An Actor critic Algorithm Based on Grassmanian Search

†Prabuchandran K.J., †Shalabh Bhatnagar, *Vivek S. Borkar

IISc-CSA-SSL-TR-2014-1


Stochastic Systems Lab
Computer Science and Automation
Indian Institute of Science, India
January 2014
An Actor critic Algorithm Based on Grassmanian Search

†Prabuchandran K.J., †Shalabh Bhatnagar, *Vivek S. Borkar

Technical Report

Abstract

We propose an online actor-critic scheme with adaptive basis to find a local optimal control policy for a Markov Decision Process (MDP) under the weighted discounted cost objective. We parametrize both the policy in the actor and the value function in the critic. The actor performs gradient search in the space of policy parameters using simultaneous perturbation stochastic approximation (SPSA) gradient estimates. This gradient computation requires estimates of value function that are provided by the critic by minimizing a mean square Bellman error objective. In order to obtain good estimates of the value function, the critic adaptively tunes the basis functions (or the features) to obtain the best representation of the value function using gradient search in the Grassmanian of features. Our control algorithm makes use of multi-timescale stochastic approximation. The actor updates its parameters along the slowest time scale. The critic uses two time scales to estimate the value function. For any given feature value, our algorithm performs gradient search in the parameter space via a residual gradient scheme on the faster timescale and, on a medium timescale, performs gradient search in the Grassman manifold of features. We prove the critic iterates converge to the value function corresponding to the current actor parameter. We also provide an outline of the proof of convergence of our control algorithm to a locally optimum policy.

Keywords: Control, feature adaptation, online learning, residual gradient scheme, temporal difference learning, stochastic approximation, Grassman manifold.

†Department of Computer Science and Automation, Indian Institute of Science, Bangalore.
*Department of Electrical Engineering, Indian Institute of Technology, Powai.
E-mail: {prabu.kj,shalabh}@csa.iisc.ernet.in, borkar.vs@gmail.com.
1 Introduction

In sequential decision making problems one essentially faces the problem of optimally deciding under various situations (or states) at different stages in time. Markov Decision Process (MDP) offers a mathematical framework for studying such sequential decision making problems under uncertainty. The objective in an MDP setting is to choose a sequence of actions so as to minimize the long-term cost incurred. Based on the nature of the application, one either minimizes long-term discounted cost or average cost. In our work, we develop an algorithm for solving MDP to minimize the weighted long-term discounted cost objective. We note that similar objectives have been considered in [1], [2] and chapter 6.10 of [3].

Reinforcement learning (RL) methods are model-free methods to solve MDP. The use of function approximation with RL algorithms makes it a powerful tool for solving large MDPs. Function approximation may be carried out by parameterizing the value function (critic-only methods [4]) or policy (actor-only methods [5]) or both (actor-critic methods)[[6],[7]]. The actor-critic setup offers several advantages over actor-only and critic-only methods [6].

In our work we use the actor-critic architecture that parameterizes both the value function and the policy. The critic uses an approximation architecture using features and uses simulation to learn the value function of the policy for the given actor (policy) parameter $\theta$. The actor’s policy parameter $\theta$ is updated in the direction of improving the performance metric (in our case the performance metric is the weighted long-run discounted cost of a policy denoted by $\rho(\theta)$). The direction of improvement is found by changing the policy parameters along the negative gradient direction [6],[7]. We have convergence guarantees for such schemes whenever the value function is well approximated. The error in approximation will depend on the choice of the features used to approximate the value function. In many algorithms the features are fixed a priori, and as a result the approximation may be poor. Hence, the policy obtained using such features in critic may perform poorly. To overcome this problem, we propose to adaptively tune these features in addition, so as to obtain the best features in an online scheme. [8] develops a policy evaluation algorithm that incorporates adaptive feature tuning to estimate the value function for a discounted cost MDP for a given stationary deterministic policy (SDP). In our current work, we first extend the algorithm to estimate the value function of a stationary randomized policy (SRP) in the discounted cost MDP framework and then use this estimate to develop a full RL control algorithm using SPSA [9].

Various feature adaptation based methods to approximate value function have been studied in the literature. In [1], radial basis functions (RBF) with parameterization are considered as the feature vectors. The parameters of RBF are then tuned using two methods, namely, gradient descent which converges to the local optimum and the cross entropy method which converges to global optimum. A general framework for studying adaptive basis as an extension of [1] is presented in [10]. An automatic basis function construction has been proposed in [11] where the
high-dimensional state space is mapped to a low-dimensional space using neighborhood component analysis and state aggregation is used in the lower dimensional space to construct the basis functions. In [12], a non-parameterized adaptive scheme for basis selection is proposed in conjunction with TD.

All the methods discussed above have been developed for approximating the value function of a given policy for the discounted cost MDP. Further, no extensions of these methods in the context of control (or policy improvement) with adaptive basis have been studied in the literature. The problem of control with adaptive bases for the average cost setting is considered for the first time in [13] and actor-critic algorithms are developed. The natural actor critic algorithm proposed in [7] is extended in [13] by incorporating an adaptive basis selection. The basis functions are parameterized and their parameters are updated using the method given in [11]. [13] utilizes the policy gradient theorem to update the parameters of the actor. But, in the discounted cost framework such updates are difficult to carry out and have not been studied in the literature as such. So, in our algorithm we resort to SPSA based gradient estimates to update the policy parameters.

