Importance Sampling Actor-Critic Algorithms

Jason L. Williams John W. Fisher III Alan S. Willsky
Laboratory for Information and Decision Systems and
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
jlwil@mit.edu fisher@csail.mit.edu willsky@mit.edu

***Abstract*— Importance Sampling (IS) and actor-critic are two methods which have been used to reduce the variance of gradient estimates in policy gradient optimization methods. We show how IS can be used with Temporal Difference methods to estimate a cost function parameter for one policy using the entire history of system interactions incorporating many different policies. The resulting algorithm is then applied to improving gradient estimates in a policy gradient optimization. The empirical results demonstrate a 20-40$\times$ reduction in variance over the IS estimator for an example queueing problem, resulting in a similar factor of improvement in convergence for a gradient search.**

I. Introduction

Many problems of practical interest can be formulated and, conceptually, solved optimally using dynamic programming (*cf* [1]). However, the practical applicability of the method to problems with large state spaces is limited due to the so-called *curse of dimensionality*. The absence of exact models for the systems of interest further limits applicability, through the so-called *curse of modelling*. Approximations which address both of these difficulties have been studied extensively over the past decade, and may be divided into two broad categories: cost function approximation methods and policy approximation methods.

Cost function approximation methods seek to approximate the optimal cost-to-go function with a particular parametric form, such as a linear combination of basis functions. The parameter vector can be learned from simulation using the method of Temporal Differences [2]. In policy approximation, one chooses a particular parameterized policy family, and seeks to find the parameter value which minimizes the cost of employing the policy. The minimization is commonly performed using stochastic gradient methods. The policy gradient method [3] obtains a noisy estimate of the gradient of the objective with respect to the policy parameter from a single simulation trajectory. The actor-critic method [4] improves this estimate by incorporating cost function approximation: by retaining information from previous simulations, and constraining the estimates of the cost-to-go function to a low-dimensional subspace, the variance is reduced substantially.

Frequently, interaction with the system (or simulation of the system) is expensive, computationally or otherwise, hence it is desirable to exploit the limited simulation data which is available as much as possible. The Importance Sampling (IS) method of [5], discussed in detail in Section II-B, provides a means of utilizing information from the entire history of interactions with the system (using many different policies) to compute a reduced variance estimate of the objective gradient. We present an algorithm in Section III that combines the IS algorithm with a cost function approximation method. By restricting the cost estimates to lie in a low-dimensional subspace, the variance of the gradient estimate is reduced substantially. The simulation results in Section IV demonstrate a 20-40$\times$ reduction in variance over the IS estimator for an example queueing problem.

Whereas Konda and Tsitsiklis' actor-critic method relies on two time constants, one of which controls the faster adaption rate of the approximation parameter, and the other which controls the slower adaption rate of the policy parameter, our method adaptively weights the entire system interaction history to calculate approximation parameters. This allows the system designer to trade off the number of interactions required with the system against computational cost.

II. Background

Consider a Markov Decision Process (MDP) with states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. We assume a randomized policy parameterized by $\boldsymbol{\theta}$, such that action a is selected in state s with probability $u_{\boldsymbol{\theta}}(s,a)$. If action a is selected in state s, the immediate cost incurred is $g_a(s)$, and the next state is drawn from the transition distribution on $\mathcal{S}$, $p(\cdot|s,a)$.

For our purposes, we consider stochastic shortest path formulations[1] in which, under all policies, there is a single recurrent class of states $\mathcal{T} \subset \mathcal{S}$ and that these states are cost-free ($g_a(s) = 0 \; \forall \; s \in \mathcal{T}$). Thus the goal of the policy is to minimize the cost incurred before reaching the terminal set $\mathcal{T}$. As in [3], we assume that the starting state is drawn from the probability distribution on $\mathcal{S}$, $\bar{\pi}(\cdot)$. We denote by $J_{\boldsymbol{\theta}}(s)$ the expected cost-to-go from state s using the policy defined by the parameter vector $\boldsymbol{\theta}$. Consequently, the objective we wish to minimize is:

$$\chi(\boldsymbol{\theta}) = \int_{\mathcal{S}} \bar{\pi}(s) J_{\boldsymbol{\theta}}(s) \mathrm{d}s \tag{1}$$

