A Model based Search Method for Prediction in Model-free Markov Decision Process

Ajin George Joseph
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore
Email: ajin@csa.iisc.ernet.in

Shalabh Bhatnagar
Dept. of Computer Science and Automation &
Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science, Bangalore
Email: shalabh@csa.iisc.ernet.in

***Abstract*—In this paper, we provide a new algorithm for the problem of prediction in the model-free MDP setting, *i.e.*, estimating the value function of a given policy using the linear function approximation architecture, with memory and computation costs scaling quadratically in the size of the feature set. The algorithm is a multi-timescale variant of the very popular cross entropy (CE) method which is a model based search method to find the global optimum of a real-valued function. This is the first time a model based search method is used for the prediction problem. A proof of convergence using the ODE method is provided. The theoretical results are supplemented with experimental comparisons. The algorithm achieves good performance fairly consistently on many benchmark problems.**

I. Introduction and Preliminaries

In this paper, we follow the Markov decision process (MDP) framework as described in [1], [2]. We consider a discrete time MDP which is a 4-tuple ($\mathbb{S}$, $\mathbb{A}$, R, P), where $\mathbb{S}$ denotes the set of *states* and $\mathbb{A}$ is the set of *actions*. $\mathrm{R} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$ is the *reward function*, where $\mathrm{R}(s, a, s')$ represents the reward obtained in state s after taking action a and transitioning to s'. In this paper, we assume that the reward function is bounded, *i.e.*, $|\mathrm{R}(.,.,.)| \leq \mathrm{R}_{\max} < \infty$. Also, $\mathrm{P} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to [0, 1]$ is the *transition probability kernel*, where $\mathrm{P}(s, a, s')$ is the probability of next state being s' conditioned on the fact that the current state is s and action taken is a. We assume that the state and action spaces are finite. A *stationary policy* $\pi : \mathbb{S} \to \mathbb{A}$ is a function from states to actions, where $\pi(s)$ is the action taken in state s. A given policy π along with the transition kernel P determines the system dynamics, where the system behaves as a homogeneous Markov chain with transition matrix $\mathrm{P}^\pi(s, s') = \mathrm{P}(s, \pi(s), s')$. The policy can also be stochastic, where given $s \in \mathbb{S}$, $\pi(\cdot|s)$ is a probability measure over the action space $\mathbb{A}$.

For a given policy π, the system evolves at each discrete time step and this evolutionary process can be captured as a sequence of triplets $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t), t \geq 0\}$, where the random variable $\mathbf{s}_t$ represents the state at time t, $\mathbf{s}'_t$ is the transitioned state from $\mathbf{s}_t$ and $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}'_t)$ is the reward associated with the stochastic transition. In this paper, we are concerned with the problem of *prediction*, *i.e.*, estimating the long run γ-discounted cost $V^\pi \in \mathbb{R}^{|\mathbb{S}|}$ (also called the *value function*) corresponding to the policy π. Here, given $s \in \mathbb{S}$, we let

$$V^\pi(s) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \big| \mathbf{s}_0 = s\right], \tag{1}$$

where the constant $\gamma \in [0, 1)$ is called the *discount factor* and $\mathbb{E}[\cdot]$ is the expectation over sample trajectories obtained in turn from P^π when starting from the initial state s. V^π satisfies the well known *Bellman equation* which is given by

$$V^\pi = \mathrm{R}^\pi + \gamma \mathrm{P}^\pi V^\pi \triangleq T^\pi V^\pi, \tag{2}$$

where $\mathrm{R}^\pi \triangleq (\mathrm{R}^\pi(s), s \in \mathbb{S})^\top$ with $\mathrm{R}^\pi(s) = \mathbb{E}[\mathbf{r}_t | \mathbf{s}_t = s]$, $V^\pi \triangleq (V^\pi(s), s \in \mathbb{S})^\top$ and $T^\pi V^\pi \triangleq ((T^\pi V^\pi)(s),\ s \in \mathbb{S})^\top$, respectively. Here T^π is called the *Bellman operator*. If the model information, *i.e.*, P^π and R^π are available, then we can obtain the value function V^π by solving analytically the linear system $V^\pi = (\mathbb{I} - \gamma \mathrm{P}^\pi)^{-1} \mathrm{R}^\pi$.

A. Model-free MDP Setting

In this paper, we follow the usual model-free MDP setting, where we assume that the model is inaccessible; only a sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^{\infty}$ is available, where at each instant t, state $\mathbf{s}_t$ of the triplet $(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)$ is sampled using an arbitrary probability distribution ν over $\mathbb{S}$ which is called the *sampling distribution*. The transitioned state $\mathbf{s}'_t$ is drawn using $\mathrm{P}^\pi(\mathbf{s}_t, \cdot)$ and $\mathbf{r}_t$ is the immediate reward for the transition. The value function V^π has to be estimated from the sample trajectory.

To further make the problem more arduous, we consider here settings, where the state space is huge. The ensuing combinatorial blow-ups exemplify the underlying problem with the value function estimation, commonly referred to as the *curse of dimensionality*. In this case, the value function is unrealizable due to both storage and computational limitations. A common approach in this context is the function approximation method, specifically, the *linear function approximation technique* [1], [2], where a linear architecture consisting of a set of k, $|\mathbb{S}|$-dimensional feature vectors, $1 \leq k \ll |\mathbb{S}|$, $\{\phi_i \in \mathbb{R}^{\mathbb{S}}\}$, $1 \leq i \leq k$, is chosen *a priori*. For a state $s \in \mathbb{S}$, we define $\phi(s) \triangleq [\phi_1(s), \phi_2(s) \ldots \phi_k(s)]^\top$ and $\Phi \triangleq [\phi(s_1)^\top, \phi(s_2)^\top \ldots \phi(s_{|\mathbb{S}|})^\top]$, where the $k \times 1$ vector $\phi(\cdot)$ is called the *feature vector*, while the $|\mathbb{S}| \times k$ matrix Φ is called the *feature matrix*.

Given Φ, the best approximation of V^π is its projection on to the closed subspace $\{\Phi z | z \in \mathbb{R}^k\}$ with respect to an

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

arbitrary norm. Typically, one uses the weighted semi-norm $\|.\|_\nu$, where $\nu(\cdot)$ is an arbitrary probability distribution over $\mathbb{S}$. The most common choice for ν is the stationary distribution of P^π. The semi-norm $\|.\|_\nu$ and its associated linear projection operator Π^ν are defined as follows:

$$\|V\|_\nu^2 = \sum_{i=1}^{|\mathbb{S}|} V(i)^2 \nu(i), \;\; \Pi^\nu = \Phi(\Phi^\top D^\nu \Phi)^{-1} \Phi^\top D^\nu, \quad (3)$$

where D^ν is the diagonal matrix with $D^\nu_{ii} = \nu(i), i = 1, \ldots, |\mathbb{S}|$. A familiar objective in most linear function approximation algorithms is to find a vector $z^* \in \mathbb{R}^k$ such that $\Phi z^* \approx \Pi^\nu V^\pi$. Also note that the projection is obtained by minimizing the squared ν-weighted distance from the true value function V^π. This distance is referred to as the *mean squared error (MSE)* which is defined as follows:

$$\mathrm{MSE}(z) = \|V^\pi - \Phi z\|_\nu^2, \quad z \in \mathbb{R}^k. \quad (4)$$

Thus $\Pi^\nu V^\pi = \arg\min_{z \in \mathbb{R}^k} \mathrm{MSE}(z)$.

