Mach Learn (2018) 107:1385–1429
https://doi.org/10.1007/s10994-018-5727-z

An online prediction algorithm for reinforcement learning with linear function approximation using cross entropy method

Ajin George Joseph[1,2] · Shalabh Bhatnagar[3]

Received: 4 November 2017 / Accepted: 8 June 2018 / Published online: 3 July 2018

Abstract In this paper, we provide two new stable online algorithms for the problem of prediction in reinforcement learning, i.e., estimating the value function of a model-free Markov reward process using the linear function approximation architecture and with memory and computation costs scaling quadratically in the size of the feature set. The algorithms employ the multi-timescale stochastic approximation variant of the very popular cross entropy optimization method which is a model based search method to find the global optimum of a real-valued function. A proof of convergence of the algorithms using the ODE method is provided. We supplement our theoretical results with experimental comparisons. The algorithms achieve good performance fairly consistently on many RL benchmark problems with regards to computational efficiency, accuracy and stability.

Keywords Markov decision process · Prediction problem · Reinforcement learning · Stochastic approximation algorithm · Cross entropy method · Linear function approximation · ODE method

1 Introduction

In this paper, we follow the reinforcement learning (RL) framework as described in Sutton and Barto (1998), White (1993), Bertsekas (2013). The basic structure in this setting is the discrete time Markov decision process (MDP) which is a 4-tuple $(\mathbb{S}, \mathbb{A}, \mathrm{R}, \mathrm{P})$, where $\mathbb{S}$ denotes

Editors: Jesse Davis, Elisa Fromont, Derek Greene, and Bjorn Bringmaan.

✉ Ajin George Joseph
ajin@iisc.ac.in

[1] Department of Computing Science, University of Alberta, Edmonton, Canada

[2] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

[3] Department of Computer Science and Automation and the Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore, India

the set of *states* and $\mathbb{A}$ is the set of *actions*. We assume that the state and action spaces are finite. The function $\mathrm{R} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$ is called the *reward function*, where $\mathrm{R}(s, a, s')$ represents the reward obtained in state s after taking action a and transitioning to s'. Without loss of generality, we assume that the reward function is bounded, *i.e.*, $|\mathrm{R}(\cdot, \cdot, \cdot)| \leq \mathrm{R}_{\max} < \infty$. Also, $\mathrm{P} : \mathbb{S}\times\mathbb{A}\times\mathbb{S} \to [0, 1]$ is the *transition probability kernel*, where $\mathrm{P}(s, a, s') = \mathbb{P}(s'|s, a)$ is the probability of next state being s' conditioned on the fact that the current state is s and the action taken is a. A *stationary policy* $\pi : \mathbb{S} \to \mathbb{A}$ is a function from states to actions, where $\pi(s)$ is the action taken whenever the system is in state s (independent of time).[1] A given policy π along with the transition kernel P determine the state dynamics of the system. For a given policy π, the system behaves as a Markov chain with transition matrix $\mathrm{P}^{\pi}(s, s') = \mathrm{P}(s, \pi(s), s')$.

For a given policy π, the system evolves at each discrete time step and this process can be captured as a coupled sequence of transitions and rewards $\{\mathbf{s}_0, \mathbf{r}_0, \mathbf{s}_1, \mathbf{r}_1, \mathbf{s}_2, \mathbf{r}_2, \ldots\}$, where $\mathbf{s}_t$ is the random variable which represents the state at time t, $\mathbf{s}_{t+1}$ is the transitioned state from $\mathbf{s}_t$ and $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}_{t+1})$ is the reward associated with the transition. In this paper, we are concerned with the problem of *prediction*, *i.e.*, estimating the expected long run γ-discounted cost $V^{\pi} \in \mathbb{R}^{\mathbb{S}}$ (also referred to as the *value function*) corresponding to the given policy π. Here, given $s \in \mathbb{S}$, we let

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

where $\gamma \in [0, 1)$ is a constant called the *discount factor* and $\mathbb{E}[\cdot]$ is the expectation over sample trajectories of states obtained in turn from P^{π} when starting from the initial state s. V^{π} satisfies the well known *Bellman equation* (Bertsekas 2013) under policy π, 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 $\mathrm{R}^{\pi}(s) = \mathbb{E}\left[\mathbf{r}_t | \mathbf{s}_t = s\right]$, $V^{\pi} \triangleq (V^{\pi}(s), s \in \mathbb{S})^{\top}$ 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*.