We assume that the state is known at each decision step. We make the common assumptions that $\exists \, n < \infty$ such that

[1] Analogous statements can be made for average cost per stage problems by considering renewal intervals, as per the development in [3].

the probability of terminating within n steps from any state and under any policy is uniformly bounded below by $\epsilon > 0$, and that the policy distribution $u_{\boldsymbol{\theta}}(s,a)$ is differentiable w.r.t. $\boldsymbol{\theta}$ and $\frac{\nabla_{\boldsymbol{\theta}} u_{\boldsymbol{\theta}}(s,a)}{u_{\boldsymbol{\theta}}(s,a)}$ is bounded.

A. λ-Least Squares Temporal Difference

λ-Least Squares Temporal Difference (λ-LSTD) provides a method of approximate policy evaluation with convergence at a much faster rate than traditional temporal difference methods [6], [7]. Given a set of basis functions, the cost-to-go is approximated by:

$$\check{J}(s,r) = \boldsymbol{\phi}(s)^T \boldsymbol{r} \tag{2}$$

To apply it to a stochastic shortest path problem, we take a series of independent simulation trajectories and accrue for the i-th simulation, $x^i = (s_0^i, a_0^i, \ldots, s_{n^i-1}^i, a_{n^i-1}^i, s_{n^i}^i)$, the following quantities:

$$\begin{aligned} \mathbf{A}(x^i) &= \sum_{m=0}^{n^i-1} \boldsymbol{z}_m^i [\boldsymbol{\phi}(s_{m+1}^i) - \boldsymbol{\phi}(s_m^i)]^T \\ \boldsymbol{b}(x^i) &= \sum_{m=0}^{n^i-1} \boldsymbol{z}_m^i g(s_m^i) \end{aligned} \tag{3}$$

where $\boldsymbol{z}_m^i = \sum_{k=0}^{m} \lambda^{m-k} \boldsymbol{\phi}(s_k^i)$. We can then combine N simulation trajectories $(x^1, \ldots, x^N)$ to give an overall estimate:

$$\begin{aligned} \mathbf{A}_N &= \frac{1}{N} \sum_{m=1}^{N} \mathbf{A}(x^i) \\ \boldsymbol{b}_N &= \frac{1}{N} \sum_{m=1}^{N} \boldsymbol{b}(x^i) \\ \boldsymbol{r}_N &= -\mathbf{A}_N^{-1} \boldsymbol{b}_N \end{aligned} \tag{4}$$

This is equivalent to the method described in [6], except that the eligibility trace $\boldsymbol{z}^i$ is reset after each simulation (which is natural for a stochastic shortest path problem). Under mild conditions,[2] $\mathbf{A}_N$ and $\boldsymbol{b}_N$ will converge to their expected values as $N \to \infty$ w.p. 1, hence the parameter vector $\boldsymbol{r}_N$ also converges. The choice of basis functions $\phi(s)$ is problem dependent; Section IV provides an example for a queueing problem.

B. Importance Sampling

Policy gradient methods estimate $\nabla_{\boldsymbol{\theta}} \chi(\theta)$ via a small number of Monte Carlo simulations under an essentially fixed policy. The resulting estimate often exhibits high variance, increasing convergence times while decreasing the utility of the resulting policy. Peshkin and Shelton [5] suggest an Importance Sampling (IS) approach, using sample trajectories under *varied* policies, as a means of incorporating previous simulations and, consequently, reducing the variance. The importance sampling estimate of $\mathrm{E}\{f(x)\}$, using samples $\{x^i\}_{i=1}^N$ drawn from distribution $q(\cdot)$, is expressed by:

$$\begin{aligned} \hat{f} &= \frac{1}{N} \sum_{i=1}^{N} \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} f(x^i) \\ \tilde{f} &= \frac{\frac{1}{N} \sum_{i=1}^{N} \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} f(x^i)}{\frac{1}{N} \sum_{i=1}^{N} \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)}} \end{aligned} \tag{5}$$