B. Background on Existing Algorithms

TD(0) algorithm with function approximation [3] is one of the fundamental prediction algorithm. TD(0) is an online, incremental algorithm, where at each discrete time t, the weight vectors are adjusted to better approximate the target value function. Van Roy and Tsitsiklis [3] gave a proper characterization for the limit point of TD(0) as the fixed point of the *projected Bellman operator* $\Pi^\nu T^\pi$, *i.e.*, the limit point of TD(0) satisfies $\Phi z = \Pi^\nu T^\pi \Phi z$. This characterization yields an error function called the *mean squared projected Bellman error (MSPBE)* which is defined as follows:

$$\mathrm{MSPBE}(z) \triangleq \|\Phi z - \Pi^\nu T^\pi \Phi z\|_\nu^2, \quad z \in \mathbb{R}^k. \quad (5)$$

In [4], MSPBE is maneuvered to derive stable algorithms like *TDC* and *GTD2*. Another pertinent error function is the *mean square Bellman residue (MSBR)* which is defined as

$$\mathrm{MSBR}(z) \triangleq \mathbb{E}\left[(\mathbb{E}\left[\delta_t(z)|\mathbf{s}_t\right])^2\right], z \in \mathbb{R}^k. \quad (6)$$

MSBR is a measure of how closely the prediction vector represents the solution to the Bellman equation (2). *Residual gradient (RG)* algorithm [5] minimizes the error function MSBR directly using stochastic gradient search. RG however requires *double sampling*, *i.e.*, generating two independent samples $\mathbf{s}'_t$ and $\mathbf{s}''_t$ of the next state when in the current state $\mathbf{s}_t$. Even though RG algorithm guarantees convergence, due to large variance, the convergence rate is small.

An important aspect of the prediction method one is most concerned is the stability. An iterative procedure is said to be stable if the iterates generated by the procedure converge. The stability of the TD(0) method is guaranteed under restricted settings, where the Markov chain is assumed ergodic and the sampling distribution is the stationary distribution of the Markov chain [3]. However, LSTD, LSPE, GTD2 and RG are shown to be stable under more general conditions [6].

Put succinctly, when linear function approximation is applied in a model-free MDP setting, the prediction task can be cast as an optimization problem whose objective function is one of the aforementioned error functions. Typically, almost all the state-of-the-art algorithms employ gradient search technique to solve the minimization problem. In this paper, we apply a model based search method to solve the prediction problem. *Model based search methods* [7] are a class of zero-order optimization algorithms, where the search is conducted in a space of parametrized probability models with the goal to find the unique discrete probability measure whose entire mass is uniformly distributed on the set of global optima (assumed finite) of the objective function. Prominent algorithms of this type include *cross entropy (CE) method*, MRAS [8] and EDA [9]. The unique characteristic of the model based methods is its non-dependency on the structural properties of the objective function and are referred to as *gradient-free* methods, where the algorithms do not incorporate information on the gradient or higher order derivatives of the objective function, rather use the function values themselves to guide the search.

Model based search methods have been applied to the control problem in [10], [11] and in basis adaptation[12], but this is the first time such a procedure has been applied to the prediction problem. Model based search methods operate offline and they assume that estimates or true objective function values can be obtained without much hardness or delay. However, in a dynamic model-free MDP setting, the transitions are asynchronous and delay between the transitions is random. The learning algorithm is further required to learn and evolve as new transitions are revealed. In such scenarios, the model-based search methods due to its offline nature, are not an appropriate choice. In this paper, we propose an online incremental version of the cross entropy method which is a widely recognized model based search method. We further adapt it to operate in a dynamic model-free MDP setting, where we propose a stable and efficient solution to the prediction problem.

The proposed algorithm possesses the following attractive characteristics: (1) There is minimal restriction on the feature set (2) The computational complexity is quadratic in the number of features (3) It is empirically shown to be competitive with other state-of-the-art algorithms in terms of accuracy (4) It is online with incremental updates (5) It gives guaranteed convergence to the global minimum of the MSPBE.

Summary of Notation: We use $\mathbf{x}$,$\mathbf{z}$ for random variable and x,z for deterministic variable. For set A, $\mathbb{I}_A$ represents the indicator function of A, *i.e.*, $\mathbb{I}_A(x) = 1$ if $x \in A$ and 0 otherwise. Let $f_\theta(\cdot)$ denote the *probability density function* (PDF) parametrized by θ. Let $\mathbb{E}_\theta[\cdot]$ and P_θ denote the *expectation* and the *probability measure* w.r.t. f_θ. For $\rho \in (0,1)$ and $H : \mathbb{R}^n \to \mathbb{R}$, let $\gamma_\rho(H(\cdot), \theta)$ denote the $(1-\rho)$-quantile of $H(\mathbf{x})$ w.r.t. f_θ, *i.e.*, $\gamma_\rho(H(\cdot), \theta) \triangleq \sup\{l : P_\theta(H(\mathbf{x}) \geq l) \geq \rho\}$. Let $int(A)$ be the *interior* of set A. Let $\mathcal{N}_n(m, V)$ denotes the n-variate Gaussian distribution with mean m and covariance V.

C. CE Method

The *cross entropy (CE) method* [13], [14], [15] solves problems of the following form: *find* $x^* \in \arg\max_{x \in \mathcal{X} \subset \mathbb{R}^k} \mathcal{H}(x)$, where $\mathcal{H} : \mathbb{R}^k \to \mathbb{R}$ is a multi-modal function and $\mathcal{X}$ is called the *solution space*. The goal of the CE method is to find an optimal "*model*" or probability distribution over the solution space $\mathcal{X}$ which concentrates on the global optima of $\mathcal{H}$. The CE method aims to find a sequence of model parameters $\{\theta_t\}_{t \in \mathbb{N}}$, where $\theta_t \in \Theta$ and an increasing sequence of thresholds $\{\gamma_t\}_{t \in \mathbb{N}}$, where $\gamma_t \in \mathbb{R}$, with the property that the event $\{\mathcal{H}(\mathbf{x}) \geq \gamma_t\}$ is a high probability event with respect to the probability measure induced by the model parameter θ_t. By assigning greater weight to higher values of $\mathcal{H}$ at each iteration, the expected behaviour of the model sequence should improve. Usually, the threshold γ_{t+1} is taken as $\gamma_\rho(\mathcal{H}(\cdot), \theta_t)$, the $(1-\rho)$-quantile of $\mathcal{H}(\mathbf{x})$ w.r.t. the PDF f_{θ_t}, where $\rho \in (0,1)$ is set *a priori*. The most commonly used family of distributions (which defines Θ) is the *natural exponential family of distributions (NEF)*.

Natural Exponential Family of Distributions: These are denoted as $\mathcal{C} \triangleq \{f_\theta(x) = h(x)e^{\theta^\top \Gamma(x) - K(\theta)} \mid \theta \in \Theta \subset \mathbb{R}^d\}$, where $h : \mathbb{R}^k \longrightarrow \mathbb{R}$, $\Gamma : \mathbb{R}^k \longrightarrow \mathbb{R}^d$ and $K : \mathbb{R}^d \longrightarrow \mathbb{R}$. The Gaussian distribution with mean vector μ and the covariance matrix Σ belongs to $\mathcal{C}$. In this case,

$$f_\theta(x) = ((2\pi)^k |\Sigma|)^{-\frac{1}{2}} e^{-(x-\mu)^\top \Sigma^{-1}(x-\mu)/2}, \tag{7}$$

and one may let $h(x) = (2\pi)^{-k/2}$, $\Gamma(x) = (x, xx^\top)^\top$ and $\theta = (\Sigma^{-1}\mu, -(2\Sigma)^{-1})^\top$.

⊛ ***Assumption (A1)*:** The parameter space Θ is compact.

In this paper, we take Gaussian distribution as the preferred choice for f_θ. In this case, the model parameter is $\theta = (\mu, \Sigma)^\top$, where $\mu \in \mathbb{R}^k$ is the mean vector and $\Sigma \in \mathbb{R}^{k \times k}$ is the covariance matrix. Here, the CE method is an iterative procedure which starts with an initial value $\theta_0 = (\mu_0, \Sigma_0)^\top$ and at each iteration t, a new parameter θ_{t+1} is derived from the previous value θ_t as follows (from Section 4 of [8]):

$$\theta_{t+1} = \arg\max_{\theta \in \Theta} \mathbb{E}_{\theta_t} \left\{ S(\mathcal{H}(\mathbf{x})) \mathbb{I}_{\{\mathcal{H}(\mathbf{x}) \geq \gamma_{t+1}\}} \log f_\theta(\mathbf{x}) \right\}, \tag{8}$$

where S is positive and strictly monotonically increasing.