In this paper, we present an online actor-critic control algorithm for the weighted discounted cost MDP that incorporates basis feature tuning for approximating the value function. The feature search is performed using gradient descent on the Grassman manifold of features. Our algorithm is considerably different from many other feature adaptation algorithms as ours is a control algorithm whereas the other algorithms in most of the prior work are policy evaluation schemes. Also, ours is the first control algorithm that considers adaptive bases for the discounted cost formulation by utilizing gradient search on the Grassmanian for the critic parameters and gradient search on the space of policies for actor parameters. We provide a proof of convergence of our algorithm to a locally optimum policy. Our algorithm exhibits good empirical performance on settings involving randomly generated MDPs. We also show the results of experiments using a similar algorithm with TD run on the faster timescale in place of the residual gradient scheme. Whereas TD is not known to optimize the MSBE error objective, it requires less computation and is nevertheless seen to result in good performance.

The rest of the paper is organized as follows: In Section 2, we discuss the problem setting of MDP and function approximation. In Section 3, we describe a key result related to the function gradient on the Grassman manifold from [14]. In Section 4, we characterize the minimas of our objective function. In Section 5, we present our actor-critic control algorithm. An outline of the proof of convergence using the ordinary differential equation (ODE) technique is presented in Section 6. The detailed convergence analysis will be made available in a journal version of this paper. Results of numerical experiments using our algorithm and TD are then presented in Section 7. Finally, we present our concluding remarks and discuss future work in Section 8.
2 The Framework and Preliminaries

We consider an MDP with finite states and actions. Let $S$ and $A$ respectively denote the state and action spaces of the MDP. We assume $S = \{1, 2, \ldots, N\}$ where $|S| = N$ is a finite number. For simplicity we assume that all actions in $A$ are feasible in every state. The state transitions in the MDP are driven by the probability function $p : S \times S \times A \to \mathcal{R}$, where $p(i, j, a), i, j \in S, a \in A$ gives the probability of moving to the next state $j$ from the current state $i$ under the current action $a$. The cost function is a mapping $k : S \times A \to \mathcal{R}$, where $k(i, a), i \in S, a \in A$ denotes the single-stage cost when the state is $i$ and action $a$ is chosen.

The actions being chosen are often viewed in the form of a given policy, i.e., a rule or method for selecting actions. A deterministic policy $\bar{\pi}$ is a sequence of maps $\bar{\pi} \triangleq \{\mu_0, \mu_1, \ldots\}$ with $\mu_j : S \to A, j \geq 0$. If the policy can be represented via a single map, i.e., $\mu_j \equiv \mu, \forall j \geq 0$, where $\mu$ does not depend on $j$, we call $\bar{\pi}$ or by abuse of notation $\mu$ itself, a stationary deterministic policy (SDP). A stationary randomized policy (SRP) $\pi$ is a mapping that assigns for each state $i \in S$ a probability distribution over $A$. In our work, we consider SRPs $\{\pi_\theta, \theta \in \mathcal{R}^L\}$, where we parametrize the policy $\pi$ using a parameter $\theta$. For each pair $(i, a) \in S \times A$, $\pi_\theta(i, a)$ denotes the probability of choosing action $a$ when the current state is $i$. With a slight abuse of notation, we will interchangeably use SRP $\theta$ for SRP $\pi_\theta$ corresponding to $\theta$. Note that under any SRP the state sequence $\{X_n\}$ and the sequence of state-action pair $\{(X_n, Z_n)\}$ of the MDP form a Markov chain with state-spaces $S$ and $S \times A$ respectively. We make the following assumptions about the policy parameterization.

Assumption 1 Under any SRP $\theta \in \mathcal{R}^L$, the Markov chains $\{X_n\}$ and $\{(X_n, Z_n)\}$ resulting from the MDP are aperiodic and irreducible.

Assumption 2 For any state-action pair $(i, a)$, $\pi_\theta(i, a)$ is continuously differentiable in the parameter $\theta$.

These assumptions are natural and essential in a actor-critic architecture (see [6], [7]). A commonly used parametrization that we also use in our experiments satisfies Assumption 2 is the Gibbs distribution, i.e., $\pi_\theta(i, a) = \frac{\exp(\theta^T \sigma(i, a))}{\sum_a \exp(\theta^T \sigma(i, a))}$, where $\sigma(i, a) \in \mathcal{R}^L$ corresponds to the policy feature for the state-action pairs.

Our goal is to find a SRP $\theta^*$ that minimizes the weighted long-run discounted cost criterion. The weighted discounted cost $\rho(\theta)$ of an SRP $\theta$ with the given weights $\beta(l), l \in \{1, 2, \ldots, S\}$ is given by

$$\rho(\theta) = \sum_{l=1}^{N} \beta(l)V^\theta(l),$$

(1)
where $V^\theta$ corresponds to the state value function for a given SRP $\theta$ and is defined for all $i \in S$ by

$$V^\theta(i) = \sum_{n=0}^{\infty} \mathbb{E}[\gamma^n k(X_n, Z_n) | X_0 = i, \theta],$$

(2)

where $\gamma \in (0, 1)$ is the given discount factor of the MDP. The state value function $V^\theta$ gives the expected long-term discounted single stage cost incurred by following the SRP $\theta$.

