\nabla\eta(\theta)}{1-\alpha} \tag{8}$$

for any $\alpha \in [0, 1)$, so finding $\nabla\eta(\theta)$ is equivalent to finding $\nabla\eta_\alpha(\theta)$.

To ensure the existence of suitable gradients (and the boundedness of certain random variables), we require that the parameterized class of stochastic matrices satisfies the following additional assumption.

Assumption 3. *The derivatives,*

$$\nabla P(\theta) := \left[\frac{\partial p_{ij}(\theta)}{\partial \theta_k} \right]_{i,j=1\dots n; k=1\dots K}$$

exist for all $\theta \in \mathbb{R}^K$. The ratios

$$\left[\frac{\left| \frac{\partial p_{ij}(\theta)}{\partial \theta_k} \right|}{p_{ij}(\theta)} \right]_{i,j=1\dots n; k=1\dots K}$$

are uniformly bounded by $B < \infty$ for all $\theta \in \mathbb{R}^K$.

Now, suppressing θ dependencies, and since the reward r does not depend on θ, we have:

$$\nabla\eta = \nabla\pi' r. \tag{9}$$

(Think of equations like (9) as shorthand notation for K equations of the form

$$\frac{\partial\eta(\theta)}{\partial\theta_k} = \left[\frac{\partial\pi(\theta,1)}{\partial\theta_k},\ldots,\frac{\partial\pi(\theta,n)}{\partial\theta_k}\right][r(1),\ldots,r(n)]'$$

where $k = 1,\ldots,K$. Alternatively, view the equations for each θ_k as stacked up "back into the page" in a tensor-like fashion.) To compute $\nabla\pi$, first differentiate the balance equations (3) to obtain

$$\nabla\pi'(I-P) = \pi'\nabla P. \tag{10}$$

The system of equations (10) is underconstrained because $I-P$ is not invertible (the balance equations show that $I-P$ has a left eigenvector with zero eigenvalue). However, $I-P+e\pi'$, where $e = [1,1,\ldots,1]'$, is invertible [16]. Since $\nabla\pi' e = \nabla(\pi' e) = \nabla(1) = 0$, we can rewrite (10) as

$$\nabla\pi' = \pi'\nabla P\left[I-P+e\pi'\right]^{-1}. \tag{11}$$

Hence,

$$\nabla\eta = \pi'\nabla P\left[I-P+e\pi'\right]^{-1} r. \tag{12}$$

Note that (11) is essentially a proof that $\nabla\pi$ exists under our assumptions.

For MDP's with a sufficiently small number of states, (12) could be solved exactly to yield the precise gradient direction. However, in general, if the state space is small enough that an exact solution of (12) is possible, then it will be small enough to derive the optimal policy using policy iteration and table-lookup, and there would be no point in pursuing a gradient based approach in the first place.

Thus, for problems of practical interest, (12) will be intractable and we will need to find some other way of computing the gradient. One approximate technique for doing this is presented in the next section.

4 Approximating the Gradient in Parameterized Markov Chains

In this section, we show that the gradient can be split into two components, one of which becomes negligible as a discount factor β approaches 1.

Theorem 2. *For all $\theta \in \mathbb{R}^K$ and $\beta \in [0,1)$,*

$$\nabla\eta = (1-\beta)\nabla\pi' J_\beta + \beta\pi'\nabla P J_\beta. \tag{13}$$

Proof. Observe that J_β satisfies the *Bellman* equations:

$$J_\beta = r + \beta P J_\beta. \tag{14}$$

(See, for example, [6]). Hence,

$$\begin{aligned}
\nabla\eta &= \nabla\left[\pi' r\right] \\
&= \nabla\pi'\left[J_\beta - \beta P J_\beta\right] && \text{by (14)} \\
&= \nabla\pi' J_\beta - \beta\nabla\pi' J_\beta + \beta\pi'\nabla P J_\beta && \text{by (10)} \\
&= (1-\beta)\nabla\pi' J_\beta + \beta\pi'\nabla P J_\beta.
\end{aligned}$$

□

We shall see in the next section that the second term in (13) can be estimated from a single sample path of the Markov chain. In fact, Theorem 1 in [17] shows that the gradient estimates of the algorithm presented in that paper converge to $(1-\beta)\pi'\nabla J_\beta$. By the Bellman equations (14), this is equal to $(1-\beta)\beta(\pi'\nabla P J_\beta + \pi'\nabla J_\beta)$, which implies $(1-\beta)\pi'\nabla J_\beta = \beta\pi'\nabla P J_\beta$. Thus the algorithm in [17] estimates the second term in the expression for $\nabla\eta(\theta)$ given by (13).

The following theorem shows that the first term in (13) becomes negligible as β approaches 1. Notice that this is not immediate from Theorem 2, since J_β can become arbitrarily large in the limit $\beta \to 1$.

Theorem 3. *For all $\theta \in \mathbb{R}^K$,*

$$\nabla\eta = \lim_{\beta\to 1} \nabla_\beta\eta, \tag{15}$$

where

$$\nabla_\beta\eta := \pi'\nabla P J_\beta. \tag{16}$$

Proof. Propositions 1.2 and 2.5 in [6, chapter 4] show that

$$\lim_{\beta\to 1}(1-\beta)J_\beta = e\eta. \tag{17}$$

Hence, from Theorem 2,

$$\begin{aligned}
\nabla\eta &= \nabla\pi' e\eta + \lim_{\beta\to 1}\beta\pi'\nabla P J_\beta \\
&= \lim_{\beta\to 1}\pi'\nabla P J_\beta,
\end{aligned}$$

since $\nabla\pi' e = 0$. □

Theorem 3 shows that $\nabla_\beta\eta$ is a good approximation to the gradient as β approaches 1, but it turns out that values of β very close to 1 lead to large variance in the estimates of $\nabla_\beta\eta$ that we describe in the next section. However, the following theorem shows that $1-\beta$ need not be too small, provided the Markov chain has a short *mixing time*. From any initial state, the distribution over states of a Markov chain converges to the