If the gradient w.r.t. θ of the objective function in (8) is equated to 0 and using (7) for f_θ, we obtain

$$\mu_{t+1} = \frac{\mathbb{E}_{\theta_t}\left[\mathrm{g}_1(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{t+1})\right]}{\mathbb{E}_{\theta_t}\left[\mathrm{g}_0(\mathcal{H}(\mathbf{x}), \gamma_{t+1})\right]} \triangleq \Upsilon_1(\mathcal{H}(\cdot), \theta_t, \gamma_{t+1}), \tag{9}$$

$$\Sigma_{t+1} = \frac{\mathbb{E}_{\theta_t}\left[\mathrm{g}_2(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{t+1}, \mu_{t+1})\right]}{\mathbb{E}_{\theta_t}\left[\mathrm{g}_0(\mathcal{H}(\mathbf{x}), \gamma_{t+1})\right]} \triangleq \Upsilon_2(\mathcal{H}(\cdot), \theta_t, \gamma_{t+1}, \mu_{t+1}), \tag{10}$$

where $\mathrm{g}_0(\mathcal{H}(x), \gamma) \triangleq S(\mathcal{H}(x)) \mathbb{I}_{\{\mathcal{H}(x) \geq \gamma\}}$,

$$\mathrm{g}_1(\mathcal{H}(x), x, \gamma) \triangleq S(\mathcal{H}(x)) \mathbb{I}_{\{\mathcal{H}(x) \geq \gamma\}} x, \tag{11}$$

$$\mathrm{g}_2(\mathcal{H}(x), x, \gamma, \mu) \triangleq S(\mathcal{H}(x)) \mathbb{I}_{\{\mathcal{H}(x) \geq \gamma\}} (x-\mu)(x-\mu)^\top.$$

Remark I.1. *The function $S(\cdot)$ is positive and strictly monotonically increasing and is used to account for the cases when the objective function $\mathcal{H}(x)$ takes negative values for some x. Note that in the expression of μ_{t+1} in (9), x is being weighted with $S(\mathcal{H}(x))$ in the region $\{x | \mathcal{H}(x) \geq \gamma_{t+1}\}$. Since the function S is positive and strictly monotonically increasing, the region where $\mathcal{H}(x)$ is higher (hence $S(\mathcal{H}(x))$ is also higher) is given more weight and hence μ_{t+1} concentrates in the region where $\mathcal{H}(x)$ takes higher values. In most general cases, we take $S(x) = exp(rx), r \in \mathbb{R}_+$.*

II. Proposed Algorithm

We propose an *algorithm to approximate the value function V^π with linear function approximation, where the prediction vector is obtained by minimizing the error function MSPBE by using a multi-timescale stochastic approximation variant of the CE method*. Since the CE method is a maximization algorithm, the objective function in the optimization problem here is the negative of MSPBE. We consider the following problem

$$\text{Find } \arg\max_{z \in \mathcal{Z} \subseteq \mathbb{R}^k} \mathcal{J}(z). \tag{12}$$

Here $\mathcal{J}(z) \triangleq -\text{MSPBE}(z)$. Here $\mathcal{Z}$ is the space of parameter values of the function approximator. For brevity, we define $z^* = \arg\max_{z \in \mathcal{Z}} \mathcal{J}(z)$ and $\mathcal{J}^* \triangleq \mathcal{J}(z^*)$. Note that $\mathcal{J}$ is a convex function [6] and hence z^* is well-defined.

⊛ ***Assumption (A2)*:** The solution space $\mathcal{Z}$ is compact.

In [4], a compact expression for MSPBE is given as follows:

$$\text{MSPBE}(z) = (\Phi^\top D^\nu (T^\pi V_z - V_z))^\top (\Phi^\top D^\nu \Phi)^{-1} (\Phi^\top D^\nu (T^\pi V_z - V_z)),$$

where $V_z = \Phi z$. Now $\Phi^\top D^\nu (T^\pi V_z - V_z)$ can be rewritten as

$$\begin{aligned}\Phi^\top D^\nu (T^\pi V_z - V_z) &= \mathbb{E}\Big[\mathbb{E}[\phi_t(\mathbf{r}_t + \gamma z^\top \phi_t' - z^\top \phi_t)|\mathbf{s}_t]\Big] \\ &= \mathbb{E}\Big[\mathbb{E}[\phi_t \mathbf{r}_t | \mathbf{s}_t]\Big] + \mathbb{E}\Big[\mathbb{E}[\phi_t(\gamma\phi_t' - \phi_t)^\top | \mathbf{s}_t]\Big] z, \end{aligned} \tag{13}$$

where $\phi_t \triangleq \phi(\mathbf{s}_t)$ and $\phi_t' \triangleq \phi(\mathbf{s}_t')$. We also have $\Phi^\top D^\nu \Phi = \mathbb{E}\left[\phi_t \phi_t^\top\right]$. Putting all together we get,

$$\begin{aligned}\text{MSPBE}(z) &= \Big(\mathbb{E}\Big[\mathbb{E}[\phi_t \mathbf{r}_t | \mathbf{s}_t]\Big] + \mathbb{E}\Big[\mathbb{E}[\phi_t(\gamma\phi_t' - \phi_t)^\top | \mathbf{s}_t]\Big] z\Big)^\top \\ &\quad (\mathbb{E}\left[\phi_t \phi_t^\top\right])^{-1} \Big(\mathbb{E}\Big[\mathbb{E}[\phi_t \mathbf{r}_t | \mathbf{s}_t]\Big] + \mathbb{E}\Big[\mathbb{E}[\phi_t(\gamma\phi_t' - \phi_t)^\top | \mathbf{s}_t]\Big] z\Big). \\ &= \left(\omega_*^{(0)} + \omega_*^{(1)} z\right)^\top \omega_*^{(2)} \left(\omega_*^{(0)} + \omega_*^{(1)} z\right), \end{aligned} \tag{14}$$

where $\omega_*^{(0)} \triangleq \mathbb{E}\Big[\mathbb{E}[\phi_t \mathbf{r}_t | \mathbf{s}_t]\Big]$, $\omega_*^{(1)} \triangleq \mathbb{E}\Big[\mathbb{E}[\phi_t(\gamma\phi_t' - \phi_t)^\top | \mathbf{s}_t]\Big]$ and $\omega_*^{(2)} \triangleq (\mathbb{E}\left[\phi_t \phi_t^\top\right])^{-1}$.

Note that in the above expression, we are able to decouple the parameter vector z and the stochastic component involving $\mathbb{E}[\cdot]$. This is advantageous since it enables us to estimate the stochastic component independent of the parameter vector z.

We have modeled our algorithm in the stochastic approximation (SA) framework. *Stochastic approximation algorithms* [16], [17] are a natural way of utilizing prior information. It does so by discounted averaging of the prior information and are usually expressed in the following form:

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_{t+1}\Delta\mathbf{z}_{t+1}, \tag{15}$$

where $\mathbf{z}_t \in \mathbb{R}^k$, $\Delta\mathbf{z}_{t+1} = q(\mathbf{z}_t) + \mathbb{M}_{t+1}$ is the *increment term*, $q : \mathbb{R}^k \to \mathbb{R}^k$ is a Lipschitz continuous function and $\{\mathbb{M}_t \in \mathbb{R}^k\}$ is a *martingale difference noise sequence*, *i.e.*, $\mathbb{M}_t$ is $\mathcal{F}_t$-measurable and integrable and $\mathbb{E}[\mathbb{M}_{t+1}|\mathcal{F}_t] = 0, \forall t$. Here $\{\mathcal{F}_t\}$ is a filtration, where the σ-field $\mathcal{F}_t = \sigma(\mathbf{z}_i, \mathbb{M}_i, 1 \leq i \leq t, \mathbf{z}_0)$. The step-size α_t satisfies $\Sigma_t\alpha_t = \infty$, $\Sigma_t\alpha_t^2 < \infty$.