We achieve the goal of minimizing $\rho(\theta)$ by performing gradient descent in the parameter space of $\theta$. In our algorithm, we update the actor parameter $\theta$ along the negative gradient direction of $\rho(\theta)$, using SPSA gradient estimates according to (20) in Section 5. To determine the gradient at the policy parameter $\theta$, the SPSA scheme requires two simulations to estimate the objective $\rho(\theta + \epsilon \Delta)$ and $\rho(\theta - \epsilon \Delta)$ where $\epsilon$ is a small perturbation parameter and $\Delta$ is a zero mean Bernoulli random vector whose components $\Delta(l), l \in \{1, 2, \ldots, L\}$ are independent Bernoulli random variables taking values $\pm 1$ with probability $\frac{1}{2}$. In our algorithm, we update the parameter $\theta$ after every $2M$ iterations. In the first $M$ iterations, we set the policy parameter at $\theta + \epsilon \Delta$ and estimate the objective $\rho(\theta + \epsilon \Delta)$. In the subsequent $M$ iterations, we set the policy parameter to $\theta - \epsilon \Delta$ and estimate $\rho(\theta - \epsilon \Delta)$. These two simulations could also be run in parallel as the simulations are independent of one another. Using these two estimates we update the $\theta$ parameter according to (20) at the end of $2M$ iterations. This update rule under mild conditions on step sizes $c(n), n \geq 0$ converges to a parameter $\theta^*$ such that $\pi_{\theta^*}$ is a locally optimum policy.

Now as discussed above, to follow the update rule in (20), the actor needs an estimate of the state value function. This will be obtained in our algorithm by the critic through the Grassmanian gradient search. The critic solves the problem of prediction by estimating the value function of each state under a given SRP $\theta$. The state value function $V^\theta(i), i \in S$ satisfies the Bellman equation.

$$V^\theta(i) = \sum_{a \in A} \pi_{\theta}(i, a)[k(i, a) + \gamma \sum_{j \in S} p(i, j, a)V^\theta(j)], i \in S.$$

(3)

Alternatively, in vector-matrix notation, the same can be written as

$$V^\theta = k^\theta + \gamma P^\theta V^\theta,$$

(4)

where $P^\theta$ is the transition probability matrix of $\{X_n\}$ under SRP $\theta$ whose $(i, j)$th component $P^\theta(i, j) = \sum_{a \in A} p_{i,j}(a)\pi_{\theta}(i, a)$ and $k^\theta = (\sum_{a \in A} \pi_{\theta}(i, a)k(i, a), i \in S)^T$ is the vector of single-stage costs.

To solve the system of equations in (4), one needs the matrix $P^\theta$ and the cost vector $k^\theta$ explicitly. Then, one can utilize a method such as value iteration to solve (4) for $V^\theta$. In practice,
\( P^\theta \) is often not explicitly known and may need to be estimated in order to numerically solve (4). Again, estimating the transition probabilities \( P^\theta(i, j) \) for all states \( i, j \in S \) would also be a computationally infeasible task. A common workaround is to use an approximation architecture for the value function and combine the same with stochastic approximation.

We thus use a linear approximation architecture to approximate \( V^\theta(i) \approx \phi_i^T r \), where \( \phi_i = (\phi_i(1), \ldots, \phi_i(K))^T \) is a \( K \)-dimensional feature vector associated with state \( i \). The parameter vectors \( r = (r_1, \ldots, r_K)^T \) weigh the various feature components. The linear approximation architecture is simple to use and has convergence guarantees when combined with temporal difference learning algorithms [15]. Let \( \Phi \) denote the \( N \times K \) feature matrix with \( \phi_i \), \( i \in S \), as its rows. Thus \( \Phi = [\phi_i(k)]_{i \in S, k=1,...,K} \). Let \( \phi(k) \triangleq (\phi(k), i \in S)^T \) denote the \( k \)th column of \( \Phi \), \( k \in \{1, \ldots, K\} \) having dimension \( N \). From the foregoing, the \( (j,k) \)th element of \( \Phi \) corresponds to \( \phi_j(k) \). We now make the following assumption.

**Assumption 3** The \( K \) columns of the matrix \( \Phi \), i.e., \( \phi(1), \ldots, \phi(K) \) are linearly independent. Further, \( K \leq N \).

From Assumption 3, \( \Phi \) has full column rank. Further, in most real-life applications, the value of \( K \) is typically much less than \( N \). Let \( d^\theta(i) \) be the stationary probability of \( \{X_n\} \) (under SRP \( \theta \)) being in state \( i \in S \). Also, let \( D^\theta \) be a diagonal matrix with entries \( d^\theta(i), i \in S \) along the diagonal. Let the norm \( \| \cdot \|_{D^\theta} \) be defined according to \( \| z \|_{D^\theta} = \sqrt{z^T D^\theta z}, \) where \( z \in \mathbb{R}^N \).

There are various objective functions to measure the approximation error due to function approximation. We shall use the mean square Bellman error (MSBE) objective that is defined by \( G^\theta(\Phi, r) = \| \Phi r - (k^\theta + \gamma P^\theta \Phi r) \|_{D^\theta}^2 \), with \( r \in \mathbb{R}^K \) and the aim is to find a parameter \( r^*_\theta, \Phi \in \mathbb{R}^K \) that minimizes \( G^\theta(\Phi, r) \) over all \( r \). It is assumed many times that the matrix \( \Phi \) is fixed or given a priori that may lead to poor approximation of the value function. In section 5 we present the scheme ((17), (18), (19)) to tune the feature matrix \( \Phi \) using a two-timescale stochastic approximation scheme for a given SRP \( \theta \) with MSBE as the objective criterion. Through the adaptive tuning of features we will estimate the state value function of an SRP \( \theta \). For the faster timescale updates, we use the residual gradient algorithm from [16] that can be seen to track the minimum in \( r \) of \( G^\theta(\Phi, r) \) (the MSBE objective) for a given \( \Phi \) and \( \theta \).