The estimate $\hat{f}$ is unbiased, provided that $q(x) > 0 \ \forall \ x : p_{\boldsymbol{\theta}}(x) > 0$, and converges to the true value as the number of samples increases. The normalized estimate $\tilde{f}$ reduces the variance in $\hat{f}$ at the expense of inducing a bias (which vanishes asymptotically). Samples corresponding to many different parameter values $\{\boldsymbol{\theta}_i, \ i \in \{1, \ldots, N\}\}$ may be admitted by treating them as collectively belonging to the mixture density $q(x) = \frac{1}{N} \sum_{i=1}^{N} p_{\boldsymbol{\theta}_i}(x)$, where $p_{\boldsymbol{\theta}_i}(x)$ is the likelihood of trajectory x under the policy defined by the i-th parameter $\boldsymbol{\theta}_i$. In a stochastic shortest path problem, each simulation trajectory forms an independent sample, starting at an independently drawn initial state $s_0 \sim \bar{\pi}(\cdot)$, and ending with $s_n \in \mathcal{T}$. The likelihood of obtaining a given trajectory $x = (s_0, a_0, \ldots, s_{n-1}, a_{n-1}, s_n)$ may be calculated as:

$$p_{\boldsymbol{\theta}}(x) = \left[p(s_0) \prod_{i=1}^{n} p(s_i | s_{i-1}, a_{i-1}) \right] \cdot \left[\prod_{i=0}^{n-1} u_{\boldsymbol{\theta}}(s_i, a_i) \right] \tag{6}$$

where only the second term in brackets has dependence on the policy parameter vector $\boldsymbol{\theta}$. Consequently, in calculating the ratios $\frac{p(\cdot)}{q(\cdot)}$ to estimate the cost at parameter value $\boldsymbol{\theta}$, the transition probabilities cancel leaving:

$$\frac{p_{\boldsymbol{\theta}}(x)}{q(x)} = \frac{\prod_{i=0}^{n-1} u_{\boldsymbol{\theta}}(s_i, a_i)}{\frac{1}{N} \sum_{j=0}^{N} \prod_{i=0}^{n-1} u_{\boldsymbol{\theta}_j}(s_i, a_i)} \tag{7}$$

In order to estimate the gradient of the objective, one can use a simple estimate of the cost along a trajectory, $\hat{J}(x) = \sum_{k=0}^{n} g_{a_k}(s_k)$. Noting that $\mathrm{E}\{\hat{J}(x) | s_0 \sim \bar{\pi}(\cdot)\} = \chi(\boldsymbol{\theta})$, we then have: [3]

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \chi(\boldsymbol{\theta}) &= \int \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x) \hat{J}(x) \mathrm{d}x \\ &= \int p_{\boldsymbol{\theta}}(x) \frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x)}{p_{\boldsymbol{\theta}}(x)} \hat{J}(x) \mathrm{d}x \end{aligned} \tag{8}$$

Assuming that $\frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x)}{p_{\boldsymbol{\theta}}(x)}$ is bounded, we apply Eq. (5) to obtain

$$\widehat{\nabla_{\boldsymbol{\theta}} \chi}(\boldsymbol{\theta}, \{x^i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} \frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \hat{J}(x^i) \tag{9}$$

[2] The trajectories $\{x^i\}$ are i.i.d., the per-stage costs are bounded and the usual termination assumption of stochastic shortest path problems are met (i.e. $\mathbf{A}(x^i)$ and $\boldsymbol{b}(x^i)$ have finite variance).

[3] Note that the random variable to which the variable of integration in Eq. (8) corresponds may be discrete or mixed, depending on the state and action spaces. Furthermore, it is of variable dimension, depending on the length of the trajectory. For the sake of clarity we use normal Riemann integration notation; the ideas may be made precise using measure theoretic notation, replacing the importance sampling weights with the Radon-Nikodym derivatives.

We estimate the corresponding cost similarly:

$$\hat{\chi}(\boldsymbol{\theta}, \{x^i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^N \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \hat{J}(x^i) \tag{10}$$

III. Importance Sampling Actor-Critic

The method described in the previous section applies IS to a simple trajectory-based cost estimator $\hat{J}(x)$ to estimate the objective for different parameter values. The algorithm developed below, referred to as Importance Sampling Actor-Critic (ISAC), uses a cost function approximation method to constrain these estimates to a low-dimensional subspace, reducing the variance of the resulting estimates.