An important extension of SA is the multi-timescale variant, where there are multiple stochastic recursions of the kind (15), each with possibly different step-sizes. The step-size defines the timescale of the particular recursion. So different step-sizes imply different timescales. If the increment terms are well-behaved and the step-sizes are compatible (compatibility relationship defined in Section 6.1, Chapter 6 of [16]), then the chain of recursions exhibits a well-defined asymptotic behaviour.

Recall that the stochastic variables of the ideal CE method are γ_t, Υ_1, Υ_2, θ_t and the objective function $\mathcal{J}$. In our approach, we track these variables independently using stochastic recursions of the kind (15). Thus we model our algorithm as a multi-timescale stochastic approximation algorithm which tracks the ideal CE method. We consider here the various stochastic recursions of our algorithm in detail.

Tracking the Objective Function $\mathcal{J}$: We follow an incremental, online approach, where the algorithm learns and evolves as new transitions of the sample trajectory are revealed. We also want our algorithm to learn without any additional traversal of the trajectory. For the minimization of MSPBE, we require the following assumption on the sample trajectory.

⊛ ***Assumption (A3)*:** A sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^{\infty}$, where $\mathbf{s}_t \sim \nu(\cdot)$, $\mathbf{s}'_t \sim \mathrm{P}^{\pi}(\mathbf{s}_t, \cdot)$ and $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}'_t)$ is available. Further, let ϕ_t, ϕ'_t, and $\mathbf{r}_t$ have uniformly bounded second moments. Also, $\mathbb{E}\left[\phi_t\phi_t^\top\right]$ is non-singular.

In (A3), the uniform boundedness of the second moments of ϕ_t, ϕ'_t, and $\mathbf{r}_t$ directly follows in the case of finite state space MDP. However, the non-singularity requirement of $\mathbb{E}\left[\phi_t\phi_t^\top\right]$ is strict and can be ensured by wisely choosing the feature set.

In the decoupled expression (14) of $\mathcal{J}(\cdot)$, the stochastic part can be identified by the tuple $\omega_* \triangleq (\omega_*^{(0)}, \omega_*^{(1)}, \omega_*^{(2)})^\top$. So if we can track ω_*, then it directly implies that we can track $\mathcal{J}(\cdot)$. In our algorithm, we track ω_* using the time dependent variable $\omega_t \triangleq (\omega_t^{(0)}, \omega_t^{(1)}, \omega_t^{(2)})^\top$, where $\omega_t^{(0)} \in \mathbb{R}^k$, $\omega_t^{(1)} \in \mathbb{R}^{k\times k}$ and $\omega_t^{(2)} \in \mathbb{R}^{k\times k}$. Here $\omega_t^{(i)}$ independently tracks $\omega_*^{(i)}$, $1 \leq i \leq 3$. The stochastic recursion to track ω_* is given in (28). The increment term $\Delta\omega_{t+1} \triangleq (\omega_{t+1}^{(0)}, \omega_{t+1}^{(1)}, \omega_{t+1}^{(2)})^\top$ used in this recursion is defined as follows:

$$\left.\begin{aligned} \triangle\omega_{t+1}^{(0)} &= \mathbf{r}_t\phi_t - \omega_t^{(0)}, \\ \triangle\omega_{t+1}^{(1)} &= \phi_t(\gamma\phi'_t - \phi_t)^\top - \omega_t^{(1)}, \\ \triangle\omega_{t+1}^{(2)} &= \mathbb{I}_{k\times k} - \phi_t\phi_t^\top\omega_t^{(2)}, \end{aligned}\right\} \tag{16}$$

Now we define the estimate of $\mathcal{J}(z)$ at time t as follows:

$$\bar{\mathcal{J}}(\omega_t, z) \triangleq -\left(\omega_t^{(0)} + \omega_t^{(1)}z\right)^\top \omega_t^{(2)}\left(\omega_t^{(0)} + \omega_t^{(1)}z\right). \tag{17}$$

Note that this is the same expression as (14) except for ω_t replacing ω_*. Since ω_t tracks ω_*, it is easily verifiable that $\bar{\mathcal{J}}(\omega_t, z)$ indeed tracks $\mathcal{J}(z)$ for a given $z \in \mathcal{Z}$.

Tracking $\gamma_\rho(\mathcal{J}, \theta)$: Note that the true objective function $\mathcal{J}$ is not realizable and hence we use the best available estimate of the true function $\mathcal{J}(\cdot)$ at time t, *i.e.*, $\bar{\mathcal{J}}(\omega_t, \cdot)$. Now we make use of the following lemma from [18]. The lemma provides a characterization of the $(1-\rho)$-quantile of a given real-valued function H w.r.t. to a given PDF f_θ.

Lemma II.1. *The $(1-\rho)$-quantile of a bounded real valued function $H(\cdot)$ $\left(\text{with } H(x) \in [H_l, H_u]\right)$ w.r.t. the PDF $f_\theta(\cdot)$ is reformulated as an optimization problem*

$$\gamma_\rho(H, \theta) = \underset{y\in[H_l, H_u]}{\arg\min}\ \mathbb{E}_\theta\left[\psi(H(\mathbf{x}), y)\right], \tag{18}$$

where $\psi(H(x), y) = (1-\rho)(H(x) - y)\mathbb{I}_{\{H(x)\geq y\}} + \rho(y - H(x))\mathbb{I}_{\{H(x)\leq y\}}$.

We utilize a stochastic gradient descent to solve the optimization problem (18), where we maintain a time-dependent variable γ_t to track $\gamma_\rho(\mathcal{J}, \cdot)$. The stochastic recursion to track $\gamma_\rho(\mathcal{J}, \cdot)$ is given in (29) and the increment term in the recursion is the sub-differential $\nabla_y\psi$ which is given by

$$\Delta\gamma_{t+1}(z) = -(1-\rho)\mathbb{I}_{\{\bar{\mathcal{J}}(\omega_t, z)\geq\gamma_t\}} + \rho\mathbb{I}_{\{\bar{\mathcal{J}}(\omega_t, z)\leq\gamma_t\}} \tag{19}$$

We assume the following stability condition.

⊛ ***Assumption (A4)*:** The iterate sequence $\{\gamma_t\}$ in equation (29) satisfies $\sup_t|\gamma_t| < \infty$ *a.s.*.

(A4) is a technical requirement to ensure convergence. The solution space $\mathcal{Z}$ is assumed to be compact (by (A2)) which ensures the boundedness of $\mathcal{J}$. Hence the range of $\mathcal{J}$ is contained in a convex compact subset of $\mathbb{R}$. In practice, one can easily ensure assumption (A4) by projecting the iterates γ_t back to the compact convex set containing the range of $\mathcal{J}$.

Tracking Υ_1 and Υ_2: In the ideal CE method, for a given θ_t, recall that $\Upsilon_1(\ldots\theta_t\ldots)$ and $\Upsilon_2(\ldots\theta_t\ldots)$ form the subsequent model parameter θ_{t+1}. We employ two dependent stochastic recursions to track the above quantities. We maintain two time-dependent variables $\xi_t^{(0)}$ and $\xi_t^{(1)}$ to track Υ_1 and Υ_2 respectively. The respective stochastic recursions are given in (30) and (31) and the increment functions used in these recursions are defined as follows:

$$\Delta\xi_{t+1}^{(0)}(z) = \mathrm{g}_1(\bar{\mathcal{J}}(\omega_t, z), z, \gamma_t) - \xi_t^{(0)}\mathrm{g}_0(\bar{\mathcal{J}}(\omega_t, z), \gamma_t),$$
$$\Delta\xi_{t+1}^{(1)}(z) = \mathrm{g}_2(\bar{\mathcal{J}}(\omega_t, z), z, \gamma_t, \xi_t^{(0)}) - \xi_t^{(1)}\mathrm{g}_0(\bar{\mathcal{J}}(\omega_t, z), \gamma_t).$$

Model Parameter Update: In the ideal CE, recall that we have the discrete change $\theta_{t+1} = (\Upsilon_1(\dots\theta_t\dots), \Upsilon_2(\dots\theta_t\dots))^\top$. But in our algorithm, we adopt a smooth update of the model parameters. The recursion is defined in equation (33). The smoothed approach prevents premature convergence of the model sequence to any of the suboptimal solutions.