Prediction problem[2] (Sutton 1988; Maei et al. 2009; Sutton et al. 2009): In this paper, we follow a generalized RL framework, where we assume that the model, *i.e.*, P and R are inaccessible; only a sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^{\infty}$ is available where at each instant t, the state $\mathbf{s}_t$ of the triplet $(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)$ is sampled using an arbitrary distribution ν over $\mathbb{S}$ called the *sampling distribution*, while the next state $\mathbf{s}'_t$ is drawn using $\mathrm{P}^{\pi}(\mathbf{s}_t, \cdot)$ following the underlying Markov dynamics and $\mathbf{r}_t$ is the immediate reward for the transition, *i.e.*, $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}'_t)$. We assume that $\nu(s) > 0, \forall s \in \mathbb{S}$. The goal of the prediction problem is to estimate the value function V^{π} from the given sample trajectory.

Remark 1 The framework that we consider in this paper is a generalized setting, commonly referred to as the off-policy setting.[3] In the literature, one often finds the on-policy setting

[1] The policy can also be stochastic in order to incorporate exploration. In that case, for a given $s \in \mathbb{S}$, $\pi(\cdot|s)$ is a probability distribution over the action space $\mathbb{A}$.

[2] The prediction problem is related to policy evaluation except that the latter procedure evaluates the value of a policy given complete model information. We are however in an online setting where model information is completely unknown, however, a realization of the model dynamics in the form of a sample trajectory as described above is made available in an incremental fashion. The goal then is to predict at each time instant the value of each state in $\mathbb{S}$ (both observed and unobserved) under this constraint using the sample trajectory revealed till that instant.

[3] One may find the term off-policy to be a misnomer in this context. Usually on-policy refers to RL settings where the underlying Markovian system is assumed ergodic and the sample trajectory provided follows the

where the underlying Markovian system induced by the evaluation policy is assumed ergodic, *i.e.*, aperiodic and irreducible, which directly implies the existence of a unique steady state distribution (stationary distribution). In such cases, the sample trajectory is presumed to be a continuous roll-out of a particular instantiation of the underlying transition dynamics in the form of $\{\mathbf{s}_0, \mathbf{r}_1, \mathbf{s}_1, \mathbf{r}_2, \ldots\}$, where $\mathbf{s}_0$ is chosen arbitrarily. Since the system is ergodic, the distribution of the states in the sample trajectory will eventually follow the steady state distribution. Hence, the on-policy setting becomes a special case of the off-policy setting, where the sampling distribution is nothing but the stationary distribution and $\mathbf{s}_{t+1} = \mathbf{s}'_t$, $\forall t \in \mathbb{N}$.

Unfortunately, the number of states $|\mathbb{S}|$ may be large in many practical applications (Kaelbling et al. 1996; Doya 2000), for example, elevator dispatching (Crites and Barto 1996), robotics (Kober et al. 2013) and board games such as Backgammon (10^{20} states, Tesauro 1995) and computer Go (10^{170} states, Silver et al. 2007). The impending 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. Apparently one has to resort to approximate solution methods where we sacrifice precision for computational tractability. A common approach in this context is the function approximation method (Sutton and Barto 1998), where we approximate the value function of unobserved states using the knowledge of the observed states and their transitions.

In the *linear function approximation technique*, a linear architecture consisting of a set of k feature vectors ($|\mathbb{S}|$-dimensional) $\{\phi_i \in \mathbb{R}^{|\mathbb{S}|}, 1 \leq i \leq k\}$, where $1 \leq k \ll |\mathbb{S}|$, is chosen *a priori*. For a state $s \in \mathbb{S}$, we define

$$\phi(s) \triangleq \begin{bmatrix} \phi_1(s) \\ \phi_2(s) \\ \vdots \\ \phi_k(s) \end{bmatrix}_{k \times 1}, \quad \Phi \triangleq \begin{bmatrix} \phi(s_1)^\top \\ \phi(s_2)^\top \\ \vdots \\ \phi(s_{|\mathbb{S}|})^\top \end{bmatrix}_{|\mathbb{S}| \times k}, \tag{3}$$

where the vector $\phi(s)$ is called the *feature vector* corresponding to the state $s \in \mathbb{S}$, while the matrix Φ is called the *feature matrix*.