While the recursive forms of the TD(λ) method [2] and λ-LSPE [6] algorithms are convenient for online processing, they do not allow IS to be applied to evaluate the cost of one policy using simulations from other policies. However, the structure of the λ-LSTD estimator does provide a form which allows the quantities being accrued to be broken up into a mean of independent samples, so that IS can be applied. We outline how this can be done in Section III-A, and apply the result to policy gradient estimation in Section III-B.

A. Importance Sampling Least Squares Temporal Difference

Suppose we want to use λ-LSTD to estimate the cost of the policy resulting from parameter $\boldsymbol{\theta}$. Eq. (4) provides a means of taking a sequence of N i.i.d. trajectories and estimating the quantities $\mathbf{A}_{\boldsymbol{\theta}} = \mathrm{E}\{\mathbf{A}(x)\}$ and $\boldsymbol{b}_{\boldsymbol{\theta}} = \mathrm{E}\{\boldsymbol{b}(x)\}$, and hence the cost function approximation parameter $\boldsymbol{r}_{\boldsymbol{\theta}} = \mathbf{A}_{\boldsymbol{\theta}}^{-1} \boldsymbol{b}_{\boldsymbol{\theta}}$. Using Eq. (5), we may also obtain unbiased estimates of these quantities, using simulations from different policies through importance sampling:

$$\begin{aligned}
\hat{\mathbf{A}}_{\boldsymbol{\theta}}(\{x^i\}_{i=1}^N) &= \frac{1}{N} \sum_{i=1}^N \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \mathbf{A}(x^i) \\
\hat{\boldsymbol{b}}_{\boldsymbol{\theta}}(\{x^i\}_{i=1}^N) &= \frac{1}{N} \sum_{i=1}^N \frac{p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \boldsymbol{b}(x^i)
\end{aligned} \tag{11}$$

Since the probability weights are the only variables which depend on the parameter vector, we can estimate the gradients of these quantities as:[4]

$$\begin{aligned}
\widehat{\nabla_{\boldsymbol{\theta}} \mathbf{A}}_{\boldsymbol{\theta}}(\{x^i\}_{i=1}^N) &= \frac{1}{N} \sum_{i=1}^N \frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \mathbf{A}(x^i) \\
\widehat{\nabla_{\boldsymbol{\theta}} \boldsymbol{b}}_{\boldsymbol{\theta}}(\{x^i\}_{i=1}^N) &= \frac{1}{N} \sum_{i=1}^N \frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x^i)}{q(x^i)} \boldsymbol{b}(x^i)
\end{aligned} \tag{12}$$

B. Application to Actor-Critic

If we use an estimate of the expected cost of a simulation $\tilde{J}(\boldsymbol{\theta}, x)$, which depends on the value of the policy parameter, the gradient of the expected cost in Eq. (8) becomes

$$\nabla_{\boldsymbol{\theta}} \chi(\boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(x) \tilde{J}(\boldsymbol{\theta}, x) \mathrm{d}x + \int p_{\boldsymbol{\theta}}(x) \nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}, x) \mathrm{d}x \tag{13}$$

[4] In general, the gradient of the matrix $\hat{\mathbf{A}}_{\boldsymbol{\theta}}$ w.r.t. the vector $\boldsymbol{\theta}$ is a tensor; we will treat it as a collection of matrices, each member of which is the derivative of $\hat{\mathbf{A}}_{\boldsymbol{\theta}}$ with respect to a different component of $\boldsymbol{\theta}$.

Noting that if $\tilde{J}$ is not a function $\boldsymbol{\theta}$, then the second term in Eq. (13) is zero, we see that Eq. (13) is a generalization of Eq. (8). Now consider an alternative form of cost estimator, based on the λ-LSTD algorithm. The cost of a simulation trajectory $x = (s_0, a_0, \ldots, s_{n-1}, a_{n-1}, s_n)$ is estimated as:

$$\tilde{J}(\boldsymbol{\theta}, x) = \phi(s_0)^T \boldsymbol{r}_{\boldsymbol{\theta}} \tag{14}$$

where from Eq. (11) $\boldsymbol{r}_{\boldsymbol{\theta}} = -\mathbf{A}_{\boldsymbol{\theta}}^{-1} \boldsymbol{b}_{\boldsymbol{\theta}}$. Using this estimator, only the starting state of the trajectory is used for the cost estimate: the impact of the policy parameter on the cost is taken into account through the second term in Eq. (13). Since the cost estimate depends only on the starting state, Eq. (13) may be rewritten as:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \chi(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \int \bar{\pi}(s_0) \tilde{J}(\boldsymbol{\theta}, s_0) \mathrm{d}s_0 \\
&= \int \bar{\pi}(s_0) \nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}, s_0) \mathrm{d}s_0
\end{aligned} \tag{15}$$