stationary distribution, provided the assumption (Assumption 1) about the existence and uniqueness of the stationary distribution is satisfied (see, for example, [18, Theorem 15.8.1, p. 552]). The spectral resolution theorem [18, Theorem 9.5.1, p. 314] implies that the distribution converges to stationarity at an exponential rate, and the time constant in this convergence rate (the mixing time) depends on the eigenvalues of the transition probability matrix. The existence of a unique stationary distribution implies that the largest magnitude eigenvalue is 1 and has multiplicity 1, and the corresponding left eigenvector is the stationary distribution. We order the eigenvalues in decreasing order of magnitude, so that $1 = \lambda_1 > |\lambda_2| > \cdots > |\lambda_s|$ for some $2 \leq s \leq n$. It turns out that $|\lambda_2|$ determines the mixing time of the chain.

The following theorem shows that if $1-\beta$ is small compared to $1-|\lambda_2|$, the gradient approximation described above is accurate. Since we will be using the estimate as a direction in which to update the parameters, the theorem compares the *directions* of the gradient and its estimate. In this theorem, $\kappa_2(A)$ denotes the *spectral condition number* of a nonsingular matrix A, which is defined as the product of the *spectral norms* of the matrices A and A^{-1},

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2,$$

where

$$\|A\|_2 = \max_{x:\|x\|=1} \|Ax\|,$$

and $\|x\|$ denotes the Euclidean norm of the vector x.

Theorem 4. *Suppose that the transition probability matrix $P(\theta)$ satisfies Assumption 1 with stationary distribution $\pi' = (\pi_1, \ldots, \pi_n)$, and has n distinct eigenvectors. Let $S = (x_1 x_2 \cdots x_n)$ be the matrix of right eigenvectors of P corresponding, in order, to the eigenvalues $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Then the normalized inner product between $\nabla\eta$ and $\beta\nabla_\beta\eta$ satisfies*

$$1 - \frac{\nabla\eta \cdot \beta\nabla_\beta\eta}{\|\nabla\eta\|^2} \leq \kappa_2\left(\Pi^{1/2}S\right) \frac{\|\nabla(\sqrt{\pi}_1, \ldots, \sqrt{\pi}_n)\|}{\|\nabla\eta\|} \sqrt{r'\Pi r} \frac{1-\beta}{1-\beta|\lambda_2|}, \qquad (18)$$

where $\Pi = \mathrm{diag}(\pi_1, \ldots, \pi_n)$.

Notice that $r'\Pi r$ is the expectation under the stationary distribution of $r(X)^2$.

As well as the mixing time (via $|\lambda_2|$), the bound in the theorem depends on another parameter of the Markov chain: the spectral condition number of $\Pi^{1/2}S$. If the Markov chain is reversible (that is, the transition probability matrix is symmetric and hence the eigenvectors $x_1, \ldots, x_n$ are orthogonal), this is equal to the ratio of the maximum to the minimum probability of states under the stationary distribution. However, the eigenvectors do not need to be nearly orthogonal. In fact, the condition that the transition probability matrix have n distinct eigenvectors is not necessary; without it, the condition number is replaced by a more complicated expression involving spectral norms of matrices of the form $(P - \lambda_i I)$. We will elaborate on this further in [3].

Proof. The existence of n distinct eigenvectors implies that P can be expressed as $S\Lambda S^{-1}$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ (see [18, Theorem 4.10.2, p 153]). It follows that for any polynomial f, we can write $f(P) = Sf(\Lambda)S^{-1}$.

Now, Theorem 2 shows that $\nabla\eta - \beta\nabla_\beta\eta = \nabla\pi'(1-\beta)J_\beta$. But

$$\begin{aligned}
(1-\beta)J_\beta &= (1-\beta)\left(r + \beta Pr + \beta^2 P^2 r + \cdots\right) \\
&= (1-\beta)\left(I + \beta P + \beta^2 P^2 + \cdots\right) r \\
&= (1-\beta)S\left(\sum_{t=0}^{\infty}\beta^t\Lambda^t\right)S^{-1}r \\
&= (1-\beta)\sum_{j=1}^{n} x_i y_i'\left(\sum_{t=0}^{\infty}(\beta\lambda_j)^t\right) r,
\end{aligned}$$

where

$$S^{-1} = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}.$$

It is easy to verify that y_i is the left eigenvector corresponding to λ_i, and that we can choose $y_1 = \pi$ and $x_1 = e$. Thus we can write

$$\begin{aligned}
(1-\beta)J_\beta &= (1-\beta)e\pi' r + \sum_{j=2}^{n} x_i y_i'\left(\sum_{t=0}^{\infty}(1-\beta)(\beta\lambda_j)^t\right) r \\
&= (1-\beta)e\eta + \sum_{j=2}^{n} x_i y_i'\left(\frac{1-\beta}{1-\beta\lambda_i}\right) r \\
&= (1-\beta)e\eta + SMS^{-1}r,
\end{aligned}$$

where

$$M = \operatorname{diag}\left(0, \frac{1-\beta}{1-\beta\lambda_2}, \ldots, \frac{1-\beta}{1-\beta\lambda_n}\right).$$

It follows from this and Theorem 2 that

$$\begin{aligned}
1 - \frac{\nabla\eta\cdot\beta\nabla_\beta\eta}{\|\nabla\eta\|^2} &= 1 - \frac{\nabla\eta\cdot(\nabla\eta - \nabla\pi'(1-\beta)J_\beta)}{\|\nabla\eta\|^2} \\
&= \frac{\nabla\eta\cdot\nabla\pi'(1-\beta)J_\beta}{\|\nabla\eta\|^2} \\
&= \frac{\nabla\eta\cdot\nabla\pi'\left((1-\beta)e\eta + SMS^{-1}r\right)}{\|\nabla\eta\|^2} \\
&= \frac{\nabla\eta\cdot\nabla\pi' SMS^{-1}r}{\|\nabla\eta\|^2} \\
&\leq \frac{\left\|\nabla\pi' SMS^{-1}r\right\|}{\|\nabla\eta\|},
\end{aligned}$$

by Cauchy-Schwartz' inequality. Since $\nabla\pi' = \nabla\left(\sqrt{\pi'}\right)\Pi^{1/2}$, we can apply Cauchy-