Primarily, the task in linear function approximation is to find a weight vector $z \in \mathbb{R}^k$ such that the predicted value function $\Phi z \approx V^\pi$. Given Φ, the best approximation of V^π is its projection on to the closed subspace $\mathbb{H}^\Phi = \{\Phi z | z \in \mathbb{R}^k\}$ (column space of Φ) with respect to some norm on $\mathbb{R}^{|\mathbb{S}|}$. Typically, one uses the weighted semi-norm $\|\cdot\|_\nu$ on $\mathbb{R}^{|\mathbb{S}|}$, where $\nu(\cdot)$ is the sample probability distribution with which the states $\mathbf{s}_t$ occur in the sample trajectory. It is assumed that $\nu(s) > 0, \forall s \in \mathbb{S}$. The semi-norm $\|\cdot\|_\nu$ on $\mathbb{R}^{|\mathbb{S}|}$ is defined as $\|V\|_\nu^2 = \sum_{s \in \mathbb{S}} V(s)^2 \nu(s)$. The associated linear projection operator Π^ν is defined as $\Pi^\nu V^\pi = \arg\min_{h \in \mathbb{H}^\Phi} \|V^\pi - h\|_\nu^2$. It is not hard to derive the following closed form expression for Π^ν.

$$\Pi^\nu = \Phi(\Phi^\top D^\nu \Phi)^{-1} \Phi^\top D^\nu, \tag{4}$$

where D^ν is the diagonal matrix with $D^\nu_{ii} = \nu(s_i), i = 1, \ldots, |\mathbb{S}|$. On a technical note, observe that the projection is obtained by minimizing the squared ν-weighted distance from the true value function V^π and this distance is referred to as the *mean squared error (MSE)*, *i.e.*,

Footnote 3 continued
dynamics of the system. Hence, off-policy can be interpreted as a contra-positive statement of this definition of on-policy and in that sense, our setting is indeed off-policy. See Sutton et al. (2009).

$$\text{MSE}(z) \triangleq \|V^{\pi} - \Phi z\|_{\nu}^{2}, \quad z \in \mathbb{R}^{k}. \tag{5}$$

However, it is hard to evaluate or even estimate Π^{ν} since it requires complete knowledge of the sampling distribution ν and also requires $|\mathbb{S}|$ amount of memory for storing D^{ν}. Therefore, one has to resort to additional approximation techniques to estimate the projection $\Pi^{\nu} V^{\pi}$ which is indeed the prime objective of this paper.

Goal of this paper: To find a vector $z^{*} \in \mathbb{R}^{k}$ such that $\Phi z^{*} \approx \Pi^{\nu} V^{\pi}$ without knowing Π^{ν} or trying to estimate the same.

A caveat is in order. It is important to note that the efficacy of the learning method depends on the choice of the feature set $\{\phi_i\}$ (Lagoudakis and Parr 2003). One can either utilize prior knowledge about the system to develop hard-coded features or employ off-the-shelf basis functions[4] from the literature. In this paper, we assume that a carefully chosen set of features is available *a priori*.

2 Related work

The existing algorithms can be broadly classified as

1. Linear methods which include temporal difference method (TD(λ), $\lambda \in [0, 1]$ Sutton (1988); Tsitsiklis and Roy (1997)), gradient temporal difference methods (GTD Sutton et al. 2009; GTD2 Sutton et al. 2009; TDC Sutton et al. 2009) and residual gradient (RG) schemes (Baird 1995), whose computational complexities are linear in k and hence are good for large values of k and
2. Second-order methods which include least squares temporal difference (LSTD) (Bradtke and Barto 1996; Boyan 2002) and least squares policy evaluation (LSPE) (Nedić and Bertsekas 2003) whose computational complexities are quadratic in k and are useful for moderate values of k. Second-order methods, albeit computationally expensive, are seen to be more data efficient than others except in the case when trajectories are very small (Dann et al. 2014).

In cases where the Markov chain is ergodic (*i.e.* irreducible and aperiodic) and the sampling distribution ν is the stationary distribution of the Markov chain, then with Φ being a full column rank matrix, the convergence of TD(λ) is guaranteed (Tsitsiklis and Roy 1997). But in cases where the sampling distribution ν is not the stationary distribution of the Markov chain or the projected subspace is a non-linear manifold, then TD(λ) can diverge (Tsitsiklis and Roy 1997; Baird 1995). However, both LSTD and LSPE algorithms are stable (Schoknecht 2002) and are also seen to be independent of the sampling distribution ν. However, there do not exist any extensions of LSTD and LSPE to the non-linear function approximation.

Tsitsiklis and Roy (1997) gave a different characterization for the stable limit point of TD(0) as the fixed point of the *projected Bellman operator* $\Pi^{\nu} T^{\pi}$,

$$\Phi z = \Pi^{\nu} T^{\pi} \Phi z, \tag{6}$$

where ν is the stationary distribution of the underlying ergodic chain.