Step-sizes and Timescales: The algorithm uses two step-sizes α_t and β_t which are deterministic, positive, non-increasing and satisfy the following conditions:

$$\sum_{t=1}^{\infty}\alpha_t = \sum_{t=1}^{\infty}\beta_t = \infty,\ \sum_{t=1}^{\infty}\left(\alpha_t^2+\beta_t^2\right) < \infty,\ \lim_{t\to\infty}\frac{\alpha_t}{\beta_t} = 0. \quad (20)$$

In a multi-timescale stochastic approximation setting, it is important to understand the difference between timescale and step-size. The timescale of a stochastic recursion is defined by its step-size. Since α_t decays to 0 faster than β_t, the timescale obtained from β_t is considered faster as compared to the other. So in a multi-timescale stochastic recursion scenario, the evolution of the recursions controlled by the faster step-sizes (converges faster to 0) is slower compared to the recursions controlled by the slower step-sizes. Therefore, the faster timescale recursions converge faster compared to their slower counterparts. In our algorithm, the recursions of ω_t and θ_t proceed along the slowest timescale, while the recursions of $\gamma_t, \xi_t^{(0)}$ and $\xi_t^{(1)}$ proceed along the faster timescale.

Sample Requirement: The streamline nature inherent in the stochastic approximation algorithms demands only a single sample per iteration. In fact, we use two samples $\mathbf{z}_{t+1}$ (generated in (27)) and $\mathbf{z}_{t+1}^p$ (generated in (32)).

Mixture Distribution: In the algorithm, we use a mixture distribution $\widehat{f}_{\theta_t}$ to generate the sample $\mathbf{z}_{t+1}$, where $\widehat{f}_{\theta_t} = (1-\lambda)f_{\theta_t} + \lambda f_{\theta_0}$ with λ the mixing weight and $\lambda \in [0,1)$. The initial distribution parameter θ_0 is chosen *s.t.* the PDF f_{θ_0} is strictly positive on every point in the solution space $\mathcal{Z}$, *i.e.*, $f_{\theta_0}(z) > 0, \forall z \in \mathcal{Z}$. The mixture approach facilitates exploration of the solution space and prevents the iterates from getting stranded in suboptimal solutions.

The proposed algorithm which minimizes MSPBE is formally presented in Algorithm 1. The algorithm is named SCE-MSPBEM which is an acronym for *stochastic cross entropy mean squared projected Bellman error minimization*.

III. Convergence Analysis

Lemma III.1. *Let the step-sizes α_t and β_t satisfy (20). For the sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}_t')\}_{t=0}^{\infty}$, we let assumption (A3) hold and let ν be the sampling distribution. Then, for a given $z \in \mathcal{Z}$, the iterates ω_t in equation (28) satisfy,*

$$\lim_{t\to\infty}(\omega_t^{(0)} + \omega_t^{(1)} z) = \omega_*^{(0)} + \omega_*^{(1)} z \ \ a.s.,$$
$$\lim_{t\to\infty}\omega_t^{(2)} = \omega_*^{(2)} \ \ a.s. \ \text{and} \ \lim_{t\to\infty}\bar{\mathcal{J}}(\omega_t, z) = \mathcal{J}(z) \ \ a.s.,$$

where $\bar{\mathcal{J}}(\omega_t, z)$ is defined in (17) and $\mathcal{J}(z)$ in (12).

Proof. By rearranging equations in (28), for $t \in \mathbb{N}$, we get

$$\omega_{t+1}^{(0)} = \omega_t^{(0)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,0)} + h^{(0,0)}(\omega_t^{(0)})\big), \quad (21)$$

where $\mathbb{M}_{t+1}^{(0,0)} = \mathbf{r}_t\phi_t - \mathbb{E}\left[\mathbf{r}_t\phi_t\right]$ and $h^{(0,0)}(x) = \mathbb{E}\left[\mathbf{r}_t\phi_t\right] - x$.

$$\text{Similarly, } \omega_{t+1}^{(1)} = \omega_t^{(1)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,1)} + h^{(0,1)}(\omega_t^{(1)})\big), \quad (22)$$

where $\mathbb{M}_{t+1}^{(0,1)} = \phi_t(\gamma\phi_t' - \phi_t)^\top - \mathbb{E}\left[\phi_t(\gamma\phi_t' - \phi_t)^\top\right]$ and $h^{(0,1)}(x) = \mathbb{E}\left[\phi_t(\gamma\phi_t' - \phi_t)^\top\right] - x$.

$$\text{Finally, } \omega_{t+1}^{(2)} = \omega_t^{(2)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,2)} + h^{(0,2)}(\omega_t^{(2)})\big), \quad (23)$$

where $\mathbb{M}_{t+1}^{(0,2)} = \mathbb{E}\left[\phi_t\phi_t^\top\omega_t^{(2)}\right] - \phi_t\phi_t^\top\omega_t^{(2)}$ and $h^{(0,2)}(x) = \mathbb{I}_{k\times k} - \mathbb{E}\left[\phi_t\phi_t^\top x\right]$. It is easy to verify that $h^{(0,i)}, 0 \le i \le 2$ are Lipschitz continuous and $\{\mathbb{M}_{t+1}^{(0,i)}\}_{t\in\mathbb{N}}$, $0 \le i \le 2$ are martingale difference noise terms.

Since ϕ_t, ϕ_t' and $\mathbf{r}_t$ have uniformly bounded second moments,

$$\exists K_{0,i} > 0 \ \textit{s.t.}\ \mathbb{E}\left[\|\mathbb{M}_{t+1}^{(0,i)}\|^2|\mathcal{F}_t\right] \le K_{0,i}(1+\|\omega_t^{(i)}\|^2), t \ge 0.$$

Also $h_c^{(0,0)}(x) \triangleq \frac{h^{(0,0)}(cx)}{c} = \frac{\mathbb{E}[\mathbf{r}_t\phi_t|\mathcal{F}_t]-cx}{c} = \frac{\mathbb{E}[\mathbf{r}_t\phi_t|\mathcal{F}_t]}{c} - x$. So $h_\infty^{(0,0)}(x) = \lim_{t\to\infty} h_c^{(0,0)}(x) = -x$. Since the ODE $\dot{x}(t) = h_\infty^{(0,0)}(x)$ is globally asymptotically stable to the origin, we obtain that the iterates $\{\omega_t^{(0)}\}_{t\in\mathbb{N}}$ are almost surely stable, *i.e.*, $\sup_t\|\omega_t^{(0)}\| < \infty$ *a.s.*, from Theorem 7, Chapter 3 of [16]. Similarly we can show that $\sup_t\|\omega_t^{(1)}\| < \infty$ *a.s.*

$$\text{Now define } h_c^{(0,2)}(x) \triangleq \frac{h^{(0,2)}(cx)}{c} = \frac{\mathbb{I}_{k\times k} - \mathbb{E}\left[\phi_t\phi_t^\top cx|\mathcal{F}_t\right]}{c}.$$

Hence $h_\infty^{(0,2)}(x) = \lim_{t\to\infty} h_c^{(0,2)}(x) = -x\mathbb{E}\left[\phi_t\phi_t^\top\right]$. The ∞-system ODE given by $\dot{x}(t) = h_\infty^{(0,2)}(x)$ is also globally asymptotically stable to the origin since $\mathbb{E}[\phi_t\phi_t^\top]$ is positive definite (as it is non-singular and positive semi-definite). So $\sup_t\|\omega_t^{(2)}\| < \infty$ *a.s.* from Theorem 7, Chapter 3 of [16].