Schwartz' inequality again to obtain

$$1 - \frac{\nabla\eta \cdot \beta\nabla_\beta\eta}{\|\nabla\eta\|^2} \leq \frac{\left\|\nabla\left(\sqrt{\pi'}\right)\right\| \left\|\Pi^{1/2}SMS^{-1}r\right\|}{\|\nabla\eta\|}. \tag{19}$$

We use spectral norms to bound the second factor in the numerator. It is clear from the definition that the spectral norm of a product of nonsingular matrices satisfies $\|AB\|_2 \leq \|A\|_2\|B\|_2$, and that the spectral norm of a diagonal matrix is given by $\|\operatorname{diag}(d_1, \ldots, d_n)\|_2 = \max_i |d_i|$. It follows that

$$\begin{aligned}
\left\|\Pi^{1/2}SMS^{-1}r\right\| &= \left\|\Pi^{1/2}SMS^{-1}\Pi^{-1/2}\Pi^{1/2}r\right\| \\
&\leq \left\|\Pi^{1/2}S\right\|_2 \left\|S^{-1}\Pi^{-1/2}\right\|_2 \left\|\Pi^{1/2}r\right\| \|M\|_2 \\
&\leq \kappa_2\left(\Pi^{1/2}S\right)\sqrt{r'\Pi r}\frac{1-\beta}{1-\beta|\lambda_2|}.
\end{aligned}$$

Combining with Equation (19) proves (18). □

5 Estimating the Gradient in Parameterized Markov Chains

Algorithm 1 introduces MCG (**M**arkov **C**hain **G**radient), an algorithm for estimating the approximate gradient $\nabla_\beta\eta$ from a single on-line sample path $i_0, i_1, \ldots$ from the Markov chain $M(\theta)$. MCG requires only $2K$ reals to be stored, where K is the dimension of the parameter space.

Theorem 5. *Under Assumptions 1, 2 and 3, the* MCG *algorithm starting from any initial state i_0 will generate a sequence $\Delta_0, \Delta_1, \ldots, \Delta_t, \ldots$ satisfying*

$$\lim_{t\to\infty} \Delta_t = \nabla_\beta\eta \quad \text{w.p.1.} \tag{20}$$

Proof. Let $X_0, X_1, \ldots$ denote the random process corresponding to $M(\theta)$. By Assumption 1, $\{X_t\}$ is asymptotically stationary, and we can write

$$\begin{aligned}
\pi'\nabla P J_\beta &= \sum_{i,j} \pi(i)\nabla p_{ij}(\theta) J_\beta(j) \\
&= \sum_{i,j} \pi(i) p_{ij}(\theta)\frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)} J_\beta(j) \\
&= \sum_{i,j} \Pr(X_t = i)\Pr(X_{t+1} = j | X_t = i)\frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)}\mathbf{E}(J(t+1)|X_{t+1} = j),
\end{aligned} \tag{21}$$

where the first probability is with respect to the stationary distribution and $J(t+1)$ is the process

$$J(t+1) = \sum_{s=t+1}^{\infty} \beta^{s-t-1} r(X_s).$$

Algorithm 1 The MCG (**M**arkov **C**hain **G**radient) algorithm

Given:

- Parameter $\theta \in \mathbb{R}^K$.
- Parameterized class of stochastic matrices $\mathcal{P} = \{P(\theta) : \theta \in \mathbb{R}^K\}$ satisfying Assumptions 3 and 1.
- $\beta \in [0, 1)$.
- Arbitrary starting state i_0.
- State sequence $i_0, i_1, \ldots$ generated by $M(\theta)$ (i.e. the Markov chain with transition probabilities $P(\theta)$).
- Reward sequence $r(i_0), r(i_1), \ldots$ satisfying Assumption 2.

Set $z_0 = 0$ and $\Delta_0 = 0$ ($z_0, \Delta_0 \in \mathbb{R}^K$).
for each state i_{t+1} visited **do**
    $z_{t+1} = \beta z_t + \frac{\nabla p_{i_t i_{t+1}}(\theta)}{p_{i_t i_{t+1}}(\theta)}$
    $\Delta_{t+1} = \Delta_t + \frac{1}{t+1}\left[r(i_{t+1}) z_{t+1} - \Delta_t\right]$
end for

The fact that $\mathbf{E}(J(t+1)|X_{t+1}) = J_\beta(X_{t+1})$ for all X_{t+1} follows from the boundedness of the magnitudes of the rewards (Assumption 2) and Lebesgue's dominated convergence theorem. We can rewrite Equation (21) as

$$\pi' \nabla P J_\beta = \sum_{i,j} \mathbf{E}\left[\chi_i(X_t)\chi_j(X_{t+1}) \frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)} J(t+1)\right],$$

where $\chi_i(\cdot)$ denotes the indicator function for state i,

$$\chi_i(X_t) := \begin{cases} 1 & \text{if } X_t = i, \\ 0 & \text{otherwise,} \end{cases}$$

and the expectation is again with respect to the stationary distribution. When X_t is chosen according to the stationary distribution, the process $\{X_t\}$ is ergodic. Since the process $\{Z_t\}$ defined by

$$Z_t := \chi_i(X_t)\chi_j(X_{t+1}) \frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)} J(t+1)$$

is obtained by taking a fixed function of $\{X_t\}$, $\{Z_t\}$ is also stationary and ergodic (see [9, Proposition 6.31]). Since $\left|\frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)}\right|$ is bounded by Assumption 3, from the

ergodic theorem we have (almost surely):

$$\begin{aligned}
\pi' \nabla P J_\beta &= \sum_{i,j} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \chi_i(X_t)\chi_j(X_{t+1}) \frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)} J(t+1) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} J(t+1) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} \left[\sum_{s=t+1}^{T} \beta^{s-t-1} r(X_s) + \sum_{s=T+1}^{\infty} \beta^{s-t-1} r(X_s) \right].
\end{aligned} \tag{22}$$