[4] Some of the commonly used basis functions are radial basis functions (RBF), polynomials, Fourier basis functions (Konidaris et al. 2011) and cerebellar model articulation controller (CMAC) (Eldracher et al. 1994), to name a few.

This characterization yields a new error function, the *mean squared projected Bellman error (MSPBE)* which is defined as follows:

$$\text{MSPBE}(z) \triangleq \|\varPhi z - \varPi^{\nu} T^{\pi} \varPhi z\|_{\nu}^{2}, \quad z \in \mathbb{R}^{k}. \tag{7}$$

The LSTD algorithm (Bradtke and Barto 1996; Boyan 2002) is a fitted value function method (least squares approach) obtained by directly solving MSPBE over the sample trajectory using sample averaging of the individual transitions. However, the LSPE method (Nedić and Bertsekas 2003) solves MSPBE indirectly using a double minimization procedure where the primary minimizer finds the projection of the Bellman operator value using the least squares approach with the proximal Bellman operator value being obtained from the secondary gradient based minimizer. In Sutton et al. (2009), MSPBE is exquisitely manoeuvred to derive multiple stable $\varTheta(k)$ algorithms like TDC and GTD2. A non-linear function approximation version of the GTD2 algorithm is also available (Maei et al. 2009). The method is shown to be stable and the convergence to the sub-optimal solutions is also guaranteed under reasonably realistic assumptions (Maei et al. 2009). The sub-optimality of the solutions is expected as GTD2 is a gradient-based method and the convexity of the objective function does not always hold in non-linear function approximation settings.

Another pertinent error function is the *mean squared Bellman residue* (MSBR) which is defined as follows:

$$\text{MSBR}(z) \triangleq \mathbb{E}\left[(\mathbb{E}\left[\delta_t(z)|\mathbf{s}_t\right])^2\right], z \in \mathbb{R}^k, \tag{8}$$

where $\delta_t(z) \triangleq \mathbf{r}_t + \gamma z^{\top}\phi(\mathbf{s}_t') - \phi(\mathbf{s}_t)$ is the temporal difference error under function approximation when z is the associated approximation parameter. Note that MSBR is a measure of how closely the prediction vector represents the solution to the Bellman equation.

Residual gradient (RG) algorithm (Baird 1995) minimizes the error function MSBR directly using stochastic gradient search. Indeed, RG solves $\nabla_z \text{MSBR} = 0 \Rightarrow \mathbb{E}\Big[\mathbb{E}\left[\delta_t(z)|\mathbf{s}_t\right]\Big]$ $\mathbb{E}\Big[\mathbb{E}\left[(\gamma\phi(\mathbf{s}_t) - \phi(\mathbf{s}_t))|\mathbf{s}_t\right]\Big] = 0$. The above expression is a product of two expectations conditioned on the current state $\mathbf{s}_t$. Hence it requires two independent samples $\mathbf{s}_t'$ and $\mathbf{s}_t''$ of the next state when in the current state $\mathbf{s}_t$. This is generally referred to as *double sampling*. Even though the RG algorithm guarantees convergence, due to large variance, the convergence rate is small (Schoknecht and Merke 2003).

Eligibility traces (Sutton 1988) are a mechanism to accelerate learning by blending temporal difference methods with Monte Carlo simulation (averaging the values) and weighted using a geometric distribution with parameter $\lambda \in [0, 1]$. Eligibility traces can be integrated into most of these algorithms.[5] In this paper, we do not consider the treatment of eligibility traces.

Table 1 provides a list of important TD based algorithms along with the associated error objectives. The algorithm complexities and other characteristics are also shown in the table.

Put succinctly, when linear function approximation is applied in an RL setting, the main 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 gradient-free technique called the *cross entropy (CE) method* instead to find the minimum. By '*gradient-free*', we mean the algorithm does not incorporate information about the gradient of the objective function, rather it uses the function values themselves. The cross entropy method as such lies within the general class of *model based search methods* (Zlochin et al. 2004).

[5] The algorithms with eligibility traces are named with (λ) appended, for example TD(λ), LSTD(λ) *etc*.