Now consider the following ODEs associated with (21)-(23):

$$\frac{d}{dt}\omega^{(0)}(t) = \mathbb{E}\left[\mathbf{r}_t\phi_t\right] - \omega^{(0)}(t), \qquad t \in \mathbb{R}_+, \ (24)$$

$$\frac{d}{dt}\omega^{(1)}(t) = \mathbb{E}\left[\phi_t(\gamma\phi_t' - \phi_t)^\top\right] - \omega^{(1)}(t)), \qquad t \in \mathbb{R}_+, \ (25)$$

$$\frac{d}{dt}\omega^{(2)}(t) = \mathbb{I}_{k\times k} - \mathbb{E}\left[\phi_t\phi_t^\top\right]\omega^{(2)}(t), \qquad t \in \mathbb{R}_+. \ (26)$$

For the ODE (24), the point $\mathbb{E}\left[\mathbf{r}_t\phi_t\right]$ is a globally asymptotically stable equilibrium. Similarly for the ODE (25), the point $\mathbb{E}\left[\phi_t(\gamma\phi_t' - \phi_t)^\top\right]$ is a globally asymptotically stable equilibrium. For the ODE (26), since $\mathbb{E}\left[\phi_t\phi_t^\top\right]$ is non-negative definite and non-singular (from assumption (A3)), the ODE (26) is globally asymptotically stable to the point $\mathbb{E}\left[\phi_t\phi_t^\top\right]^{-1}$.

It can now be shown from Theorem 2, Chapter 2 of [16] that the asymptotic properties of the recursions (21), (22), (23) and their associated ODEs (24), (25), (26) are similar. Hence

$$\lim_{t\to\infty}\omega_t^{(0)} = \mathbb{E}\left[\mathbf{r}_t\phi_t\right] \ a.s. \ = \omega_*^{(0)}.$$
$$\lim_{t\to\infty}\omega_t^{(1)} = \mathbb{E}\left[\phi_t(\gamma\phi_t' - \phi_t)^\top\right] \ a.s. \ = \omega_*^{(1)}.$$
$$\lim_{t\to\infty}\omega_{t+1}^{(2)} = \mathbb{E}\left[\phi_t\phi_t^\top\right]^{-1} \ a.s. \ = \omega_*^{(2)}.$$

It easily follows that $\bar{\mathcal{J}}(\omega_t, z) \to \bar{\mathcal{J}}(\omega_*, z) = \mathcal{J}(z)$ *a.s.*

□

Algorithm 1: SCE-MSPBEM

Data: $\alpha_t, \beta_t, \lambda \in [0,1)$, $\epsilon_1 \in (0,1)$, $c_t \in (0,1)$, $c_t \to 0$;

Initialization: $t=0$, $\gamma_0=0$, $\gamma_0^p=-\infty$, $\theta_0=(\mu_0,\Sigma_0)^\top$, $T_0=0$, $\xi_t^{(0)}=0_{k\times 1}$, $\xi_t^{(1)}=0_{k\times k}$, $\omega_0^{(0)}=0_{k\times 1}, \omega_0^{(1)}=0_{k\times k}, \omega_0^{(2)}=0_{k\times k}, \theta^p=NULL$;

foreach $(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)$ *of the trajectory* **do**

$$\mathbf{z}_{t+1} \sim \widehat{f}_{\theta_t}(\cdot) \text{ where } \widehat{f}_{\theta_t} \triangleq (1-\lambda) f_{\theta_t} + \lambda f_{\theta_0}; \tag{27}$$

► **[*Objective Function Evaluation*]**

$$\omega_{t+1} = \omega_t + \alpha_{t+1}\Delta\omega_{t+1}; \tag{28}$$

$$\bar{\mathcal{J}}(\omega_t, \mathbf{z}_{t+1}) = -\left(\omega_t^{(0)} + \omega_t^{(1)}\mathbf{z}_{t+1}\right)^\top \omega_t^{(2)} \left(\omega_t^{(0)} + \omega_t^{(1)}\mathbf{z}_{t+1}\right);$$

► **[*Threshold Evaluation*]**

$$\gamma_{t+1} = \gamma_t - \beta_{t+1}\Delta\gamma_{t+1}(\mathbf{z}_{t+1}); \tag{29}$$

► **[*Tracking Υ_1 and Υ_2 of (9) and (10)*]**

$$\xi_{t+1}^{(0)} = \xi_t^{(0)} + \beta_{t+1}\Delta\xi_{t+1}^{(0)}(\mathbf{z}_{t+1}); \tag{30}$$

$$\xi_{t+1}^{(1)} = \xi_t^{(1)} + \beta_{t+1}\Delta\xi_{t+1}^{(1)}(\mathbf{z}_{t+1}); \tag{31}$$

if $\theta^p \neq NULL$ **then**

$$\left.\begin{aligned} \mathbf{z}_{t+1}^p &\sim \widehat{f}_{\theta^p}(\cdot) \triangleq \lambda f_{\theta_0} + (1-\lambda) f_{\theta^p}; \\ \gamma_{t+1}^p &= \gamma_t^p - \beta_{t+1}\Delta\gamma_{t+1}(\mathbf{z}_{t+1}^p); \end{aligned}\right\} \tag{32}$$

► **[*Threshold Comparison*]**

$$T_{t+1} = T_t + c\left(\mathbb{I}_{\{\gamma_{t+1} > \gamma_{t+1}^p\}} - \mathbb{I}_{\{\gamma_{t+1} \leq \gamma_{t+1}^p\}} - T_t\right);$$

► **[*Updating Model Parameter*]**

if $T_{t+1} > \epsilon_1$ **then**

$$\theta^p = \theta_t;$$

$$\theta_{t+1} = \theta_t + \alpha_{t+1}\left((\xi_t^{(0)}, \xi_t^{(1)})^\top - \theta_t\right); \tag{33}$$

$$\gamma_{t+1}^p = \gamma_t; \quad T_t = 0; \quad c = c_t; \tag{34}$$

else

$$\gamma_{t+1}^p = \gamma_t^p; \quad \theta_{t+1} = \theta_t;$$

$t := t+1$;

We now state our main theorem. As a prerequisite to the theorem, we define $\Psi(\theta) = (\Psi_1(\theta), \Psi_2(\theta))^\top$, where

$$\Psi_1(\theta) \triangleq \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathrm{g}_1\left(\mathcal{J}(\mathbf{z}), \mathbf{z}, \gamma_\rho(\mathcal{J}, \widehat{\theta})\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathrm{g}_0(\mathcal{J}(\mathbf{z}), \gamma_\rho(\mathcal{J}, \widehat{\theta}))\right]}, \tag{35}$$

$$\Psi_2(\theta) \triangleq \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathrm{g}_2\left(\mathcal{J}(\mathbf{z}), \mathbf{z}, \gamma_\rho(\mathcal{J}, \widehat{\theta}), \Psi_1(\theta)\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathrm{g}_0\left(\mathcal{J}(\mathbf{z}), \gamma_\rho(\mathcal{J}, \widehat{\theta})\right)\right]}. \tag{36}$$

Theorem III.2. *Let $S(z) = exp(rz)$, $r \in \mathbb{R}_+$. Let $\rho \in (0,1)$, $\lambda \in [0,1)$. Let $\theta_0 = (\mu_0, qI_{k\times k})^\top$, where $q \in \mathbb{R}_+$. Let the step-sizes α_t, β_t satisfy (20). Also let $c_t \to 0$. Suppose $\{\theta_t = (\mu_t, \Sigma_t)^\top\}_{t\in\mathbb{N}}$ is the sequence generated by Algorithm 1 and assume $\theta_t \in int(\Theta)$, $\forall t \in \mathbb{N}$. Also, let the assumptions (A1), (A2), (A3) and (A4) hold. Further, we assume that there exists a continuously differentiable function $V : \Theta \to \mathbb{R}_+$ s.t. $\nabla V(\theta)^\top \Psi(\omega_*, \theta) < 0$, $\forall \theta \in \Theta \setminus \{\theta^*\}$ and $\nabla V(\theta^*)^\top \Psi(\omega_*, \theta^*) = 0$. Then, there exists $q^* \in \mathbb{R}_+$ and $r^* \in \mathbb{R}_+$ s.t. $\forall q > q^*$ and $\forall r > r^*$,*

$$\lim_{t\to\infty} \bar{\mathcal{J}}(\omega_t, \mu_t) = \mathcal{J}^* \quad \textit{and} \quad \lim_{t\to\infty} \theta_t = \theta^* = (z^*, 0_{k\times k})^\top,$$

where $\mathcal{J}^*$ and z^* are defined in (12).