Concentrating on the second term in the right-hand-side of (22), observe that:

$$\begin{aligned}
&\left| \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} \sum_{s=T+1}^{\infty} \beta^{s-t-1} r(X_s) \right| \\
&\qquad \leq \frac{1}{T} \sum_{t=0}^{T-1} \left| \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} \right| \sum_{s=T+1}^{\infty} \beta^{s-t-1} \left| r(X_s) \right| \\
&\qquad \leq \frac{BR}{T} \sum_{t=0}^{T-1} \sum_{s=T+1}^{\infty} \beta^{s-t-1} \\
&\qquad = \frac{BR}{T} \sum_{t=0}^{T-1} \frac{\beta^{T-t}}{1-\beta} \\
&\qquad = \frac{BR\beta\left(1-\beta^T\right)}{T\left(1-\beta\right)^2} \\
&\qquad \to 0 \text{ as } T \to \infty,
\end{aligned}$$

where R and B are the bounds on the magnitudes of the rewards and $\frac{|\nabla p_{ij}|}{p_{ij}}$ from Assumptions 2 and 3. Hence,

$$\pi' \nabla P J_\beta = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} \sum_{s=t+1}^{T} \beta^{s-t-1} r(X_s) \tag{23}$$

Unrolling the equation for Δ_T in the MCG algorithm shows it is equal to

$$\frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{i_t i_{t+1}}(\theta)}{p_{i_t i_{t+1}}(\theta)} \sum_{s=t+1}^{T} \beta^{s-t-1} r(i_s),$$

hence $\Delta_T \to \pi' \nabla P J_\beta$ w.p.1 as required. □

6 Estimating the Gradient in Partially Observable Markov Decision Processes

Algorithm 1 applies to any parameterized class of stochastic matrices $P(\theta)$ for which we can compute the gradients $\nabla p_{ij}(\theta)$. In this section we consider the special case of $P(\theta)$ that arise from a parameterized class of randomized policies controlling a partially observable Markov decision process (POMDP). The 'partially observable' qualification means we assume that these policies have access to an observation process that depends on the state, but in general they may not see the state.

Specifically, assume that there are N controls $\mathcal{U} = \{1, \ldots, N\}$ and M observations $\mathcal{Y} = \{1, \ldots, M\}$. Each $u \in \mathcal{U}$ determines a stochastic matrix $P(u)$ which does not depend on the parameters θ. 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 $\nu_y(i)$. A *randomized policy* is simply a function μ mapping observations $y \in \mathcal{Y}$ into probability distributions over the controls $\mathcal{U}$. That is, for each observation y, $\mu(y)$ is a distribution over the controls in $\mathcal{U}$. Denote the probability under μ of control u given observation y by $\mu_u(y)$.

To each randomized policy $\mu(\cdot)$ and observation distribution $\nu(\cdot)$ there corresponds a Markov chain in which state transitions are generated by first selecting an observation y in state i according to the distribution $\nu(i)$, then selecting a control u according to the distribution $\mu(y)$, and then generating a transition to state j according to the probability $p_{ij}(u)$. To parameterize these chains we parameterize the policies, so that μ now becomes a function $\mu(\theta, y)$ of a set of parameters $\theta \in \mathbb{R}^K$ as well as the observation y. The Markov chain corresponding to θ has state transition matrix $[p_{ij}(\theta)]$ given by

$$p_{ij}(\theta) = \mathbf{E}_{y \sim \nu(i)} \mathbf{E}_{u \sim \mu(\theta, y)} p_{ij}(u). \tag{24}$$

Equation (24) implies

$$\nabla p_{ij}(\theta) = \sum_{u,y} \nu_y(i) p_{ij}(u) \nabla \mu_u(\theta, y). \tag{25}$$

Algorithm 2 introduces the POMDPG algorithm (for **P**artially **O**bservable **M**arkov **D**ecision **P**rocess **G**radient), a modified form of Algorithm 1 in which updates of z_t are based on $\mu_{u_t}(\theta, y_t)$, rather than $p_{i_t i_{t+1}}(\theta)$. Note that Algorithm 2 does not require knowledge of the transition probability matrix P, nor of the observation process ν; it only requires knowledge of the randomized policy μ.

For convergence of Algorithm 2 we need to replace Assumption 3 with a similar bound on the gradient of μ:

Assumption 4. *The derivatives,*

$$\frac{\partial \mu_u(\theta, y)}{\partial \theta_k}$$

exist for all $u \in \mathcal{U}$, $y \in \mathcal{Y}$ and $\theta \in \mathbb{R}^K$. The ratios

$$\left[\frac{\left| \frac{\partial \mu_u(\theta, y)}{\partial \theta_k} \right|}{\mu_u(\theta, y)} \right]_{y=1\ldots M; u=1\ldots N; k=1\ldots K}$$

Algorithm 2 The POMDPG algorithm.

1: **Given:**

- Parameterized class of randomized policies $\{\mu(\theta,\cdot) : \theta \in \mathbb{R}^K\}$ satisfying Assumption 4.
- Partially observable Markov decision process 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 $y_0, y_1, \ldots$ generated by the POMDP with controls $u_0, u_1, \ldots$ generated randomly according to $\mu(\theta, y_t)$.
- Reward sequence $r(i_0), r(i_1), \ldots$ satisfying Assumption 2, where $i_0, i_1, \ldots$ is the (hidden) sequence of states of the Markov decision process.