Table 1 Comparison of the state-of-the-art function approximation RL algorithms

Algorithm	Complexity	Error	Elig. Trace	Stability	NLFA[a]
LSTD	$\Theta(k^3)$	MSPBE	√	√	×
TD	$\Theta(k)$	MSPBE	√	×	√
LSPE	$\Theta(k^3)$	MSPBE	√	√	×
GTD	$\Theta(k)$	MSPBE	–	√	×
GTD2	$\Theta(k)$	MSPBE	√	√	√
TDC	$\Theta(k)$	MSPBE	√	√	×
RG	$\Theta(k)$	MSBR	√	√	×

[a] NLFA non-linear function approximation

Other methods in this class are *model reference adaptive search (MRAS)* (Hu et al. 2007), *gradient-based adaptive stochastic search for simulation optimization (GASSO)* (Zhou et al. 2014), *ant colony optimization (ACO)* (Dorigo and Gambardella 1997) and *estimation of distribution algorithms (EDAs)* (Mühlenbein and Paass 1996). Model based search methods have been applied to the control problem[6] in Hu et al. (2008), Mannor et al. (2003), Busoniu et al. (2009) and in basis adaptation[7] (Menache et al. 2005), but this is the first time such a procedure has been applied to the prediction problem. However, due to the naive batch based approach of the original CE method, it cannot be directly applied to the online RL setting. In this paper, therefore, we propose two incremental, adaptive, online algorithms which solve MSBR and MSPBE respectively by employing a stochastic approximation version of the cross entropy method proposed in Joseph and Bhatnagar (2016a, b, 2018).

2.1 Our contributions

The *cross entropy (CE) method* (Rubinstein and Kroese 2013; Boer et al. 2005) is a model based search algorithm to find the global maximum of a given real valued objective function. In this paper, we propose for the first time, an adaptation of this method to the problem of parameter tuning in order to find the best estimates of the value function V^π for a given policy π under the linear function approximation architecture. We propose two prediction algorithms using the multi-timescale stochastic approximation framework (Robbins and Monro 1951; Borkar 1997; Kushner and Clark 1978) which minimize MSPBE and MSBR respectively. The algorithms possess the following attractive features:

1. A remodelling of the famous CE method to a model-free MDP framework using the stochastic approximation framework.
2. Stable with minimal restrictions on both the structural properties of the underlying Markov chain and on the sample trajectory.
3. Minimal restriction on the feature set.
4. Computational complexity is quadratic in the number of features (this is a significant improvement compared to the cubic complexity of the least squares algorithms).
5. Competitive with least squares and other state-of-the-art algorithms in terms of accuracy.
6. Algorithms are incremental update, adaptive, streamlined and online.

[6] The problem here is to find the optimal basis of the MDP.

[7] The basis adaptation problem is to find the best parameters of the basis functions for a given policy.

7. Algorithms provide guaranteed convergence to the global minimum of MSPBE (or MSBR).
8. Relative ease in extending the algorithms to non-linear function approximation settings.

A noteworthy observation is that under linear architecture, both MSPBE and MSBR are strongly convex functions (Dann et al. 2014) and hence their local and global minima overlap. Hence, the fact that CE method finds the global minima as opposed to local minima, unlike gradient search, does not provide any tangible advantage in terms of the quality of the solution. Nonetheless, in the case of non-linear function approximators, the convexity property does not hold in general and so there may exist multiple local minima in the objective and the gradient search schemes would get stuck in local optima unlike CE based search. We have not explored analytically the non-linear case in this paper. Notwithstanding, we have applied our algorithm to the non-linear MDP setting defined in section X of Tsitsiklis and Roy (1997) and the results obtained are quite impressive. The MDP setting in Tsitsiklis and Roy (1997) is a classic example where TD(0) is shown to diverge and GTD2 is shown to produce sub-optimal solutions. This demonstrates the robustness of our algorithm which is quite appealing, considering the fact that the state-of-the-art RL algorithms are specifically designed to perform in a linear environment and extending them to domains beyond the realm of linearity is quite tedious and often impossible. In view of all these alluring features, our approach can be viewed as a significant first step towards efficiently using model based search for policy evaluation in a generalized RL environment.

3 Summary of notation

We use $\mathbf{X}$ for random variable and x for deterministic variable. Let $\mathbb{I}_{k\times k}$ and $0_{k\times k}$ be the identity matrix and the zero matrix with dimensions $k \times k$ respectively. For set A, I_A represents the indicator function of A, *i.e.*, $I_A(x) = 1$ if $x \in A$ and 0 otherwise. Let $f_\theta : \mathbb{R}^n \to \mathbb{R}$ denote the *probability density function* (PDF) over $\mathbb{R}^n$ parametrized by θ. Let $\mathbb{E}_\theta[\cdot]$ and P_θ denote the *expectation* and the induced *probability measure w.r.t.* f_θ. For $\rho \in (0, 1)$ and $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}$, let $\gamma_\rho(\mathcal{H}, \theta)$ denote the $(1 - \rho)$-quantile of $\mathcal{H}(\mathbf{X})$ *w.r.t.* f_θ, *i.e.*,

$$\gamma_\rho(\mathcal{H}, \theta) \triangleq \sup\{\ell \in \mathbb{R} \mid P_\theta(\mathcal{H}(\mathbf{X}) \geq \ell) \geq \rho\}. \tag{9}$$