In the next section, for ease of notation we will drop the dependence of \( \theta \) on the objective function \( G^\theta(\Phi, r) \), single stage cost \( k^\theta \), the transition probability \( P^\theta \) and the stationary distribution matrix \( D^\theta \), respectively, and will simply denote these quantities by \( G(\Phi, r), k, P \) and \( D \) respectively for the underlying SRP \( \theta \).
3 Gradient on the Grassmanian of Features

Let us define the function which we would like to minimize for adapting the features so as to estimate the value function.

\[ G(\Phi, r) = \| \Phi r - (k + \gamma P \Phi r) \|_D^2 = \| (I - \gamma P) \Phi r - k \|_D^2. \]  
\[(5)\]

We would like to minimize the function \( G(\Phi, r) \) as a function of both \( \Phi \) and \( r \). This can be achieved by first performing minimization over \( r \) for a fixed \( \Phi \) and then over \( \Phi \) itself. Define \( F(\Phi) = \min_r G(\Phi, r) \). This function can be rewritten as, \( F(\Phi) = G(\Phi, r^*(\Phi)) \), where \( r^*(\Phi) \) is the minimizer of \( G(\Phi, r) \) for a given fixed value of \( \Phi \). We make the following assumption:

**Assumption 4** The feature matrices \( \Phi \) are orthonormal.

The Grassmanian \( (\mathcal{M}) \) is the set of subspaces \( S \doteq \{ \Phi r \mid r \in \mathbb{R}^K \} \subset \mathbb{R}^N \) for which the (feature) matrices \( \Phi \) satisfy Assumption 4.

The gradient of the function \( F(\Phi) \) in the Grassmanian \( \mathcal{M} \) can be computed as the following, [14]:

\[ \nabla F = (I - \Phi \Phi^T) \frac{dF}{d\Phi}, \]  
\[(6)\]

see Eq.(2.70), pp. 321 of [14] for the above calculation of gradient of a function \( F(\Phi) \), with \( \Phi \) taking values in the set of orthonormal \( N \times K \) matrices, i.e., matrices \( \Phi \) for which \( \Phi^T \Phi = I \) (the identity matrix). The partial derivative of \( F(\Phi) \) with respect to \( \Phi \) denoted by \( \frac{dF}{d\Phi} \) can be obtained through envelope theorem, i.e., \( \frac{dF(\Phi)}{d\Phi} = \left. \frac{\partial (G(\Phi, r))}{\partial \Phi} \right|_{r=r^*(\Phi)} \). Hence, one needs to compute the partial derivative of the function \( G(\Phi, r) \) by keeping \( r \) fixed and evaluate the derivative at \( r^*(\Phi) \).

In [8], the gradient of a similar function \( G(\Phi, r) \) is derived for the discounted cost MDP for a given SDP. Set \( \Delta = (I - \gamma P) \) for notational simplicity. Then, \( (5) \) can be rewritten as

\[ G(\Phi, r) = (\Delta \Phi r - k)^T D(\Delta \Phi r - k). \]

By differentiating w.r.t. \( r \) the above, and equating to zero, one obtains

\[ r^*(\Phi) = \arg \min_r G(\Phi, r) = (\Phi^T \Delta^T D \Delta \Phi)^{-1} \Phi^T \Delta^T D k. \]  
\[(7)\]
The inverse on the RHS of (7) exists because of Assumptions 1 and 3. Since we use multi-timescale stochastic approximation, for any given update of $\Phi$, $r^*(\Phi)$ will be estimated in our scheme along the faster timescale.

The computation of $\frac{dF(\Phi)}{d\Phi}$ can be done along the lines of [8] (see Section 3 in [8]) or using matrix calculus. Then one obtains the partial derivative to be,

$$\frac{dF}{d\Phi} = 2\Delta^T D(\Delta \Phi r^*(\Phi) - k)(r^*(\Phi))^T.$$  \hspace{1cm} (8)

Under Assumption 4, from (6) (cf. Eq.(2.70) of [14]), one can write using (8) that

$$\nabla F = -2(I - \Phi \Phi^T)y(r^*(\Phi))^T,$$ \hspace{1cm} (9)

where $y = \Delta^T D(k - \Delta \Phi r^*(\Phi))$ is an $N$-dimensional column vector. To compute the gradient in (9), we need to compute both $y$ and $r^*(\Phi)$. Our algorithm runs a separate recursion to track $y$ along the same faster time scale recursion of $r$. $\Phi$ is then updated along the direction computed according to (9) on a timescale slower compared to the $y$ and $r$ updates (See Section 5 for the update rules of $r$, $y$ and $\Phi$).

4 Characterization of the Minima

In this section, we show a characterization of the minima of the objective function $F(\Phi)$. From (9), by setting the derivatives $\frac{dF}{d\Phi}$ equal to zero, we get

$$\Delta^T D(\Delta \Phi r^* - k)(r^*)^T = 0,$$ \hspace{1cm} (10)

where $r^*$ is given in (7) and $\Delta = (I - \gamma P)$ as defined before. Note that (10) is an outer-product $ab^T$ where $a \triangleq \Delta^T D(\Delta \Phi r^* - k) \in \mathcal{R}^N$ and $b \triangleq r^* \in \mathcal{R}^K$. Now for (10) to be zero, we need either of $a$ and $b$ to be zero vector, otherwise the outer-product $ab^T$ will not be zero.

Let us first assume $a = 0$, this implies $\Delta^T D(\Delta \Phi r^* - k) = 0$. For this to be true, since $\Delta^T D$ is a full rank matrix,

$$\Delta \Phi r^* - k = 0,$$ \hspace{1cm} (11)

$$\Delta \Phi r^* = k; \text{ hence} \quad \Phi r^* = \Delta^{-1}k \triangleq V,$$ \hspace{1cm} (12)

where $V$ is the value function of the given policy. Thus, if $\Phi$ satisfies (11), then from (12), it follows that $\Phi r^*$ correspond to the value function. Note if $\Phi$ is such that $k$ lies in the subspace
spanned by the column vectors of $\Delta \Phi$ (denoted by $S$), then $\frac{dF}{d\Phi} = 0$ and $F(\Phi) = 0$, thus implying such a $\Phi$ is a local minimum. In this case, the $\Phi$ will correspond to global minimum since $F(\Phi) = 0$.

