53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

An Actor Critic Algorithm Based on Grassmanian Search

Prabuchandran K.J.[1], Shalabh Bhatnagar[1], *Senior Member, IEEE*, Vivek S. Borkar[2], *Fellow, IEEE*

***Abstract*— We propose the first 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 parameterize 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 provide an outline of the proof of convergence of our control algorithm to a locally optimum policy. We show empirical results using our algorithm as well as a similar algorithm that uses temporal difference (TD) learning in place of the residual gradient scheme for the faster timescale updates.**

***Index Terms*— Control, feature adaptation, online learning, residual gradient scheme, temporal difference learning, stochastic approximation, Grassman manifold.**

I. Introduction

In sequential decision making problems one essentially faces the problem of optimal decisions 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.

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 [1]) or policy (actor-only methods [2]) or both (actor-critic methods) [3]. The actor-critic setup offers several advantages over actor-only and critic-only methods [3].

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 θ. The actor's policy parameter θ is updated in the direction of improving the performance metric (in our case the weighted long-run discounted cost of a policy, $\rho(\theta)$). The direction of improvement is found by changing the policy parameters along the negative gradient direction [3]. 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. [4] 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 [5], [6].

Various feature adaptation based methods to approximate value function have been studied in the literature. In [7], 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 [7] is presented in [8]. In [9], a non-parameterized adaptive scheme for basis selection is proposed in conjunction with TD.

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 bases have been studied in the literature for discounted cost MDP. The problem of control with adaptive

[1] Department of Computer Science and Automation, Indian Institute of Science, Bangalore-560012, India. {prabu.kj,shalabh}@csa.iisc.ernet.in

[2] Department of Electrical Engineering, Indian Institute of Technology, Mumbai, Powai-400076, India. borkar.vs@gmail.com

The work of PKJ has been supported through a TCS fellowship. SB acknowledges support from projects supported by DST, Xerox Corporation, USA and the Robert Bosch Centre, IISc. VSB acknowledges support from the J.C. Bose fellowship & a grant from Dept. of Science and Technology for a project titled 'Distributed computation over large networks and high-dimensional data analysis'.

bases for the average cost setting is considered for the first time in [10] and actor-critic algorithms are developed. The basis functions are parameterized and their parameters are updated using the method given in [7]. [10] 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 [5], [6] 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 most other algorithms 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 a randomly generated MDP setting. 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.

The rest of the paper is organized as follows: In Section II, we discuss the problem setting of MDP and function approximation. In Section III, we describe a key result related to the function gradient on the Grassman manifold from [11]. In Section IV, we characterize the minima of our objective function. In Section V, 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 VI. Results of numerical experiments using our algorithm and TD are then presented in Section VII. Finally, we present our concluding remarks and discuss future work in Section VIII.

II. 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 $S = \{1, 2, \ldots, N\}$ with $N < \infty$. 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 $p : S \times S \times A \to [0, 1]$, 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.

A deterministic policy $\bar{\pi}$ is a sequence of maps $\bar{\pi} \stackrel{\triangle}{=} \{\mu_0, \mu_1, \ldots\}$ 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 μ does not depend on j, we call $\bar{\pi}$ or by abuse of notation μ itself, a stationary deterministic policy (SDP). A stationary randomized policy (SRP) π 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 parameterize the policy π using a parameter θ. 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 often use SRP θ for SRP π_θ.

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 θ.

A commonly used parameterization that we also use in our experiments and which satisfies Assumption 2 is the Gibbs distribution, $\pi_\theta(i, a) = \dfrac{\exp(\theta^T \sigma(i, a))}{\sum_a \exp(\theta^T \sigma(i, a))}$, where $\sigma(i, a) \in \mathcal{R}^L$ is the policy feature for the state-action pairs.

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

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

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

$$V^\theta(i) = \sum_{n=0}^{\infty} \mathbb{E}[\gamma^n k(X_n, Z_n) | X_0 = i, \theta], \tag{2}$$

where $\gamma \in (0, 1)$ is the given discount factor of the MDP.