2: Set $z_0 = 0$ and $\Delta_0 = 0$ ($z_0, \Delta_0 \in \mathbb{R}^K$).
3: **for** each observation y_t, control u_t, and subsequent reward $r(i_{t+1})$ **do**
4: $\quad z_{t+1} = \beta z_t + \frac{\nabla \mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)}$
5: $\quad \Delta_{t+1} = \Delta_t + \frac{1}{t+1}\left[r(i_{t+1}) z_{t+1} - \Delta_t\right]$
6: **end for**

are uniformly bounded by $B_\mu < \infty$ for all $\theta \in \mathbb{R}^K$.

Theorem 6. *Under Assumptions 1, 2 and 4, Algorithm 2 starting from any initial state i_0 will generate a sequence $\Delta_0, \Delta_1, \ldots, \Delta_t, \ldots$ satisfying*

$$\lim_{t\to\infty} \Delta_t = \nabla_\beta \eta \quad \text{w.p.1.} \tag{26}$$

Proof. The proof follows the same lines as the proof of Theorem 5. In this case,

$$\begin{aligned}
\pi' \nabla P J_\beta &= \sum_{i,j} \pi(i) \nabla p_{ij}(\theta) J_\beta(j) \\
&= \sum_{i,j,y,u} \pi(i) p_{ij}(u) \nu_y(i) \nabla \mu_u(\theta, y) J_\beta(j) \text{ from (25)} \\
&= \sum_{i,j,y,u} \pi(i) p_{ij}(u) \nu_y(i) \frac{\nabla \mu_u(\theta, y)}{\mu_u(\theta, y)} \mu_u(\theta, y) J_\beta(j), \\
&= \sum_{i,j,y,u} \mathbf{E} Z_t',
\end{aligned}$$

where the expectation is with respect to the stationary distribution of $\{X_t\}$, and the

process $\{Z'_t\}$ is defined by

$$Z'_t := \chi_i(X_t)\chi_j(X_{t+1})\chi_u(U_t)\chi_y(Y_t)\frac{\nabla\mu_u(\theta,y)}{\mu_u(\theta,y)}J(t+1),$$

where U_t is the control process and Y_t is the observation process. The result follows from the same arguments used in the proof of Theorem 5. □

6.1 Control dependent rewards $\mathbf{r(u,i)}$

There are many circumstances in which the rewards may themselves depend on the controls u. For example, some controls may consume more energy than others and so we may wish to add a penalty term to the reward function in order to conserve energy. The simplest way to deal with this is to define for each state i the expected reward $\bar{r}(i)$ by

$$\bar{r}(i) = \mathbf{E}_{y\sim\nu(i)}\mathbf{E}_{u\sim\mu(\theta,y)}r(u,i), \tag{27}$$

and then redefine J_β in terms of $\bar{r}$:

$$\bar{J}_\beta(\theta,i) := \lim_{N\to\infty}\mathbf{E}_\theta\left[\sum_{t=0}^{N}\beta^t\bar{r}(i_t)|i_0=i\right], \tag{28}$$

where the expectation is over all trajectories $i_0, i_1, \ldots$. The performance gradient then becomes

$$\nabla\eta = \nabla\pi'\bar{r} + \pi'\nabla\bar{r},$$

which can be approximated by

$$\nabla_\beta\eta = \pi'\left[\nabla P\bar{J}_\beta + \nabla\bar{r}\right],$$

due to the fact that $\bar{J}_\beta$ satisfies the Bellman equations (14) with $\bar{r}$ replaced by r.

For POMDPG to take account of the dependence of r on the controls, one simply replaces its fifth line by

$$\Delta_{t+1} = \Delta_t + \frac{1}{t+1}\left[r(u_{t+1},i_{t+1})\left(z_{t+1} + \frac{\nabla\mu_{u_{t+1}}(\theta,y_{t+1})}{\mu_{u_{t+1}}(\theta,y_{t+1})}\right) - \Delta_t\right].$$

It is straightforward to extend the proofs of Theorems 3, 4 and 6 to this setting.

6.2 Extensions to infinite state, observation, and control spaces

The convergence proof for Algorithm 2 relied on finite state ($\mathcal{S}$), observation ($\mathcal{Y}$) and control ($\mathcal{U}$) spaces. However, it should be clear that with no modification Algorithm 2 can be applied immediately to POMDPs with countably or uncountably infinite $\mathcal{S}$ and $\mathcal{Y}$, and countable $\mathcal{U}$. In addition, with the appropriate interpretation of $\nabla\mu/\mu$, it can be applied to uncountable $\mathcal{U}$. Specifically, if $\mathcal{U}$ is a subset of $\mathbb{R}^N$ then $\mu(y,\theta)$ will be a probability *density* function on $\mathcal{U}$ with $\mu_u(y,\theta)$ the density at u. Theorem 6 can

be extended to show that the estimates produced by this algorithm converge almost surely to $\nabla_\beta \eta$. In fact, we can prove a more general result that implies both this case of densities on subsets of $\mathbb{R}^N$ as well as the finite case of Theorem 6. We allow $\mathcal{U}$ and $\mathcal{Y}$ to be general spaces satisfying the following topological assumption. (For definitions see, for example, [13].)

Assumption 5. *The control space $\mathcal{U}$ has an associated topology that is separable, Hausdorff, and first-countable. For the corresponding Borel σ-algebra $\mathcal{B}$ generated by this topology, there is a σ-finite measure λ defined on the measurable space $(\mathcal{U}, \mathcal{B})$. We say that λ is the* reference measure *for $\mathcal{U}$.*

Similarly, the observation space $\mathcal{Y}$ has a topology, Borel σ-algebra, and reference measure satisfying the same conditions.