Let $int(A)$ be the *interior* of set A. Let $\mathcal{N}_n(m, V)$ represent the n-variate Gaussian distribution with mean vector $m \in \mathbb{R}^n$ and covariance matrix $V \in \mathbb{R}^{n\times n}$. A function $L : \mathbb{R}^n \to \mathbb{R}$ is *Lipschitz continuous*, if $\exists K \geq 0$ *s.t.* $|L(x) - L(y)| \leq K\|x - y\|, \forall x, y \in \mathbb{R}^n$, where $\|\cdot\|$ is some norm defined on $\mathbb{R}^n$.

4 Background: the CE method

To better understand our algorithm, we explicate the original CE method first.

4.1 Objective of CE

The *cross entropy (CE) method* (Rubinstein and Kroese 2013; Boer et al. 2005) solves problems of the following form:

$$\text{Find} \quad x^* \in \mathop{\arg\max}_{x\in\mathcal{X}\subseteq\mathbb{R}^m} \mathcal{H}(x),$$

where $\mathcal{H}(\cdot)$ is a multi-modal real-valued 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 maxima of $\mathcal{H}(\cdot)$. The CE method adopts an iterative procedure where at each iteration t, a search is conducted on a space of parametrized probability distributions $\{f_\theta | \theta \in \Theta\}$ over $\mathcal{X}$, where Θ (a subset of the multi-dimensional Euclidean space) is the parameter space, to find a distribution parameter θ_t which reduces the *Kullback–Leibler (KL)* divergence (also called the cross entropy distance) (Kullback 1959) from the optimal model. The most commonly used class here is the *natural exponential family of distributions (NEF).*

Natural exponential family of distributions (Morris 1982): These are denoted as $\mathcal{C} \triangleq \{f_\theta(x) = h(x)e^{\theta^\top \Gamma(x) - K(\theta)} \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$, where $h : \mathbb{R}^m \longrightarrow \mathbb{R}$, $\Gamma : \mathbb{R}^m \longrightarrow \mathbb{R}^d$ and $K : \mathbb{R}^d \longrightarrow \mathbb{R}$. By rearranging the parameters, we can show that the Gaussian distribution with mean vector μ and the covariance matrix Σ belongs to $\mathcal{C}$. In this case,

$$f_\theta(x) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right\}, \tag{10}$$

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

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

4.2 CE method (ideal version)

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 very 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 probability distribution sequence should improve. The most common choice for γ_{t+1} is $\gamma_\rho(\mathcal{H}, \theta_t)$, the $(1-\rho)$-quantile of $\mathcal{H}(\mathbf{X})$ *w.r.t.* the probability density function f_{θ_t}, where $\rho \in (0, 1)$ is set *a priori* for the algorithm. We take the Gaussian distribution as the preferred choice for f_θ in this paper. In this case, the model parameter is $\theta = (\mu, \Sigma)^\top$ where $\mu \in \mathbb{R}^m$ is the mean vector and $\Sigma \in \mathbb{R}^{m\times m}$ is the covariance matrix.

The CE algorithm is an iterative procedure which starts with an initial value $\theta_0 = (\mu_0, \Sigma_0)^\top$ of the mean vector and the covariance matrix tuple and at each iteration t, a new parameter $\theta_{t+1} = (\mu_{t+1}, \Sigma_{t+1})^\top$ is derived from the previous value θ_t as follows (from Section 4 of Hu et al. 2007):

$$\theta_{t+1} = \arg\max_{\theta\in\Theta} \mathbb{E}_{\theta_t}\left[S(\mathcal{H}(\mathbf{X}))I_{\{\mathcal{H}(\mathbf{X})\geq\gamma_{t+1}\}} \log f_\theta(\mathbf{X})\right], \tag{11}$$

where $S : \mathbb{R} \to \mathbb{R}_+$ is a positive and strictly monotonically increasing function.
If the gradient *w.r.t.* θ of the objective function in Eq. (11) is equated to 0, considering Gaussian PDF for f_θ (*i.e.*, using the expression provided in Eq. (10) for f_θ)and $\gamma_{t+1} = \gamma_\rho(\mathcal{H}, \theta_t)$, we obtain the following:

$$\mu_{t+1} = \frac{\mathbb{E}_{\theta_t}\left[\mathbf{g_1}\left(\mathcal{H}(\mathbf{X}), \mathbf{X}, \gamma_\rho(\mathcal{H}, \theta_t)\right)\right]}{\mathbb{E}_{\theta_t}\left[\mathbf{g_0}\left(\mathcal{H}(\mathbf{X}), \gamma_\rho(\mathcal{H}, \theta_t)\right)\right]} \triangleq \Upsilon_1(\mathcal{H}, \theta_t), \tag{12}$$

$$\Sigma_{t+1} = \frac{\mathbb{E}_{\theta_t}\left[\mathbf{g_2}\left(\mathcal{H}(\mathbf{X}), \mathbf{X}, \gamma_\rho(\mathcal{H}, \theta_t), \Upsilon_1(\mathcal{H}, \theta_t)\right)\right]}{\mathbb{E}_{\theta_t}\left[\mathbf{g_0}\left(\mathcal{H}(\mathbf{X}), \gamma_\rho(\mathcal{H}, \theta_t)\right)\right]} \triangleq \Upsilon_2(\mathcal{H}, \theta_t). \tag{13}$$

where

$$\mathbf{g_0}(\mathcal{H}(x), \gamma) \triangleq S(\mathcal{H}(x)) I_{\{\mathcal{H}(x) \geq \gamma\}}, \tag{14}$$

$$\mathbf{g_1}(\mathcal{H}(x), x, \gamma) \triangleq S(\mathcal{H}(x)) I_{\{\mathcal{H}(x) \geq \gamma\}} x, \tag{15}$$

$$\mathbf{g_2}(\mathcal{H}(x), x, \gamma, \mu) \triangleq S(\mathcal{H}(x)) I_{\{\mathcal{H}(x) \geq \gamma\}} (x - \mu)(x - \mu)^\top. \tag{16}$$

Remark 2 The function $S(\cdot)$ in Eq. (11) 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 Eq. (12), 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 case where $\mathcal{H}(\cdot)$ is positive, we can choose $S(x) = x$. However, in general scenarios, where $\mathcal{H}(\cdot)$ takes positive and negative values, the identity function is not an appropriate choice since the effect of the positive weights is reduced by the negative ones. In such cases, we take $S(x) = exp(rx), r \in \mathbb{R}_+$.

Thus the ideal CE algorithm can be expressed using the following recursion:

$$\theta_{t+1} = (\Upsilon_1(\mathcal{H}, \theta_t), \Upsilon_2(\mathcal{H}, \theta_t))^\top. \tag{17}$$

An illustration demonstrating the evolution of the model parameters of the CE method with Gaussian distribution during the optimization of a multi-modal objective function is provided in Fig. 16 of the "Appendix".

5 Comparison of the objectives: MSPBE and MSBR

This question is critical since most reinforcement learning algorithms can be characterized via some optimization problem which minimizes either MSBR or MSPBE. A comprehensive comparison of the two error functions is available in the literature (Schoknecht and Merke 2003; Schoknecht 2002; Scherrer 2010). A direct relationship between MSBR and MSPBE can be easily established as follows:

$$\text{MSBR}(z) = \text{MSPBE}(z) + \|T^\pi \Phi z - \Pi^\nu T^\pi \Phi z\|^2, \quad z \in \mathbb{R}^k. \tag{18}$$

This follows directly from Babylonian–Pythagorean theorem and the fact that $(T^\pi \Phi z - \Pi^\nu T^\pi \Phi z) \perp (\Pi^\nu T^\pi \Phi z - \Phi z), \forall z \in \mathbb{R}^k$. A vivid depiction of this relationship is shown in Fig. 1.

If the columns of the feature matrix Φ are linearly independent, then both the error functions MSBR and MSPBE are strongly convex (Dann et al. 2014). However, the respective minima of MSBR and MSPBE are related depending on whether the feature set is perfect or not. A feature set is *perfect* if $V^\pi \in \{\Phi z | z \in \mathbb{R}^k\}$. In the perfect case, $\exists z_0 \in \mathbb{R}^k$ *s.t.* $\Phi z_0 = V^\pi$ and hence MSBR(z_0) = 0. Since MSBR$(z) \geq 0, \forall z \in \mathbb{R}^k$, we have $z_0 = \arg\min_z$ MSBR(z). Now from (18), we get MSPBE$(z_0) = 0$ and $z_0 = \arg\min_z$ MSPBE(z) (again since MSPBE$(z) \geq 0, \forall z \in \mathbb{R}^k$). Hence in the perfect feature set scenario, the respective minima of MSBR and MSPBE coincide. However, in the imperfect case, they might differ since MSPBE$(z) \neq$ MSBR(z) for some $z \in \mathcal{Z}$ (follows from Eq. (18)).