Let us consider the next case $b = 0$, i.e., $r^* = 0$. Thus, from the expression for $r^*$ from (7) we have,

$$
(k^T D \Delta \Phi)(\Phi^T \Delta^T D \Delta \Phi)^{-1} = 0.
$$

$$
\implies (k^T D \Delta \Phi) = 0. \quad (13)
$$

(13) implies that $k$ is $D$-orthogonal to the column vectors of $\Delta \Phi$. For any $\Phi$, $F(\Phi)$ is the projection error corresponding to the projection of the cost vector $k$ into the subspace $S$. This will be maximum if $k$ lies in the orthogonal complement of $S$. In our present case (13) holds. Thus, $\Phi$ satisfying (13) will correspond to the global maximum and the objective function value is $F(\Phi) = k^T Dk$.

If $k$ neither lies in the span of $\Delta \Phi$ nor in its orthogonal complement, then atleast one entry in the vector $a$ will be non-zero and atleast one entry in the vector $b$ will be non-zero. Hence, their outer product will be non-zero indicating that the derivative is non-zero. Thus, only if $\Phi$ is such that either $k$ lies in the range space of $\Delta \Phi$ or in its orthogonal complement (with respect to the D-norm), the derivative of $F(\Phi)$ will be zero. In summary, the $\Phi$ corresponding to $\frac{dF}{d\Phi} = 0$ will either correspond to global minima or global maxima of the function $F(\Phi)$. In particular $\Phi$’s corresponding to the global minima give exact representation of the value function and $\Phi$ corresponding to the global maxima result in unstable equilibrias. From the foregoing, it is evident that the gradient descent on the Grassmanian will converge to the global minima of $F$. From (12), the product $\Phi r^*$, then will correspond to the value function with 'sufficiently rich' noise, maxima or unstable equilibria for the algorithm ([17], chapter IV).

5 The Actor-critic Control Algorithm

In this section we describe the stochastic update rules corresponding to our control algorithm. Let \{a(n)\}, \{b(n)\} and \{c(n)\} be three sequences of step-size schedules that satisfy the following requirements:
Assumption 5 The step-sizes \(a(n), b(n), c(n) > 0\), \(\forall n\). Further,

\[
\sum_n a(n) = \sum_n b(n) = \sum_n c(n) = \infty, \quad (14)
\]

\[
\sum_n (a^2(n) + b^2(n) + c^2(n)) < \infty, \quad (15)
\]

\[
\lim_{n \to \infty} \frac{b(n)}{a(n)} = \lim_{n \to \infty} \frac{c(n)}{b(n)} = 0. \quad (16)
\]

Note that (14) and (15) are standard requirements on step-size sequences. From (16), the timescale corresponding to \(a(n), n \geq 0\) is the fastest and the one corresponding to \(c(n), n \geq 0\) is the slowest.

Let \(X_n\) denote the state of the MDP at time \(n\) and \(Z_n\) denote the action chosen at time \(n\). We divide the number of iterations into subintervals of length \(M\) as \(\{n \geq 0\}\) can be divided using epochs as \((p - 1) \times 2M\) to \((p \times 2M) - 1, p \geq 1\). The policy parameter \(\theta\) gets updated according to (20) only at the end of every \(2M\) epochs. The update rules for \(r, y\) and \(\Phi ((17), (18)\) and (19)) are followed with the step-sizes \(a(n), b(n)\) fixed during the epoch. The step-sizes \(\{a(n), b(n)\}\), the perturbation parameters \(\{\epsilon_n, \Delta_n\}\) are fixed during the epoch and changed only at the end of the epochs. Here, \(\Delta_n = (\Delta_n(1), \Delta_n(2), \ldots, \Delta_n(l))\) with \(\Delta_n(k), k \in \{1, 2, \ldots, L\}, n \geq 0\) are independent and identically distributed (i.i.d) are Bernoulli random variables which takes values \(\{\pm 1\}\) with probability \(\frac{1}{2}\) and the perturbation parameter \(\epsilon_n \to 0\) 'slowly enough' (See [9] for more details). During the odd \(M\)-step time intervals, the policy parameter \(\theta\) is set at \(\theta_n + \epsilon_n \Delta_n\), and in the event intervals, the policy parameter is set at \(\theta_n - \epsilon_n \Delta_n\). At the end of odd \(M\) time steps of the epoch, the objective \(\rho(\theta_n + \epsilon_n \Delta_n)\) is computed as \(\Phi_n r_n\) and in a similar fashion at the end of even time steps of the epoch, \(\rho(\theta_n - \epsilon_n \Delta_n)\) is computed as \(\Phi_n r_n\). At the end of each \(2M\)-time interval, \(\theta_n\) is updated according to (20).

The online control algorithm with feature adaptation using stochastic approximation is given below:

(A) [First (Fastest) Time Scale Update]

(A1) [Residual Gradient Scheme] On the faster time scale, given the current update \(\Phi(n)\) of the feature matrix for the current policy with parameter \(\theta_n\), the residual gradient scheme updates the weight vector \(r\) as

\[
r_{n+1} = r_n + a\left(\left\lfloor \frac{n}{2M} \right\rfloor \right) \left(k(X_n, Z_n) + \gamma \phi_{X,n+1}^T(n)r_n\right)
\]
\[-\phi^T_{X_n}(n)r_n \times (\phi_{X_n}(n) - \gamma \phi_{\tilde{X}_{n+1}}(n)),\]  

starting from some \(r_0 = (r_0(1), \ldots, r_0(N))^T\). Here, \(\tilde{X}_{n+1}\) is also a sample generated with the distribution \(p(\cdot \mid X_n, \theta_n)\), that is conditionally independent of \(X_{n+1}\) given \(X_n\) (though both have the same conditional law given \(X_n\)). Note that the step sizes are kept constant over successive \(2M\) iterations. Here, \(\lfloor \cdot \rfloor\) denotes the floor function.