Because the distribution of starting state is not a function of the policy parameter, the first term in Eq. (13) is zero. Evaluation of Eq. (15) requires the gradient of $\tilde{J}(\boldsymbol{\theta}, s_0)$:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}, s_0) &= \phi(s_0)^T \nabla_{\boldsymbol{\theta}} \boldsymbol{r}_{\boldsymbol{\theta}} \\
&= -\phi(s_0)^T \nabla_{\boldsymbol{\theta}} [\mathbf{A}_{\boldsymbol{\theta}}^{-1} \boldsymbol{b}_{\boldsymbol{\theta}}] \qquad (16) \\
&= \phi(s_0)^T \{\mathbf{A}_{\boldsymbol{\theta}}^{-1} [\nabla_{\boldsymbol{\theta}} \mathbf{A}_{\boldsymbol{\theta}}] \mathbf{A}_{\boldsymbol{\theta}}^{-1} \boldsymbol{b}_{\boldsymbol{\theta}} - \mathbf{A}_{\boldsymbol{\theta}}^{-1} \nabla_{\boldsymbol{\theta}} \boldsymbol{b}_{\boldsymbol{\theta}}\} \qquad (17)
\end{aligned}$$

Eq. (17) should be read as meaning that the i-th element of $\nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{\theta}, s_0)$ is the RHS evaluated with gradients taken w.r.t. the i-th term of $\boldsymbol{\theta}$. The gradient estimate can then be evaluated as: (where all estimates are functions of the sample $\{x^i\}_{i=1}^N$)

$$\begin{aligned}
\widehat{\nabla_{\boldsymbol{\theta}} \chi}(\boldsymbol{\theta}, \{x^i\}_{i=1}^N) = \left[\frac{1}{N} \sum_{i=1}^N \phi(s_0^i)^T \right] \cdot \\
\left\{ \hat{\mathbf{A}}_{\boldsymbol{\theta}}^{-1} \widehat{\nabla_{\boldsymbol{\theta}} \mathbf{A}}_{\boldsymbol{\theta}} \hat{\mathbf{A}}_{\boldsymbol{\theta}}^{-1} \hat{\boldsymbol{b}}_{\boldsymbol{\theta}} - \hat{\mathbf{A}}_{\boldsymbol{\theta}}^{-1} \widehat{\nabla_{\boldsymbol{\theta}} \boldsymbol{b}}_{\boldsymbol{\theta}} \right\}
\end{aligned} \tag{18}$$

While the individual estimates for $\hat{\mathbf{A}}_{\boldsymbol{\theta}}$, $\widehat{\nabla_{\boldsymbol{\theta}} \mathbf{A}}_{\boldsymbol{\theta}}$, $\hat{\boldsymbol{b}}_{\boldsymbol{\theta}}$ and $\widehat{\nabla_{\boldsymbol{\theta}} \boldsymbol{b}}_{\boldsymbol{\theta}}$ are unbiased, the nonlinear composition in Eq. (18) will be biased. However, as the number of samples N grows, the individual estimates converge to the respective true parameter values, hence the gradient estimate will be asymptotically unbiased. The cost itself can be estimated similarly:

$$\hat{\chi}(\boldsymbol{\theta}, \{x^i\}_{i=1}^N) = \left[\frac{1}{N} \sum_{i=1}^N \phi(s_0^i)^T \right] \hat{\mathbf{A}}_{\boldsymbol{\theta}}^{-1} \hat{\boldsymbol{b}}_{\boldsymbol{\theta}} \tag{19}$$

C. Remarks

The form of the actor-critic algorithm of [4] provides insight into a subspace which the basis functions must span for the purpose of obtaining the gradient estimate. An analogous insight for the ISAC estimator is not obvious, and remains an open question. However, the empirical results in the following section demonstrate the dramatic improvement in convergence which can be achieved using a well-chosen low-dimensional cost estimate, as well as the bias which can result from a poor choice of basis functions.