In the case of Theorem 6, where $\mathcal{U}$ and $\mathcal{Y}$ are finite, the associated reference measure is the counting measure. For $\mathcal{U} = \mathbb{R}^N$ and $\mathcal{Y} = \mathbb{R}^M$, the reference measure is Lebesgue measure. We assume that the distributions $\nu(i)$ and $\mu(\theta, y)$ are absolutely continuous with respect to the reference measures, and the corresponding Radon-Nikodym derivatives (probability masses, in the finite case; densities in the Euclidean case) satisfy the following assumption.

Assumption 6. *For every $y \in \mathcal{Y}$ and $\theta \in \mathbb{R}^K$, the probability measure $\mu(\theta, y)$ is absolutely continuous with respect to the reference measure for $\mathcal{U}$. For every $i \in \mathcal{S}$, the probability measure $\nu(i)$ is absolutely continuous with respect to the reference measure for $\mathcal{Y}$.*

Let λ be the reference measure for $\mathcal{U}$. For all $u \in \mathcal{U}$, $y \in \mathcal{Y}$, $\theta \in \mathbb{R}^K$, and $k \in \{1, \ldots, K\}$, the derivatives

$$\frac{\partial}{\partial \theta_k} \frac{d\mu(\theta, y)}{d\lambda}(u)$$

exist and the ratios

$$\frac{\left| \frac{\partial}{\partial \theta_k} \frac{d\mu_u(\theta, y)}{d\lambda}(u) \right|}{\frac{d\mu_u(\theta, y)}{d\lambda}(u)}$$

are bounded by $B_\mu < \infty$.

With these assumptions, we can replace μ in Algorithm 2 with the Radon-Nikodym derivative of μ with respect to the reference measure on $\mathcal{U}$. In this case, we have the following convergence result. This generalizes Theorem 6, and also applies to densities μ on a Euclidean space $\mathcal{U}$.

Theorem 7. *Suppose the control space $\mathcal{U}$ and the observation space $\mathcal{Y}$ satisfy Assumption 5 and let λ be the reference measure on the control space $\mathcal{U}$. Consider Algorithm 2 with $\nabla \mu_{u_t}(\theta, y_t) / \mu_{u_t}(\theta, y_t)$ replaced by*

$$\frac{\nabla \frac{d\mu(\theta, y_t)}{d\lambda}(u_t)}{\frac{d\mu(\theta, y_t)}{d\lambda}(u_t)}.$$

Under Assumptions 1, 2 and 6, this algorithm, starting from any initial state i_0 will generate a sequence $\Delta_0, \Delta_1, \ldots, \Delta_t, \ldots$ satisfying

$$\lim_{t\to\infty} \Delta_t = \nabla_\beta \eta \quad \text{w.p.1.}$$

The proof needs the following topological lemma. For definitions see, for example, [13, pp. 24–25].

Lemma 1. *Let $(X, \mathcal{T})$ be a topological space that is Hausdorff, separable, and first-countable. Let $\mathcal{B}$ be the Borel σ-algebra generated by $\mathcal{T}$. Then the measurable space $(X, \mathcal{B})$ has a sequence $\mathcal{S}_1, \mathcal{S}_2, \ldots \subseteq \mathcal{B}$ of sets that satisfies the following conditions:*

1. *Each $\mathcal{S}_i$ is a partition of X (that is, $X = \bigcup\{S\colon S \in \mathcal{S}_i\}$ and any two distinct elements of $\mathcal{S}_i$ have empty intersection).*
2. *For all $x \in X$, $\{x\} \in \mathcal{B}$ and*

$$\bigcap_{i=1}^{\infty} \{S \in \mathcal{S}_i : x \in S\} = \{x\}.$$

Proof. Since X is separable, it has a countable dense subset $S = \{x_1, x_2, \ldots\}$. Since X is first-countable, each of these x_i has a countable neighbourhood base, N_i. Now, construct the partitions $\mathcal{S}_i$ using the countable set $N = \bigcup_{i=1}^{\infty} N_i$ as follows. Let $\mathcal{S}_0 = X$ and, for $i = 1, 2, \ldots$, define

$$\mathcal{S}_i = \{S \cap N_i : S \in \mathcal{S}_{i-1}\} \cup \{S \cap (X - N_i) : S \in \mathcal{S}_{i-1}\}.$$

Clearly, each $\mathcal{S}_i$ is a measurable partition of X. Since X is Hausdorff, for each pair x, x' of distinct points from X, there is a pair of disjoint open sets A and A' such that $x \in A$ and $x' \in A'$. Since S is dense, there is a pair s, s' from S with $s \in A$ and $s' \in A'$. Also, N contains neighbourhoods N_s and $N_{s'}$ with $N_s \subseteq A$ and $N_{s'} \subseteq A'$. So N_s and $N_{s'}$ are disjoint. Thus, for sufficiently large i, x and x' fall in distinct elements of the partition $\mathcal{S}_i$. Since this is true for any pair x, x', it follows that

$$\bigcap_{i=1}^{\infty} \{S \in \mathcal{S}_i : x \in S\} \subseteq \{x\}.$$

The reverse inclusion is trivial. The measurability of all singletons $\{x\}$ follows from the measurability of $S_x := \bigcup_i \{S \in \mathcal{S}_i\colon S \cap \{x\} = \phi\}$ and the fact that $\{x\} = X - S_x$. □

We shall use Lemma 1 together with the following result to show that we can approximate expectations of certain random variables using a single sample path of the Markov chain.