(A2) [Intermediate step in the computation of gradient on the Grassmanian):

\[
\psi(n) \triangleq k(X_n, Z_n) + \gamma \phi^T_{\tilde{X}_{n+1}}(n)r_n - \phi^T_{X_n}(n)r_n, \\
\tilde{\psi}(n) \triangleq k(X_n, Z_n) + \gamma \phi^T_{\tilde{X}_{n+1}}(n)r_n - \phi^T_{X_n}(n)r_n.
\]

Now for \(i = 1, \ldots, N, n \geq 0,\)

\[
y_{n+1}(i) = y_n(i) + a(\lfloor \frac{n}{2M} \rfloor)(I^i_{n+1}(\psi(n+1)) - \tilde{\psi}(n) - y_n(i)),
\]

starting from some \(y_0 = (y_0(1), \ldots, y_0(N))^T\). Here, \(I^i_{n+1}\) denotes \(I\{X_{n+1} = i\}\), the indicator random variable of state \(i\) at time \(n+1\).

(B) [Second (Medium) Time Scale Update]

\[
\Phi(n+1) = \Gamma^1(\Phi(n) + b(\lfloor \frac{n}{2M} \rfloor)2(I - \Phi(n)\Phi(n)^T)y_n(r_n)^T),
\]

\(n \geq 0,\) starting with an initial feature matrix \(\Phi(0)\) having all its columns as orthonormal vectors. In (19), \(\Gamma^1(\cdot)\) is the operator that performs the Gram-Schmidt orthonormalization step.

(C) [Third (Slowest) Time-Scale Update (policy-Update):] The \(k\)th component of the policy parameter \(\theta\) for \(k \in \{1, 2, \ldots, L\}\) gets updated as

\[
\theta_{n+1}(k) = \Gamma^2(\theta_n(k) - c(n) \times \frac{[\rho(\theta_n + \epsilon_n \Delta_n) - \rho(\theta_n - \epsilon_n \Delta_n)]}{2\epsilon \Delta_n(k)}),
\]

Note that in (20), even though \(\theta\) update is shown for all \(n\), we update \(\theta\) only when \(n \mod 2M = 0\). In (20), \(\Gamma^2 : \mathcal{R}^L \rightarrow C\) is a projection operator that projects any \(\theta \in \mathcal{R}^L\)
to a compact set $C = \{ \theta \in \mathcal{R}^L | q_i(\theta) \leq 0, i = 1, 2, \ldots, s \}$, where $q_i(\theta), i = 1, 2, \ldots, s$, are continuously differentiable functions on $\mathcal{R}^L$ that represent the constraints specifying the compact region.

6 Convergence analysis

In this section we provide a brief outline the proof of convergence of our multi-timescale stochastic approximation ((17), (18), (19) and (20)) control algorithm to a local minimum. The detailed proof will be presented in a journal version paper.

We begin with the analysis of the faster time scale recursion (17) (step (A1) of the algorithm). From Assumption 5, $c(n) = o(a(n))$ and $b(n) = o(a(n))$. Hence, we can let $\theta_n(\Delta)$ to be a constant $\theta(\Delta)$ and $\Phi_n$ to be a constant $\Phi$ while analyzing (17), see [18] or Chapter 6 of [17].

Let $F_n = \sigma(X_m, Z_m, r_m, \theta_m, \Delta_m, m \leq n)$, $n \geq 0$ denote a sequence of sigma fields generated by the mentioned quantities and let

$$M_{n+1}^1 = (k(X_n, Z_n) + \gamma \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n)$$

$$\hat{\theta}_n = \phi_X - \gamma \phi_{\hat{X}_{n+1}}$$

$$-E[(k(X_n, Z_n) + \gamma \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n)$$

$$\hat{\theta}_n = \phi_X - \gamma \phi_{\hat{X}_{n+1}} | F_n].$$

**Lemma 1** $(M_{n+1}^1, F_n), n \geq 0$ forms a martingale difference sequence with

$$E[\| M_{n+1}^1 \|^2 | F_n] \leq \hat{K}(1 + \| r_n \|^2),$$

for some constant $\hat{K} > 0$.

The ODE associated with (17) is the following:

$$\dot{r}(t) = \sum_{i \in S} d^\theta(i) \sum_{a \in A} \pi^\theta(i, a) \left[ (k(i, a) + \gamma \sum_{j \in S} p_{i,j}(a) \phi_j^T r(t) - \phi_i^T r(t)) \times \left( \phi_i - \gamma \sum_{j \in S} p_{i,j}(a) \phi_j \right) \right]$$

$$= \Phi^T (I - \gamma P^\theta)^T D^\theta (k^\theta - (I - \gamma P^\theta) \Phi r(t))$$

$$\triangle = h^2(\theta, \Phi, r(t)), (24)$$

$$\triangle = h^2(\theta, \Phi, r(t)), (25)$$
where $k^\theta$ is a vector of dimension $N$ with $i$th component $k^\theta(i) = \sum_{a \in A} \pi_\theta(i, a) k(i, a)$. $P^\theta$ is a matrix of dimension $N \times N$ with the $(i, j)$th component being $P^\theta(i, j) = \sum_{a \in A} p_{i,j}(a)\pi_\theta(i, a)$ and $D^\theta$ is a diagonal matrix of dimension $N \times N$ whose entries correspond to the stationary distribution of the SRP $\theta$.