We achieve the goal of minimizing $\rho(\theta)$ by performing gradient descent in the parameter space of θ. In our algorithm, we update the actor parameter θ along the negative gradient direction of $\rho(\theta)$, using SPSA gradient estimates according to (18) in Section V. This update rule under mild conditions on step sizes $c(n), n \geq 0$ converges to a parameter $\theta^\star$ such that π_{θ^*} is a locally optimum policy.

To compute $\rho(\theta + \epsilon\Delta)$ and $\rho(\theta - \epsilon\Delta)$ in (18), 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 θ. The state value function $V^\theta(i)$, $i \in S$ satisfies the Bellman equation.

$$V^\theta = k^\theta + \gamma P^\theta V^\theta, \tag{3}$$

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

To solve the system of equations in (3), one needs the matrix P^θ and the cost vector k^θ explicitly. In practice, P^θ is often not explicitly known and may need to be estimated in order to numerically solve (3). 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. Let Φ denote the $N \times K$ feature matrix with ϕ_i^T, $i \in S$, as its rows. Thus $\Phi = [[\phi_i(k)]]_{i \in S, k=1,\ldots,K}$. Let $\phi(k) \stackrel{\triangle}{=} (\phi_i(k), i \in S)^T$ denote the kth column of Φ, $k \in \{1, \ldots, K\}$ having dimension N. From the foregoing, the (j,k)th element of Φ corresponds to $\phi_j(k)$. We now make the following assumption.

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

From Assumption 3, Φ has full column rank. Let $d^\theta(i)$ be the stationary probability of $\{X_n\}$ (under SRP θ) being in state $i \in S$. Also, let D^θ 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 $z \in \mathcal{R}^N$.

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 $G_\theta(\Phi, r) = \| \Phi r - (k^\theta + \gamma P^\theta \Phi r) \|^2_{D^\theta}$, with $r \in \mathcal{R}^K$ and the aim is to find a parameter $r^*_{\theta,\Phi} \in \mathcal{R}^K$ that minimizes $G_\theta(\Phi, r)$ over all r. It is assumed many times that the matrix Φ is fixed or given *a priori*, That may lead to poor approximation of the value function. In section V we present the scheme (15)-(17) to tune the feature matrix Φ using a two-timescale stochastic approximation scheme for a given SRP θ with MSBE as the objective criterion. Through the adaptive tuning of features we will estimate the state value function of an SRP θ. For the faster timescale updates, we use the residual gradient algorithm from [12] that can be seen to track the minimum in r of $G_\theta(\Phi, r)$ (the MSBE objective) for a given Φ and θ.

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

III. 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.

$$\begin{aligned} G(\Phi, r) &= \| \Phi r - (k + \gamma P \Phi r) \|^2_D \\ &= \| (I - \gamma P)\Phi r - k \|^2_D . \end{aligned} \tag{4}$$

We would like to minimize the function $G(\Phi, r)$ as a function of both Φ and r. This can be achieved by first performing minimization over r for a fixed Φ and then over Φ itself. Define $F(\Phi)$ as $F(\Phi) = \min_r G(\Phi, r)$. This function can be rewritten as, $F(\Phi) = G(\Phi, r^*(\Phi))$, where $r^*(\Phi)$[1] is the minimizer of $G(\Phi, r)$ for a given fixed value of Φ. We make the following assumption :

Assumption 4: The feature matrices Φ are orthonormal.

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

The gradient of the function $F(\Phi)$ in the Grassmanian $\mathcal{M}$ can be computed as the following, [11]:

$$\nabla F = \left(I - \Phi\Phi^T\right) \frac{dF}{d\Phi}, \tag{5}$$

see Eq.(2.70), pp. 321 of [11] for the above calculation of gradient of a function $F(\Phi)$, with Φ taking values in the set of orthornormal $N \times K$ matrices, i.e., matrices Φ for which $\Phi^T\Phi = I$ (the identity matrix). The partial derivative of $F(\Phi)$ with respect to Φ 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 $r^*(\Phi)$.