IV. Experimental Results

We present here a numerical comparison of SCE-MSPBEM with various state-of-the-art algorithms in the literature on some benchmark problems. In each of the experiments, a sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^\infty$ is chosen by following (A3) and all the algorithms are updated using it. The algorithms are run on multiple trajectories and the average of the results obtained are plotted. The x-axis in the plots is $t/1000$, where t is the iteration number. The function $S(\cdot)$ is chosen as $S(x) = \exp(rx)$, where $r \in \mathbb{R}_+$ is chosen appropriately.

A. Experiment 1: Linearized Cart-Pole Balancing [6]

Setup: A pole with mass m and length l is connected to a cart of mass M . It can rotate 360° and the cart is free to move in either direction within the bounds of a linear track.
Goal: To balance the pole upright and the cart at the centre of the track.
State space: The 4-tuple $[x, \dot{x}, \psi, \dot{\psi}]$ where ψ is the angle of the pendulum *w.r.t.* the vertical axis, $\dot{\psi}$ is the angular velocity, x the relative cart position from the centre of the track and $\dot{x}$ is its velocity.
Control space: The controller applies a horizontal force a on the cart parallel to the track. The stochastic policy used in this setting corresponds to $\pi(a|s) = \mathcal{N}(a|\beta_1^\top s, \sigma_1^2)$.
System dynamics: The dynamical system equations are

$$\ddot{\psi} = \frac{-3ml\dot{\psi}^2 \sin\psi\cos\psi + (6M+m)g\sin\psi - 6(a-b\dot{\psi})\cos\psi}{4l(M+m) - 3ml\cos\psi},$$

$$\ddot{x} = \frac{-2ml\dot{\psi}^2\sin\psi + 3mg\sin\psi\cos\psi + 4a - 4b\dot{\psi}}{4(M+m) - 3m\cos\psi}.$$

By making assumptions on the initial conditions, the system dynamics can be approximated accurately by the linear system

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \\ \psi_{t+1} \\ \dot{\psi}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ \dot{x}_t \\ \psi_t \\ \dot{\psi}_t \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\psi}_t \\ \frac{3(M+m)\psi_t - 3a + 3b\dot{\psi}_t}{4Ml - ml} \\ \dot{x}_t \\ \frac{3mg\psi_t + 4a - 4b\dot{\psi}_t}{4M - m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{z} \end{bmatrix}, \tag{37}$$

where Δt is the integration time step, *i.e.*, the interval between two transitions and $\mathbf{z} \sim \mathcal{N}(0, \sigma_2)$ is a Gaussian noise.
Reward function: $\mathrm{R}(\psi, \dot{\psi}, x, \dot{x}, a) = -100\psi^2 - x^2 - \frac{1}{10}a^2$.
Feature vector: $\phi(s) = (1, s_1^2, s_2^2 \ldots, s_1 s_2 \ldots s_3 s_4)^\top \in \mathbb{R}^{11}$.
Evaluation policy: The policy evaluated in the experiment is the optimal policy $\pi^*(a|s) = \mathcal{N}(a|{\beta_1^*}^\top s, {\sigma_1^*}^2)$. The parameters β_1^* and σ_1^* are computed using dynamic programming. The feature set chosen above is a perfect feature set, *i.e.*, $V^{\pi^*} \in \{\Phi z | z \in \mathbb{R}^k\}$. The various parameter values we used in our experiment are given in Table I. The results of the experiments are shown in Figure 2.

B. Experiment 2: 5-Link Actuated Pendulum Balancing [6]

Setup: 5 poles each with mass m and length l with the top pole being a pendulum connected using 5 rotational joints.
Goal: To keep all the poles in the upright position by applying independent torques at each joint.

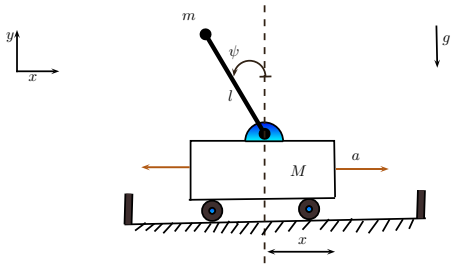
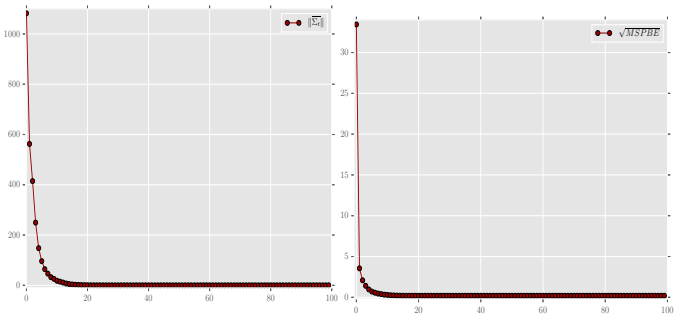


Fig. 1: The cart-pole system. The goal is to keep the pole in the upright position and the cart at the center of the track by applying a force a either to the left or the right.



(a) Plot of $\|\Sigma_t\|_F$ (where $\|\cdot\|_F$ is the Frobenius norm)

(b) Plot of $\sqrt{\text{MSPBE}(\mu_t)}$

Fig. 2: The cart-pole setting. The evolutionary trajectory of the variables $\|\Sigma_t\|_F$ and $\sqrt{\text{MSPBE}(\mu_t)}$. Note that $\sqrt{\text{MSPBE}(\mu_t)}$ converge to 0 as $t \to \infty$, while $\|\Sigma_t\|_F$ also converges to 0. This implies that the model $\theta_t = (\mu_t, \Sigma_t)^\top$ converges to the degenerate distribution concentrated on z^*.

State space: The state $s = (q, \dot{q})^\top$, where $q = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5) \in \mathbb{R}^5$ and $\dot{q} = (\dot{\psi}_1, \dot{\psi}_2, \dot{\psi}_3, \dot{\psi}_4, \dot{\psi}_5) \in \mathbb{R}^5$ with ψ_i the angle of the pole i *w.r.t.* the vertical axis and $\dot{\psi}_i$ is the angular velocity.
Control space: The action $a = (a_1, a_2, \ldots, a_5)^\top \in \mathbb{R}^5$ where a_i is the torque applied to the joint i. The stochastic policy used in this setting corresponds to $\pi(a|s) = \mathcal{N}(a|\beta_1^\top s, \sigma_1^2)$.
System dynamics: The approximate linear system dynamics is

$$\begin{bmatrix} q_{t+1} \\ \dot{q}_{t+1} \end{bmatrix} = \begin{bmatrix} I & \Delta t\, I \\ -\Delta t\, M^{-1}U & I \end{bmatrix} \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} + \Delta t \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} a + \mathbf{z},$$

where Δt is the integration time step, M is the mass matrix with $M_{ij} = l^2(6 - max(i,j))m$, U is a diagonal matrix with $U_{ii} = -gl(6-i)m$ and $\mathbf{z}$ is a Gaussian noise.