**Lemma 2** The ODE (25) has $r^*_\theta, \Phi \triangleq (\Delta \Phi)^T D^\theta (\Delta \Phi))^{-1} (\Delta \Phi)^T D^\theta k^\theta$ as its unique globally asymptotically stable equilibrium.

Let $r^*_\theta, \Phi(i)$ denote the $i$th component of $r^*_\theta, \Phi$, $i = 1, \ldots, K$. The next result follows from a crucial result on stability of stochastic approximations given in [17].

**Proposition 3** With $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi$, $\forall n$, $r(n), n \geq 0$ governed according to (17) are uniformly bounded almost surely. Further, $r(n) \to r^*_\theta, \Phi$ as $n \to \infty$ almost surely.

Now consider the $y$ recursion in Step A2 of the algorithm. Since $c(n) = o(a(n))$, one may again let $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi$ (a constant) while analyzing the update (A2). Now let

$$\psi^*(n) \triangleq k(X_n, Z_n) + \gamma \phi^T_{X_{n+1}} r^*_\theta, \Phi - \phi^T_{X_n} r^*_\theta, \Phi,$$

$$\tilde{\psi}^*(n) \triangleq k(X_n, Z_n) + \gamma \phi^T_{X_{n+1}} r^*_\theta, \Phi - \phi^T_{X_n} r^*_\theta, \Phi.$$

(26)

(27)

In view of Proposition 3, one may analyze the following recursion in place of (18):

$$y_{n+1}(i) = y_n(i) + b(n) I^i_{n+1} (\psi^*(n + 1) - \gamma \tilde{\psi}^*(n) - y_n(i)).$$

(28)

The ODE associated with recursion (18) can thus be seen to be

$$\dot{y}(t) = (I - \gamma P^\theta)^T D z^*_\theta, \Phi - y(t),$$

(29)

where $z^*_\theta, \Phi = (k^\theta - (I - \gamma P^\theta) \Phi r^*_\theta, \Phi)$.

Let $\mathcal{G}(n) = \sigma(X_m, Z_m, y_m, m \leq n), n \geq 0$ denote an increasing sequence of sigma fields and let $M^2_n, n \geq 0$ be defined as follows:

$$M^2_{n+1} = (\psi^*(n + 1) - \gamma \tilde{\psi}^*(n))$$

$$- E[(\psi^*(n + 1) - \gamma \tilde{\psi}^*(n)) | \mathcal{G}(n)].$$

(30)

It is easy to see that $(M^2_n, \mathcal{G}(n)), n \geq 0$ forms a martingale difference sequence.
**Proposition 4** Given \( \theta_n \equiv \theta \) and \( \Phi(n) \equiv \Phi, \forall n \), the updates \( y_n, n \geq 0 \) governed by (18) are uniformly bounded almost surely and converge to \( y^*_\theta, \Phi \triangleq (I - \gamma P^\theta)^T Dz^*_{\theta, \Phi} \) as \( n \to \infty \).

Consider now the medium time scale recursion in step (B) of the algorithm. From Assumption 5, \( c(n) = o(b(n)) \). Hence, we can let \( \theta_n \) to be a constant \( \theta \) while analyzing the update (19). Note that for an \( N \times K \)-matrix \( \Phi = (\phi_i^T, i = 1, \ldots, N)^T \), \( \Gamma^1(\Phi) \) is the operator that performs the Gram-Schmidt orthonormalization step. As a consequence of the \( \Gamma^1 \)-operator, the iterates in (19) remain almost surely uniformly bounded. One can rewrite the recursion (19) as follows:

\[
\Phi(n + 1) = \Gamma^2\left(\Phi(n) + c(n)2(I - \Phi(n)\Phi(n)^T)y^*_{\theta, \Phi(n)}(r^*_{\theta, \Phi(n)})^T + \epsilon(n)\right)
\]

where \( \epsilon(n) = 2(I - \Phi(n)\Phi(n)^T)(y_n(r_n)^T - y^*_{\theta, \Phi(n)}(r^*_{\theta, \Phi(n)})^T) \). From Propositions 3 and 4, it follows that \( \epsilon(n) \to 0 \) almost surely as \( n \to \infty \).

Consider now the following ODE corresponding to the recursion (31):

\[
\dot{\Phi}(t) = \hat{\Gamma}^1(-\nabla F_\theta(\Phi(t))),
\]

where for any continuous function \( v : \mathbb{R}^{N \times K} \to \mathbb{R}^{N \times K} \),

\[
\hat{\Gamma}^1(v(y)) = \lim_{\eta \to 0} \left( \frac{\Gamma^1(y + \eta v(y)) - y}{\eta} \right).
\]

In case the above limit is not unique, we let \( \hat{\Gamma}^1(v(y)) \) be the set of all possible limit points (see pp. 191 of [19]).

Let

\[
\mathcal{K}^1 \triangleq \{ \Phi \in \mathcal{M} | \hat{\Gamma}^1(\nabla F(\Phi)) = 0 \},
\]

(34)

denote the set of all fixed points of (32).