The computation of $\frac{dF(\Phi)}{d\Phi}$ can be done along the lines of [4] (see Section III in [4]) or using matrix calculus. Set $\Delta = (I - \gamma P)$ for notational simplicity. Then one obtains the partial derivative to be

$$\frac{dF}{d\Phi} = 2\Delta^T D(\Delta\Phi r^*(\Phi) - k)(r^*(\Phi))^T. \tag{6}$$

Under Assumption 4, from (5) (cf. Eq.(2.70) of [11]), one can write using (6) that

$$\nabla F = -2(I - \Phi\Phi^T) y^*(\Phi)(r^*(\Phi))^T, \tag{7}$$

where $y^*(\Phi) = \Delta^T D(k - \Delta\Phi r^*(\Phi))$ is an $N \times 1$ column vector. To compute the gradient in (7), we need to compute both $r^*(\Phi)$ and $y^*(\Phi)$. Our algorithm runs an r-recursion to track $r^*(\Phi)$ and a separate y-recursion along the same faster time scale recursion of r to track $y^*(\Phi)$. Φ is then updated along the direction computed according to (7) on a timescale slower compared to the y and r updates (See Section V for the update rules of r, y and Φ).

IV. Characterization of the Minima

In this section, we show a characterization of the minima of the objective function $F(\Phi)$. From (7), by setting the derivatives $\frac{dF}{d\Phi}$ equal to zero, we get

$$\Delta^T D(\Delta\Phi r^* - k)(r^*)^T = 0, \tag{8}$$

with $r^* \equiv r^*(\Phi)$ and $\Delta = (I - \gamma P)$ as defined before. Note that (8) is an outer-product ab^T where $a \stackrel{\triangle}{=} \Delta^T D(\Delta\Phi r^* - k) \in \mathcal{R}^N$ and $b \stackrel{\triangle}{=} r^* \in \mathcal{R}^K$. Now for (8) to be zero, we need either a or b to be the 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,

$$(\Delta\Phi r^* - k) = 0, \text{or} \tag{9}$$

$$\Phi r^* = \Delta^{-1} k \stackrel{\triangle}{=} V, \tag{10}$$

[1] $r^*(\Phi) = \arg\min_r G(\Phi, r) = (\Phi^T\Delta^T D\Delta\Phi)^{-1}\Phi^T\Delta^T Dk$.

where V is the value function of the given policy. Thus, if Φ satisfies (9), then from (10), it follows that Φr^* correspond to the value function. Note if Φ is such that k lies in the subspace spanned by the column vectors of $\Delta\Phi$ (denoted by $\bar{S}$), then $\frac{dF}{d\Phi} = 0$ and $F(\Phi) = 0$, thus implying such a Φ is a local minimum. In this case, the Φ will correspond to global minimum since $F(\Phi) = 0$.

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

$$\begin{aligned}(k^T D\Delta\Phi)(\Phi^T\Delta^T D\Delta\Phi)^{-1} &= 0.\\ \implies (k^T D\Delta\Phi) &= 0. \end{aligned} \tag{11}$$

(11) implies that k is D-orthogonal to the column vectors of $\Delta\Phi$. For any Φ, $F(\Phi)$ is the projection error corresponding to the projection of the cost vector k into the subspace $\bar{S}$. This will be maximum if k lies in the orthogonal complement of $\bar{S}$. In our present case (11) holds. Thus, Φ satisfying (11) 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 at least one entry in the vector a will be non-zero and at least 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 Φ 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 Φ corresponding to $\frac{dF}{d\Phi} = 0$ will either correspond to global minima or global maxima of the function $F(\Phi)$. In particular Φ's corresponding to the global minima give exact representation of the value function and Φ corresponding to the global maxima result in unstable equilibria. By standard properties of stochastic approximation, under mild 'richness' assumptions on noise, the gradient descent on the Grassmanian will converge to global minimum of F with probability one (see [13], Ch. 4). From (10), the product Φr^* will correspond to the value function.

V. 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,

$$\sum_n a(n) = \sum_n b(n) = \sum_n c(n) = \infty, \tag{12}$$

$$\sum_n (a^2(n) + b^2(n) + c^2(n)) < \infty, \tag{13}$$

$$\lim_{n\to\infty} \frac{b(n)}{a(n)} = \lim_{n\to\infty} \frac{c(n)}{b(n)} = 0. \tag{14}$$