The approximation architecture is primarily employed in ISAC to estimate the cost from the starting state to the end of the simulation. Therefore, it is intuitively desirable to select the approximation parameter which minimizes the difference between the true cost and the approximate cost according to a norm weighted by the distribution of starting states. Convergence bounds for temporal difference algorithms [2] (including λ-LSTD) are in terms of the L_2 norm weighted by the steady state distribution. Accordingly, if the distribution of starting state is vastly different from the steady state distribution, then the error in the resulting cost estimate may be large.

A practical issue which affects the IS and ISAC algorithms alike is computational complexity, due to the growing history of system interactions over which the estimates are calculated. In practice, one would commonly retain a subset of past interactions; in the experiments to follow, we retain the most recent 10,000 simulations. Selection of this memory length effectively allows the system designer to trade off the number of interactions required with the system against computational cost.

IV. Experimental Results

We compare the performance of the algorithm presented in Section III was compared to the IS estimator discussed in Section II-B for a queueing problem. So that we might compare performance to the optimal solution, the problem was chosen to have a small enough size such that the actual cost-to-go function can be calculated. The state in the problem corresponds to the length of a queue, $s \in \{0, \ldots, B\}$, where $B = 100$. The control $a \in \{0, 1\}$ affects the probability that the queue length is reduced:

$$P(s_k = y | s_{k-1} = x, a_{k-1} = a) = \begin{cases} 1, & y = x = 0 \\ 1 - \beta_a, & y = x > 0 \\ \beta_a, & y = x - 1 \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

where $\beta_0 = 0.2$, and $\beta_1 = 0.5$. The initial state is drawn from a modified geometric distribution:

$$P(s_0 = x) = \begin{cases} 0, & x = 0 \\ (1-\omega)^{x-1}\omega, & 1 \leq x < B \\ \sum_{y=B}^{\infty} (1-\omega)^{y-1}\omega, & x = B \end{cases} \quad (21)$$

where $\omega = 0.03$. The cost per stage is given by $g_a(s) = s + a\eta$, where $\eta = 35$. The stochastic shortest path problem terminates when the queue is emptied ($s = 0$). A reasonable parameterized policy family for this problem is a soft threshold function:

$$u_\theta(s, a = 1) = \frac{\exp[0.1(s - \theta)]}{1 + \exp[0.1(s - \theta)]} \quad (22)$$

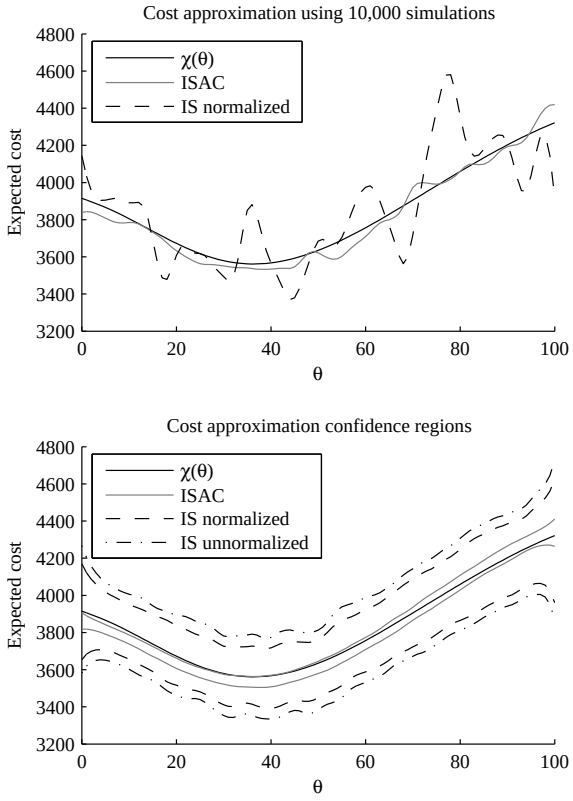

Fig. 1. Expected cost from starting state to termination as a function of θ. Upper figure shows the true cost, and costs estimated from a single set of 10,000 simulations using the ISAC and normalized IS estimators. Lower plot shows the 1-σ confidence bounds for estimates using 10,000 simulations, estimated using 400 sets of 10,000 simulations.