We now have the following result:

**Theorem 5** As \( n \to \infty \), \( \Phi(n) \to \mathcal{K}^1 \) almost surely.

Now consider the slowest recursion, i.e., the \( \theta \) update corresponding to (20), i.e., \( \forall k \in \{1, 2, \ldots, L\} \)

\[
\theta_{n+1}(k) = \Gamma^2\left(\theta_n(k) - c(n)\frac{\rho(\theta_n + \epsilon \Delta_n) - \rho(\theta_n - \epsilon \Delta_n)}{2\epsilon \Delta_n(k)}\right)
\]

(35)
where $c(n)$ is the step-size parameter and $\Delta_n(k)$ is a Bernoulli random variable which takes values $\{\pm 1\}$ with probability $\frac{1}{2}$. Let $I_n = \sigma\{X_m, Z_m, \theta_m, m \leq n, \Delta_m, m < n\}$. The ODE corresponding to (35) can be written as,

$$\dot{\theta}(t) = \hat{\Gamma}^2(-\nabla \rho(\theta)), \tag{36}$$

where for any continuous function $v : R^L \to R^L$,

$$\hat{\Gamma}^2(v(y)) = \lim_{0 < \eta \to 0} \left( \frac{\Gamma^2(y + \eta v(y)) - y}{\eta} \right). \tag{37}$$

In case the above limit is not unique, we let $\hat{\Gamma}^2(v(y))$ be the set of all possible limit points (see pp. 191 of [19]). Let

$$\mathcal{K}^2 \triangleq \{ \theta \in R^L \mid \hat{\Gamma}^2(\nabla \rho(\theta)) = 0 \}, \tag{38}$$

denote the set of all fixed points of (36). The ODE (36) is well posed from the Lipschitz continuity of $\nabla \rho(\theta)$. Note that $\hat{V}(\theta) = \rho(\theta)$ itself serves as an associated strict Lyapunov function for the ODE (36).

We now have the following main result:

**Theorem 6** As $n \to \infty$, $\theta(n) \to \mathcal{K}^2$ almost surely.

### 7 Numerical Experiments

In this section, we demonstrate the performance of our algorithm on two random MDP settings (MDP1 and MDP2 respectively) whose parameters are given in Table 1. We also show results of another algorithm where the residual gradient scheme is replaced by TD(0). The MDPs were randomly generated using a tool box. Here the transition and reward structure is arbitrarily set. We set the discount factor $\gamma = 0.9$ in both MDP settings. We randomly set the weights for the objective $\beta(l), l \in \{1, 2, \ldots, N\}$ with their sum normalized to 1. We set the duration of an epoch $M = 100$. In MDP1, the cardinalities of both state and action spaces are 10 and thus the number of SDPs is $10^{10}$. In MDP2, the same number is 100 and thus the number of SDPs is $100^{100}$. It is clear that the number of SDPs in MDP2 is significantly more than in MDP1. Our actor-critic feature adaptation control algorithm was run on each individual setting where the critic first estimates the value function and the actor subsequently uses the estimate to improve the policy. We let the step-sizes be $a(n) = 1/n^{0.6}, b(n) = 1/n^{0.8}$ and $c(n) = 1/n$, respectively. In the plots, $Y$-axis corresponds to policy performance $-\rho(\theta)$ (weighted discounted reward, i.e., negative of
Table 1: Details of the MDP settings used in the plots

<table>
<thead>
<tr>
<th>S.N.</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MDP1</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>2. MDP2</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

our objective) and X-axis corresponds to number of iterations. The policy features $\sigma(s,a)$ were randomly generated and fixed during the experiments.

We show first the results using our algorithm as described in Section 5. Figures 1 and 2 corresponds to the MDP1 and MDP2 settings respectively. The policy improvement can be seen to be noisy as is the case for any stochastic update rule. However, the average behaviour of $-\rho(\theta)$ can be seen to increase with the number of iterations.

Figures 3 and 4 depict the results of experiments when TD(0) is used in place of the residual gradient scheme for the faster scale updates in the MDP1 and MDP2 settings respectively. It can be seen from Figures 3 and 4 that TD(0) despite not minimizing the MSBE error objective that we considered shows good performance as it provides good approximation to the value function. Although the theoretical convergence of the resulting scheme (using the MSBE objective) under TD(0) has not been proved, this scheme is computationally advantageous as it uses only one simulation sample (for the ‘next’ state generation, instead of two such samples) at each iterate.

8 Concluding Remarks

We presented the first online actor-critic scheme for the weighted discounted cost MDP setting that incorporates a feature adaptation algorithm in the critic. The critic uses gradient search on the Grassman manifold of features to estimate the value function and the actor uses an SPSA scheme to improve the policy performance. The algorithm is seen to perform well over randomly generated MDP settings involving $10^{10}$ and $100^{100}$ policies. A modified version of the algorithm with TD(0) on the faster timescale is also seen to exhibit good performance even though TD(0) is not designed as such for finding an optimum for the MSBE error objective. It would be interesting to investigate theoretical guarantees when our algorithm is used with the TD(0) scheme in place of residual gradient scheme on the faster-timescale updates. As future work, we shall design a similar control algorithm using feature updates in the Grassmanian for the long-run average cost objective. We shall also include a sparsity penalty, thereby modifying the scheme to prefer sparser basis vectors to improve scalability. Further, it would be interesting to apply deterministic perturbation sequences based on Hadamard matrices for the SPSA update.
Figure 1: Plot of $-\rho(\theta)$ vs. $n$ with Residual gradient scheme on the Faster Time-Scale for MDP1 setting
Figure 2: Plot of $-\rho(\theta)$ vs. $n$ with Residual gradient scheme on the Faster Time-Scale for MDP2 setting.
Figure 3: Plot of $-\rho(\theta)$ vs. $n$ with TD(0) on the Faster Time-Scale for MDP1 setting
Figure 4: Plot of $-\rho(\theta)$ vs. $n$ with TD(0) on the Faster Time-Scale for MDP2 setting
References