Note that (12) and (13) are standard requirements on step-size sequences. From (14), 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 $2M$ as $(p-1)\times 2M$ to $(p \times 2M) - 1$, $p \geq 1$. The policy parameter θ gets updated according to (18) only at the end of every $2M$ epochs. The update rules for r, y and Φ, see (15)-(17), are followed with the step-sizes $a(n), b(n)$ fixed during the $2M$ epochs. The step-sizes $\{a(n), b(n)\}$, the perturbation parameters $\{\epsilon_n, \Delta_n\}$ are fixed during the epochs and changed only at the end of the epochs. Here, $\Delta_n = (\Delta_n(1), \Delta_n(2), \ldots, \Delta_n(l))$ with $\Delta_n(k), k \in \{1, 2, \ldots, L\}, n \geq 0$ being independent and identically distributed (i.i.d) Bernoulli random variables which take values $\{\pm 1\}$ with probability $\frac{1}{2}$ and the perturbation parameter $\epsilon_n \to 0$ 'slowly enough' (See [5] for more details). During the odd M-step time intervals, the policy parameter θ is set at $\theta_n + \epsilon_n\Delta_n$, and in the even 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 estimated as $\beta^T\Phi_n r_n$ where β corresponds to the vector of weight $\beta(l),\ l \in \{1, 2, \ldots, N\}$ as in (1) and in a similar fashion at the end of every even M time steps, $\rho(\theta_n - \epsilon_n\Delta_n)$ is computed as $\beta^T\Phi_n r_n$. At the end of each $2M$-time interval, θ_n is updated according to (18).

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 θ_n, the residual gradient scheme updates the weight vector r as

$$r_{n+1} = r_n + a(\lfloor\frac{n}{2M}\rfloor)\Big(k(X_n, Z_n) + \gamma\phi^T_{X_{n+1}}(n)r_n - \phi^T_{X_n}(n)r_n\Big) \times \Big(\phi_{X_n}(n) - \gamma\phi_{\tilde{X}_{n+1}}(n)\Big), \tag{15}$$

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, $\lfloor . \rfloor$ denotes the floor function.

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

$$\psi(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{X_{n+1}}(n)r_n - \phi^T_{X_n}(n)r_n,$$
$$\tilde{\psi}(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{\tilde{X}_{n+1}}(n)r_n - \phi^T_{X_n}(n)r_n.$$

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

$$y_{n+1}(i) = y_n(i) + a(\lfloor\frac{n}{2M}\rfloor)\Big(I^i_{n+1}(\psi(n+1) - \tilde{\psi}(n) - y_n(i)\Big), \tag{16}$$

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]

$$\Phi(n+1) = \Gamma^1(\Phi(n)+ b(\lfloor\frac{n}{2M}\rfloor)2\big(I - \Phi(n)\Phi(n)^T\big)y_n(r_n)^T), \quad (17)$$

$n \geq 0$, starting with an initial feature matrix $\Phi(0)$ having all its columns as orthonormal vectors. In (17), $\Gamma^1(\cdot)$ is the operator that performs the Gram-Schmidt orthonormalization step.

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

$$\theta_{n+1}(k) = \Gamma^2\bigg(\theta_n(k) - c(n)\times \Big[\frac{\rho(\theta_n + \epsilon_n\Delta_n) - \rho(\theta_n - \epsilon_n\Delta_n)}{2\epsilon\Delta_n(k)}\Big]\bigg). \quad (18)$$

Note that in (18), even though the θ update is shown for all n, we update θ only when $n \mod 2M = 0$. In (18), $\Gamma^2 : \mathcal{R}^L \to C \subset \mathcal{R}^L$ 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 continously differentiable functions that represent the constraints specifying the compact region.

VI. Convergence Sketch

In this section we provide a brief outline of the proof of convergence of our multi-timescale stochastic approximation algorithm to a local minimum. The detailed proof will be presented in a journal version of the paper.

We begin with the analysis of the faster time scale recursion (15) (step (A1) of the algorithm). From Assumption 5, $c(n) = o(a(n))$ and $b(n) = o(a(n))$. Hence, we can let θ_n (Δ_n) to be a constant θ ($\bar{\Delta}$) and $\Phi(n)$ to be a constant Φ while analyzing (15), see Chapter 6 of [13].