For any given θ, the true cost of the policy can be found by solving Bellman's equation: [1]

$$J_\theta(s) = \sum_a u_\theta(s, a) g_a(s) + \sum_{a,y} u_\theta(s, a) P(y|s, a) J_\theta(y), \quad s \in \{1, \ldots, B\} \quad (23)$$

where $J_\theta(0) = 0$. The value $J_\theta(s)$ is the expected cost to reach termination from state s using the policy defined by the parameter θ. From Eq. (1), the quantity which we seek to minimize is the expected cost to reach termination when the starting state is drawn from the initial distribution in Eq. (21), $\chi(\theta) = \sum_s P(s_0 = s) J_\theta(s)$. We use the value $\lambda = 1$ in the LSTD algorithm, and feature vectors $\phi(s) = [s\ s^2\ s^3]^T$.

A. Cost function estimates

In order to give a qualitative comparison of the relative merit of the two approximations, the cost estimates of Eq. (10) and Eq. (19) were evaluated for a range of values of θ with simulations computed using random policy parameter values, $\theta \sim U(0, 100)$. The upper plot in Fig. 1 compares the true cost to the cost estimates obtained using the two methods with a single set of 10,000 simulations. The gradient estimators of Eq. (9) and Eq. (18) correspond to calculating the derivative of the respective curves in Fig. 1,

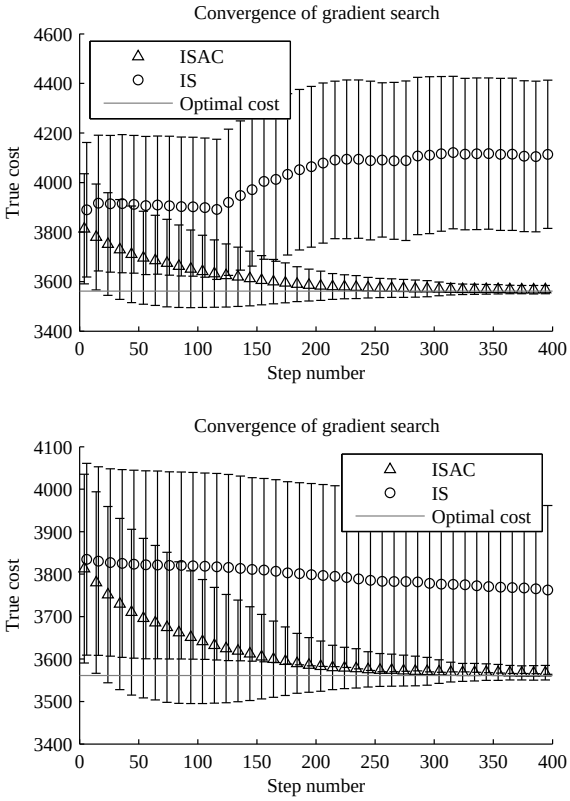

Fig. 2. Convergence of gradient search algorithms. Upper plot shows mean and standard deviation of the true cost achieved after different numbers of gradient steps using the ISAC and IS gradient estimates with step size $\tau = 0.05$. Lower plot shows the same data for ISAC alongside results using IS gradient estimate with step size $\tau = 0.0025$. Some lower error bounds fall below the optimal cost due to skewness in the distributions.

hence the diagram allows one to anticipate the behavior of a gradient optimization procedure to some degree. The large variability of the IS estimate demonstrates that gradient estimates combining this number of simulations will still have a large variance. While there are still fluctuations in the ISAC estimate (and some local minima), their variance is reduced substantially. The lower plot in Fig. 1 shows the 1-σ confidence bounds for the estimates obtained from 10,000 simulations (the confidence bounds were estimated using 400 different sets of 10,000 simulations). The reduction in the standard deviation of the ISAC estimator over the normalized IS estimator is a factor of between four and six, corresponding to a reduction in variance by a factor of between 20 and 40.