TABLE I: Parameter values used in the experiment 1

Gravitational constant (g)	$9.8\frac{m}{s^2}$
Mass of the pole (m)	$0.5kg$
Mass of the cart (M)	$0.5kg$
Length of the pole (l)	$0.6m$
Friction coefficient (b)	$0.1N(ms)^{-1}$
Integration time step (Δt)	$0.1s$
Standard deviation of z (σ_2)	0.01
Discount factor (γ)	0.95

α_t	$t^{-1.0}$
β_t	$t^{-0.6}$
c_t	0.01
λ	0.01
ϵ_1	0.95
ρ	0.1

TABLE II: Parameter values used in the experiment 2

Mass of the pole (m)	$1.0kg$
Length of the pole (l)	$1.0m$
Integration time step (Δt)	$0.1s$
Discount factor (γ)	0.95

α_t	0.001
β_t	0.05
c_t	0.05
λ	0.01
ϵ_1	0.95
ρ	0.1

Reward function: $\mathrm{R}(q, \dot{q}, a) = -q^\top q$.
Feature vector: $\phi(s) = (1, s_1^2, s_2^2 \ldots, s_1 s_2 \ldots s_9 s_{10})^\top \in \mathbb{R}^{46}$.
Evaluation policy: The policy evaluated in the experiment is the optimal policy $\pi^*(a|s) = \mathcal{N}(a|{\beta_1^*}^\top s, {\sigma_1^*}^2)$. The parameters β_1^* and σ_1^* are computed using dynamic programming. The feature set chosen above is a perfect feature set, *i.e.*, $V^{\pi^*} \in \{\Phi z | z \in \mathbb{R}^k\}$.

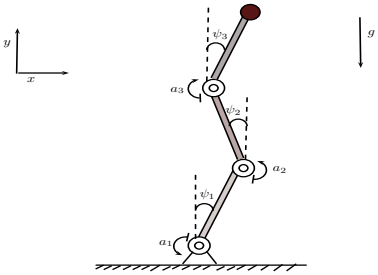
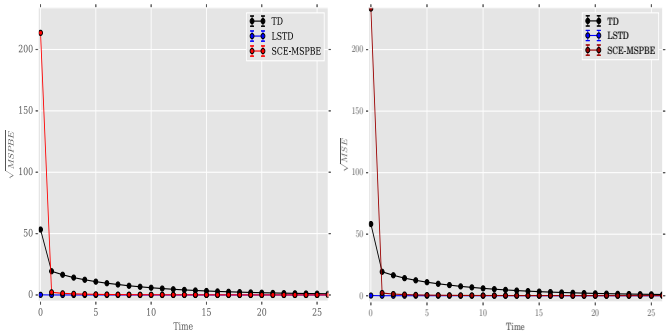


Fig. 3: 3-link actuated pendulum setting. Each rotational joint i, $1 \leq i \leq 3$ is actuated by a torque a_i . The system is parameterized by the vertical angle ψ_i and the angular velocity $\dot{\psi}_i$. The goal is to balance the pole in the upright direction, *i.e.*, all ψ_i should be close to 0.

The various parameter values used in our experiment are given in Table II. The results obtained are shown in Figure 4.



(a) $\sqrt{\text{MSPBE}(\mu_t)}$

(b) $\sqrt{\text{MSE}(\mu_t)}$

Fig. 4: 5-link actuated pendulum setting. The trajectories of the $\sqrt{\text{MSPBE}}$ and $\sqrt{\text{MSE}}$ generated by TD(0), LSTD and SCE-MSPBEM algorithms are plotted. Note that $\sqrt{\text{MSE}}$ converges to 0 since the feature set is perfect.

C. Experiment 3: Baird's 7-Star MDP [5]

Our algorithm was also tested on Baird's star problem [5]. We call it the stability test because the Markov chain in this

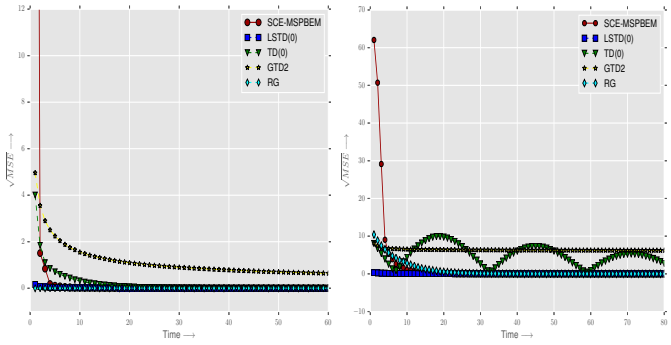
case is not ergodic and this is a classic example where TD(0) is seen to diverge [5]. We consider here an MDP with $|\mathbb{S}| = 7$, $|\mathbb{A}| = 2$ and $k = 8$. We let the sample distribution ν to be the uniform distribution over $\mathbb{S}$. The feature matrix Φ and the transition matrix P^π are given by

$$\Phi = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{P}^\pi = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The reward function is given by $\mathrm{R}(s, s') = 0$, $\forall s, s' \in \mathbb{S}$. The performance comparison of the algorithms GTD2, TD(0) and LSTD with SCE-MSPBEM is shown in Figure 5. The performance metric used here is the $\sqrt{MSE(\cdot)}$ of the prediction vector returned by the corresponding algorithm. The parameter values used in the experiment are given in Table III. A careful analysis in [19] has shown that when the discount factor $\gamma \leq 0.88$, with appropriate step-size, TD(0) converges. Nonetheless, it is also shown in the same paper that for discount factor $\gamma = 0.9$, TD(0) will diverge for all values of the step-size. This is explicitly demonstrated in Figure 5. However SCE-MSPBEM converges in both cases, which demonstrates the stable behaviour exhibited by our algorithm.

TABLE III: Parameter values for experiment 3

α_t	β_t	c_t	λ	ϵ_1	ρ
0.001	0.05	0.01	0.01	0.8	0.1



(a) Discount factor $\gamma = 0.1$ (b) Discount factor $\gamma = 0.9$

Fig. 5: Baird's 7-Star MDP with perfect feature set. For $\gamma = 0.1$, all the algorithms show almost the same rate of convergence. The initial jump of SCE-MSPBEM is due to the fact that the initial value is far from the limit. For $\gamma = 0.9$, TD(0) does not converge and GTD2 is slower. However, SCE-MSPBEM exhibits good convergence behaviour.

V. Conclusion

We propose a model based search method to the prediction problem in a model-free MDP under the linear function approximation architecture. This task is accomplished by remodeling the original CE method as a multi-timescale stochastic approximation algorithm and using it to minimize MSPBE. We also provide the proof of convergence of our algorithm to the optimal solution. The theoretical analysis is supplemented by extensive experimental comparisons with the state-of-the-art algorithms.

References

[1] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*. MIT Press, New York, USA, 1998.

[2] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, USA, 2013, vol. 2, no. 2.

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

[4] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th ICML*. ACM, 2009, pp. 993–1000.

[5] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Proceedings of the twelfth international conference on machine learning*, 1995, pp. 30–37.

[6] C. Dann, G. Neumann, and J. Peters, "Policy evaluation with temporal differences: A survey and comparison," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 809–883, 2014.

[7] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," *Annals of Operations Research*, vol. 131, no. 1-4, pp. 373–395, 2004.

[8] J. Hu, M. C. Fu, and S. I. Marcus, "A model reference adaptive search method for global optimization," *Operations Research*, vol. 55, no. 3, pp. 549–568, 2007.

[9] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *Parallel Problem Solving from NaturePPSN IV*. Springer, 1996, pp. 178–187.

[10] J. Hu, M. C. Fu, and S. I. Marcus, "A model reference adaptive search method for stochastic global optimization," *Communications in Information & Systems*, vol. 8, no. 3, pp. 245–276, 2008.

[11] S. Mannor, R. Y. Rubinstein, and Y. Gat, "The cross entropy method for fast policy search," in *ICML*, 2003, pp. 512–519.

[12] 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.

[13] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[14] J. Hu and P. Hu, "On the performance of the cross-entropy method," in *Simulation Conference (WSC), 2009 Winter*. IEEE, 2009, pp. 459–468.

[15] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

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

[17] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[18] T. Homem-de Mello, "A study on the cross-entropy method for rare-event probability estimation," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 381–394, 2007.

[19] R. Schoknecht and A. Merke, "Convergent combinations of reinforcement learning with linear function approximation," in *Advances in Neural Information Processing Systems*, 2002, pp. 1579–1586.