The ODE associated with (15) is the following:

$$\begin{aligned}\dot{r}(t) &= \sum_{i\in S} d^\theta(i) \sum_{a\in A}\pi_\theta(i,a)\bigg[\big(k(i,a) + \gamma\sum_{j\in S} p_{i,j}(a)\phi_j^T r(t) \\ &- \phi_i^T r(t)\big) \times \big(\phi_i - \gamma\sum_{j\in S} p_{i,j}(a)\phi_j\big)\bigg] \\ &= \Phi^T(I - \gamma P^\theta)^T D^\theta(k^\theta - (I - \gamma P^\theta)\Phi r(t)) \quad (19)\end{aligned}$$

where k^θ is a vector of dimension N with ith component $k^\theta(i) = \sum_{a\in A}\pi_\theta(i,a)k(i,a)$, P^θ 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^θ is a diagonal matrix of dimension $N \times N$ whose entries correspond to the stationary distribution of the SRP θ.

Lemma 1: The ODE (19) has $r^*_{\theta,\Phi} \stackrel{\triangle}{=} ((\Delta\Phi)^T D^\theta(\Delta\Phi))^{-1}(\Delta\Phi)^T D^\theta k^\theta$ as its unique globally asymptotically stable equilibrium.

Let $r^*_{\theta,\Phi}(i)$ denote the ith component of $r^*_{\theta,\Phi}$, $i = 1, \ldots, K$. The next result follows from a crucial result on stability of stochastic approximations given in [13].

Proposition 1: With $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi, \forall n$, $r(n), n \geq 0$ governed according to (15) 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

$$\psi^*(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{X_{n+1}} r^*_{\theta,\Phi} - \phi^T_{X_n} r^*_{\theta,\Phi}, \quad (20)$$

$$\tilde{\psi}^*(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{\tilde{X}_{n+1}} r^*_{\theta,\Phi} - \phi^T_{X_n} r^*_{\theta,\Phi}. \quad (21)$$

In view of Proposition 1, one may analyze the following recursion in place of (16):

$$y_{n+1}(i) = y_n(i) + b(n)I^i_{n+1}\big(\psi^*(n+1) - \gamma\tilde{\psi}^*(n) - y_n(i)\big). \quad (22)$$

The ODE associated with recursion (16) can thus be seen to be

$$\dot{y}(t) = (I - \gamma P^\theta)^T D z^*_{\theta,\Phi} - y(t), \quad (23)$$

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

Proposition 2: Given $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi, \forall n$, the updates $y_n, n \geq 0$ governed by (16) are uniformly bounded almost surely and converge to $y^*_{\theta,\Phi} \stackrel{\triangle}{=} (I - \gamma P^\theta)^T D z^*_{\theta,\Phi}$ 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 θ_n to be a constant θ while analyzing the update (17). 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 Γ^1-operator, the iterates in (17) remain almost surely uniformly bounded. One can rewrite the recursion (17) as follows:

$$\Phi(n+1) = \Gamma^1\bigg(\Phi(n) + c(n)2\big(I - \Phi(n)\Phi(n)^T\big) y^*_{\theta,\Phi(n)}(r^*_{\theta,\Phi(n)})^T + \epsilon(n)\bigg) \quad (24)$$

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 1 and 2, it follows that $\epsilon(n) \to 0$ almost surely as $n \to \infty$.

Let for any continuous function $v : \mathcal{R}^P \to \mathcal{R}^P$,

$$\hat{\Gamma}^P_j(v(y)) = \lim_{0<\eta\to 0}\bigg(\frac{\Gamma^j(y + \eta v(y)) - y}{\eta}\bigg), \quad (25)$$

$j = 1, 2$.

Consider now the following ODE corresponding to the recursion (24):

$$\dot{\Phi}(t) = \hat{\Gamma}^{N\times K}_1(-\nabla F_\theta(\Phi(t))), \quad (26)$$

Note that in (26), $P = N \times K$. Let

$$\mathcal{K}^1 \stackrel{\triangle}{=} \{\Phi \in \mathcal{M} \mid \hat{\Gamma}^{N\times K}_1(\nabla F(\Phi)) = 0\}, \quad (27)$$

denote the set of equilibria of (26), i.e., Karush-Kuhn-Tucker points for F.