Lemma 2. *Let $(X, \mathcal{B})$ be a measurable space satisfying the conditions of Lemma 1, and let $\mathcal{S}_1, \mathcal{S}_2, \ldots$ be a suitable sequence of partitions as in that lemma. Let μ be a*

probability measure defined on this space. Let f be an absolutely integrable function on X. For an event S, define

$$f(S) = \frac{\int_S f \, d\mu}{\mu(S)}.$$

For each $x \in X$ and $k = 1, 2, \ldots$, let $S_k(x)$ be the unique element of $\mathcal{S}_k$ containing x. Then for almost all x in X,

$$\lim_{k\to\infty} f(S_k(x)) = f(x).$$

Proof. Clearly, the signed finite measure ϕ defined by

$$\phi(E) = \int_E f d\mu \tag{29}$$

is absolutely continuous with respect to μ, and Equation (29) defines f as the Radon-Nikodym derivative of ϕ with respect to μ. This derivative can also be defined as

$$\frac{d\phi}{d\mu}(x) = \lim_{k\to\infty} \frac{\phi(S_k(x))}{\mu(S_k(x))}.$$

See, for example, [22, Section 10.2]. By the Radon-Nikodym Theorem [13, Theorem 5.5.4, p. 134], these two expressions are equal a.e. (μ). □

Proof. (Theorem 7.) From the definitions,

$$\begin{aligned}\nabla_\beta \eta &= \pi' \nabla P J_\beta \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \pi(i) \nabla p_{ij}(\theta) J_\beta(j).\end{aligned} \tag{30}$$

For every y, μ is absolutely continuous with respect to the reference measure λ, hence for any i and j we can write

$$p_{ij}(\theta) = \int_{\mathcal{Y}} \int_{\mathcal{U}} p_{ij}(u) \, \frac{d\mu(\theta, y)}{d\lambda}(u) \, d\lambda(u) \, d\nu(i)(y).$$

Since λ and ν do not depend on θ and $d\mu(\theta, y)/d\lambda$ is absolutely integrable, we can differentiate under the integral to obtain

$$\nabla p_{ij}(\theta) = \int_{\mathcal{Y}} \int_{\mathcal{U}} p_{ij}(u) \, \nabla \frac{d\mu(\theta, y)}{d\lambda}(u) \, d\lambda(u) \, d\nu(i)(y).$$

To avoid cluttering the notation, we shall use μ to denote the distribution $\mu(\theta, y)$ on $\mathcal{U}$, and ν to denote the distribution $\nu(i)$ on $\mathcal{Y}$. With this notation, we have

$$\nabla p_{ij}(\theta) = \int_{\mathcal{Y}} \int_{\mathcal{U}} p_{ij} \, \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}} \, d\mu \, d\nu.$$

Now, let ρ be the probability measure on $\mathcal{Y}\times\mathcal{U}$ generated by μ and ν. We can write (30) as

$$\nabla_\beta \eta = \sum_{i,j} \pi(i) J_\beta(j) \int_{\mathcal{Y}\times\mathcal{U}} p_{ij} \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}}\, d\rho.$$

Using the notation of Lemma 2, we define

$$p_{ij}(S) = \frac{\int_S p_{ij}\, d\rho}{\rho(S)},$$
$$\nabla(S) = \frac{1}{\rho(S)} \int_S \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}}\, d\rho,$$

for a measurable set $S \subseteq \mathcal{Y}\times\mathcal{U}$. Notice that, for a given i, j, and S,

$$p_{ij}(S) = \Pr\left(X_{t+1} = j \,|X_t = i,\ (y,u) \in S\right)$$
$$\nabla(S) = \mathbf{E}\left(\frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}} \middle| X_t = i,\ (Y_t, U_t) \in S \right).$$

Let $\mathcal{S}_1, \mathcal{S}_2, \ldots$ be a sequence of partitions of $\mathcal{Y}\times\mathcal{U}$ as in Lemma 1, and let $S_k(y,u)$ denote the element of $\mathcal{S}_k$ containing (y,u). Using Lemma 2, we have

$$\begin{aligned}\int_{\mathcal{Y}\times\mathcal{U}} p_{ij} \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}}\, d\rho &= \int_{\mathcal{Y}\times\mathcal{U}} \lim_{k\to\infty} p_{ij}\left(S_k(y,u)\right) \nabla\left(S_k(y,u)\right)\, d\rho(y,u) \\ &= \lim_{k\to\infty} \sum_{S\in\mathcal{S}_k} \int_S p_{ij}(S)\, \nabla(S)\, d\rho,\end{aligned}$$

where we have used Assumption 6 and the Lebesgue dominated convergence theorem to interchange the integral and the limit. Hence,

$$\begin{aligned}\nabla_\beta \eta &= \lim_{k\to\infty} \sum_{i,j} \sum_{S\in\mathcal{S}_k} \pi(i)\rho(S) p_{ij}(S) J_\beta(j) \nabla(S) \\ &= \lim_{k\to\infty} \sum_{i,j,S} \Pr(X_t = i) \Pr((Y_t,U_t)\in S) \Pr\left(X_{t+1} = j \,|X_t = i,\ (Y_t,U_t)\in S\right) \\ &\qquad \mathbf{E}\left(J(t+1)|X_{t+1} = j\right) \mathbf{E}\left(\frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}} \middle| X_t = i,\ (Y_t,U_t)\in S \right) \\ &= \lim_{k\to\infty} \sum_{i,j,S} \mathbf{E}\left[\chi_i(X_t)\chi_S(Y_t,U_t)\chi_j(X_{t+1}) J(t+1) \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}} \right],\end{aligned}$$

where probabilities and expectations are with respect to the stationary distribution π of X_t, and the distributions on Y_t, U_t. Now, the random process inside the expectation is asymptotically stationary and ergodic. From the ergodic theorem, we have (almost surely)

$$\nabla_\beta \eta = \lim_{k\to\infty} \lim_{T\to\infty} \frac{1}{T} \sum_{i,j,S} \sum_{t=0}^{T-1} \chi_i(X_t)\chi_S(Y_t,U_t)\chi_j(X_{t+1}) J(t+1) \frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}}.$$