B. Gradient search

To compare the performance of the gradient search procedures, we tested the two algorithms from the same starting point for 150 random values of $\theta_0 \sim U(0, 100)$. In each gradient iteration, we performed 100 Monte Carlo simulations using the current policy parameter θ_k, calculated the gradient using the respective estimate (Eq. (9) and Eq. (18)), and implemented the gradient step $\theta_{k+1} = \theta_k - \tau\widehat{\nabla\chi}$. Gradient estimates were calculated using a sliding window of the

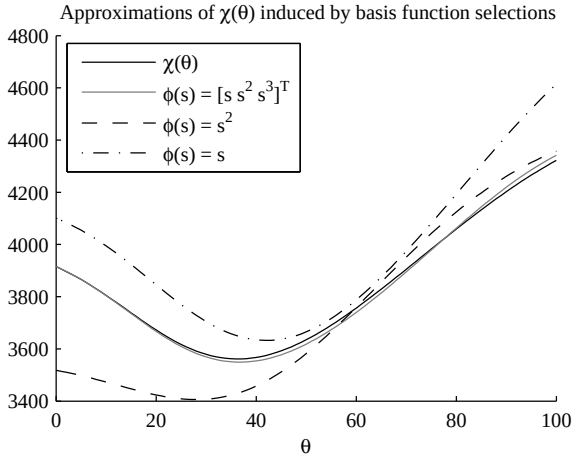

Fig. 3. Approximation of objective function resulting from different choices of basis functions.

last 10,000 simulations. The behavior of the algorithms is illustrated in Fig. 2. The upper plot shows the mean and standard deviation of the true costs of the policies after different numbers of gradient steps, with step size $\tau = 0.05$. The plot illustrates that the ISAC method consistently converges to the optimal solution, while the IS method oscillates between apparent local minima. The lower plot shows the same data for ISAC alongside results using IS gradient estimate with step size $\tau = 0.0025$. With the smaller step size, the IS estimator exhibits slow convergence towards the minimum, but a far greater number of steps is required to achieve the same degree of convergence as ISAC.

C. Impact of approximation architecture

The ISAC method discussed in Section III-B effectively optimizes the function $\check{\chi}(\theta) = \mathrm{E}\{\phi(s_0)^T\}\boldsymbol{r}_\theta$, where $\boldsymbol{r}_\theta$ is the approximation architecture parameter vector supplied by the λ-LSTD algorithm. If $\lambda = 1$, the LSTD algorithm will converge to the weighted projection of the true cost function onto the approximation architecture subspace. The approximations $\check{\chi}(\theta)$ resulting from these projections are shown for different selections of approximation architecture in Fig. 3, demonstrating the importance of choosing an approximation architecture which provides sensitivity to variations in the cost due to the changing parameter vector. In this problem, a quadratic cost approximation visually appears to provide a good fit to the true cost (*i.e.*, $J_\theta(s)$ is well-approximated by $r \cdot s^2$ for most policies), yet the resulting approximation of $\chi(\theta)$ loses much of the true structure. Comparatively, a linear cost approximation visually seems poorer, but the resulting approximation is reasonable.

V. Conclusions

This paper has shown how importance sampling can be applied to estimate the parameter of a cost function approximation architecture using λ-LSTD, and how the resulting algorithm can be applied to improve gradient estimates in a policy gradient optimization by restricting cost estimates to a

low-dimensional subspace. Our empirical results demonstrate the utility of the proposed method in several ways: analysis of the cost function showed that the proposed method results in a significantly better approximation, while gradient implementations were able to use much larger step sizes, resulting in significantly faster convergence behavior.

REFERENCES

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.

[2] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, May 1997.

[3] P. Marbach, "Simulation-based methods for markov decision processes," PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[4] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms." *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.

[5] L. Peshkin and C. R. Shelton, "Learning from scarce experience," in *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 498–505.

[6] A. Nedic and D. P. Bertsekas, "Least squares policy evaluation algorithms with linear function approximation," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 13, pp. 79–110, 2003.

[7] D. Bertsekas, V. Borkar, and A. Nedic, "Improved temporal difference methods with linear function approximation," Massachusetts Institute of Technology, Tech. Rep. LIDS-TR-2573, 2003.