We now have the following result:

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

Now consider the slowest recursion, i.e., the θ update corresponding to (18). The ODE corresponding to (18) can be written as, (with $P = L$)

$$\dot{\theta}(t) = \hat{\Gamma}_2^L(-\nabla\rho(\theta)). \tag{28}$$

Let $\mathcal{K}^2 \stackrel{\triangle}{=} \{\theta \in \mathcal{R}^L \mid \hat{\Gamma}_2^L(\nabla\rho(\theta)) = 0\}$, the Karush-Kuhn-Tucker points of ρ.

We now have the following main result:

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

From section IV and the foregoing, the algorithm converges to the set of local minima of the true weighted value function.

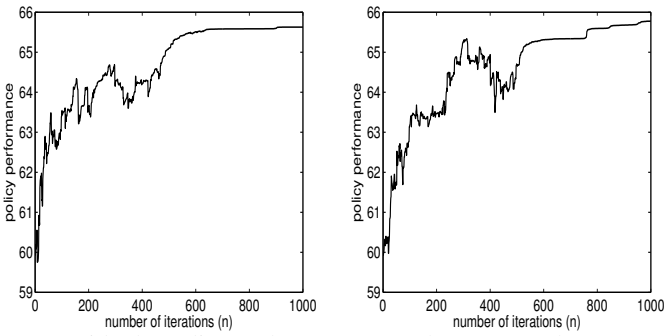
VII. Numerical Experiments

In this section, we demonstrate the performance of our algorithm as described in Section V and another algorithm where the residual gradient scheme is replaced by TD(0) on a random MDP setting with parameters $S = 100$, $A = 100$, $K = 10$ and $L = 100$. The MDP was randomly generated using a tool box. Here the transition and reward structure is arbitrarily set. We set the discount factor $\gamma = 0.9$. 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 $M = 100$. We let the step-sizes to be $a(n) = 1/n^{0.6}$, $b(n) = 1/n^{0.8}$ and $c(n) = 1/n$, respectively. In the plots, the Y-axis corresponds to policy performance $-\rho(\theta)$ (weighted discounted reward, i.e., negative of our objective) and X-axis corresponds to number of iterations. The policy features $\sigma(s, a)$ were randomly generated and fixed during the experiments.

Figs. $1(a)$ and $1(b)$ depict the results of using our algorithm as well as when TD(0) is used in place of the residual gradient scheme for the faster scale updates. 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 improve with the number of iterations. It can be seen from Figure1(b) 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.

VIII. 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 based on gradient search in the Grassmanian of features and SPSA. The algorithm is seen to perform well over a randomly generated MDP setting involving 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 TD(0) in place of the residual gradient scheme for 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.



(a) Residual gradient scheme (b) TD(0) scheme

Fig. 1. Plot of $-\rho(\theta)$ vs. n

References

[1] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *The Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.

[2] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of markov reward processes," IEEE Transactions on Automatic Control, Tech. Rep., 2001.

[3] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms." in *NIPS*, vol. 13. Citeseer, 1999, pp. 1008–1014.

[4] S. Bhatnagar, V. S. Borkar, and K. J. Prabuchandran, "Feature search in the grassmanian in online reinforcement learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 746–758, 2013.

[5] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *Automatic Control, IEEE Transactions on*, vol. 37, no. 3, pp. 332–341, 1992.

[6] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth, *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*. Springer, 2013.

[7] I. Menache, S. Mannor, and N. Shimkin, "Basis function adaptation in temporal difference reinforcement learning," *Annals of Operations Research*, vol. 134, no. 1, pp. 215–238, 2005.

[8] H. Yu and D. P. Bertsekas, "Basis function adaptation methods for cost approximation in mdp," in *Adaptive Dynamic Programming and Reinforcement Learning*. IEEE, 2009, pp. 74–81.

[9] S. Bhatnagar, V. S. Borkar, and L. Prashanth, "Adaptive feature pursuit: Online adaptation of features in reinforcement learning," *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pp. 517–534, 2012.

[10] D. Di Castro and S. Mannor, "Adaptive bases for reinforcement learning," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 312–327.

[11] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[12] L. Baird *et al.*, "Residual algorithms: Reinforcement learning with function approximation," in *ICML*, 1995, pp. 30–37.

[13] V. S. Borkar, "Stochastic approximation: a dynamical systems viewpoint," *Cambridge University Press*, 2008.