It is easy to see that the double limit also exists when the order is reversed, so

$$\begin{aligned}\nabla_\beta \eta &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \lim_{k\to\infty} \sum_{i,j,S} \chi_i(X_t)\chi_S(Y_t,U_t)\chi_j(X_{t+1})J(t+1)\frac{\nabla \frac{d\mu}{d\lambda}}{\frac{d\mu}{d\lambda}} \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla \frac{d\mu(\theta,Y_t)}{d\lambda}(U_t)}{\frac{d\mu(\theta,Y_t)}{d\lambda}(U_t)} J(t+1).\end{aligned}$$

The same argument as in the proof of Theorem 5 shows that the tails of $J(t+1)$ can be ignored when

$$\left| \frac{\nabla \frac{d\mu(\theta,Y_t)}{d\lambda}(U_t)}{\frac{d\mu(\theta,Y_t)}{d\lambda}(U_t)} \right|$$

and $|r(X_t)|$ are uniformly bounded. It follows that $\Delta_T \to \pi' \nabla P J_\beta$ w.p.1, as required. □

7 Conclusion

We have presented a general algorithm (MCG) for computing arbitrarily accurate approximations to the performance gradient of a parameterized Markov chain. The accuracy of the approximation was shown to be controlled by the size of the subdominant eigenvalue ($|\lambda_2|$) of the transition probability matrix of the Markov chain. We showed how the algorithm could be modified to apply to partially observable Markov decision processes controlled by parameterized stochastic policies, with both discrete and continuous control, observation and state spaces. For the finite state case, we proved convergence with probability 1 of both algorithms.

There are many avenues for further research. Continuous time results should follow as extensions of the results presented here. The MCG and POMDPG algorithms can be applied to countably or uncountably infinite state spaces; convergence results are also needed in these cases. In this paper we only prove convergence with probability 1. It should be possible to derive rates of convergence, for example as a function of $|\lambda_2|$.

In the companion paper [5], we present experimental results showing rapid convergence of the estimates generated by POMDPG to the true gradient $\nabla\eta$. We give on-line variants of the algorithms of the present paper, and also variants of gradient ascent that make use of the estimates of $\nabla_\beta \eta$. We present experimental results showing the effectiveness of these algorithms in a variety of problems, including a three-state MDP, a nonlinear physical control problem, and a call-admission problem.

References

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

[2] L. C. Baird. Advantage Updating. Technical Report WL-TR-93-1146, Wright Patterson AFB OH, 1993.

[3] P. L. Bartlett and J. Baxter. Direct Gradient-Based Reinforcement Learning: III. Rapid Mixing and Convergence. In preparation, July 1999.

[4] J. Baxter, A. Tridgell, and L. Weaver. Learning to Play Chess Using Temporal-Differences. *Machine Learning*, 1999. To appear.

[5] J. Baxter, L. Weaver, and P. L. Bartlett. Direct Gradient-Based Reinforcement Learning: II. Gradient Descent Algorithms and Experiments. In preparation, July 1999.

[6] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, 1995.

[7] D. P. Bertsekas. Differential Training of Rollout Policies. In *Proceedings of the 35th Allerton Conference on Communication, Control, and Computing*, Allerton Park, Ill., 1997.

[8] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[9] L. Brieman. *Probability*. Addison-Wesley, 1966.

[10] X.-R. Cao and H.-F. Chen. Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes. *IEEE Transactions on Automatic Control*, 42:1382–1393, 1997.

[11] 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.

[12] F. DeBruyne, B. D. O. Anderson, M. Gevers, and N. Linard. Iterative controller optimization for nonlinear systems. In *Proceedings of the 36rd IEEE Conference on Decision and Control*, pages 3749–3754, 1997.

[13] R. M. Dudley. *Real Analysis and Probability*. Wadsworth & Brooks/Cole, Belmont, California, 1989.

[14] H. Hjalmarsson, S. Gunnarsson, and M. Gevers. A convergent iterative restricted complexity control design scheme. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 1735–1740, 1994.

[15] L. C. Kammer, R. R. Bitmead, and P. L. Bartlett. Direct iterative tuning via spectral analysis. *Automatica*, 1999. to appear.

[16] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag, 1983.

[17] 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.

[18] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. Academic Press, San Diego, CA, 1985 (Second Edition).

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

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

[21] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3:210–229, 1959.

[22] G. E. Shilov and B. L. Gurevich. *Integral, Measure and Derivative: A Unified Approach*. Prentice-Hall, Englewood Cliffs, N.J., 1966.

[23] S. Singh. An Upper Bound on the Loss from Approximate Optimal Value Functions. *Machine Learning*, 16:227–233, 1994.

[24] S. Singh and D. Bertsekas. Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference*, pages 974–980. MIT Press, 1997.

[25] S. Singh, T. Jaakkola, and M. Jordan. Reinforcement learning with soft state aggregation. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, Cambridge, MA, 1995.

[26] R. Sutton. Learning to Predict by the Method of Temporal Differences. *Machine Learning*, 3:9–44, 1988.

[27] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge MA, 1998. ISBN 0-262-19398-1.

[28] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. Technical report, AT&T Labs–Research, 1999.

[29] G. Tesauro. Practical Issues in Temporal Difference Learning. *Machine Learning*, 8:257–278, 1992.

[30] G. Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6:215–219, 1994.

[31] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[32] L. Weaver and J. Baxter. Reinforcement Learning From State Differences. Technical report, Department of Computer Science, Australian National University, May 1999. http://cs.anu.edu.au/∼Lex.Weaver/pub_sem.

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

[34] R. J. Williams and L. Baird. Analysis of Some Incremental Variants of Policy Iteration: First Steps Toward Understanding Actor-Critic Learning Systems. Technical Report NU-CCS-93-11, College of Computer Science, Northeastern University, Boston MA, 1993.

[35] W. Zhang and T. Dietterich. A reinforcement learning approach to job-shop scheduling. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1114–1120. Morgan Kaufmann, 1995.