In Scherrer (2010), Williams and Baird (1993), a relationship between MSBR and MSE is provided as shown in (19). Recall that MSE is the error which defines the projection operator Π^ν in the linear function approximation setting. It is found that, for a given ν with

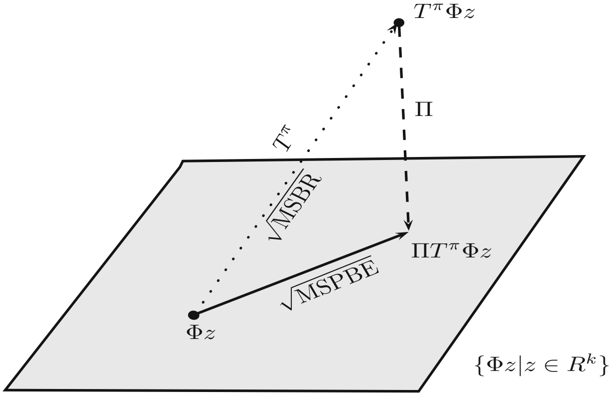


Fig. 1 Diagram depicting the relationship between the error functions MSPBE and MSBR

$\nu(s) > 0, \forall s \in \mathbb{S}$,

$$\sqrt{\text{MSE}(z)} \leq \frac{\sqrt{C(\nu)}}{1-\gamma}\sqrt{\text{MSBR}(z)}, \tag{19}$$

where $C(\nu) = \max_{s,s'} \frac{\mathrm{P}^\pi(s,s')}{\nu(s)}$. This bound (albeit loose) ensures that the minimization of MSBR is indeed stable and the solution so obtained cannot be too far from the projection $\Pi^\nu V^\pi$. A noticeable drawback with MSBR is the statistical overhead brought about by the double sampling required for its estimation. To elaborate this, recall that MSBR(z) = $\mathbb{E}\Big[\mathbb{E}\left[\delta_t(z)|\mathbf{s}_t\right]\mathbb{E}\left[\delta_t(z)|\mathbf{s}_t\right]\Big]$ (from Eq. 8). In the above expression of MSBR, we have a product of two conditional expectations conditioned on the current state $\mathbf{s}_t$. This implies that to estimate MSBR, one requires two independent samples of the next state, given the current state $\mathbf{s}_t$. Another drawback which was observed in the literature is the large variance incurred while estimating MSBR (Dann et al. 2014; Scherrer 2010), which inversely affects the rate of convergence of the optimization procedure. Also, in settings where only a finite length sample trajectory is available, the larger stochastic noise associated with the MSBR estimation will produce inferior quality solutions. MSPBE is attractive in the sense that double sampling is not required and there is sufficient empirical evidence (Dann et al. 2014) to believe that the minimum of MSPBE often has low MSE. The absence of double sampling is quite appealing, since for large complex MDPs obtaining sample trajectories is itself tedious, let alone double samples. Also, MSPBE when integrated with control algorithms is also shown to produce better quality policies (Lagoudakis and Parr 2003). Another less significant advantage is the fact that MSPBE$(z) \leq$ MSBR(z), $\forall z$ (follows from Eq. 18). This implies that the optimization algorithm can work with smaller objective function values compared to MSBR.
Now, we explore both the error functions analytically:

5.1 MSPBE

In Sutton et al. (2009), a compact expression for MSPBE is provided as follows:

$$\text{MSPBE}(z) = \left(\Phi^\top D^\nu (T^\pi V_z - V_z)\right)^\top \left(\Phi^\top D^\nu \Phi\right)^{-1} \left(\Phi^\top D^\nu (T^\pi V_z - V_z)\right), \tag{20}$$

where $V_z = \Phi z$, while Φ and D^ν are defined in Eqs. (3) and (4) respectively. Now the expression $\Phi^\top D^\nu (T^\pi V_z - V_z)$ is further rewritten as

$$\begin{aligned}\Phi^\top D^\nu (T^\pi V_z - V_z) &= \mathbb{E}\left[\mathbb{E}\left[\phi_t(\mathbf{r}_t + \gamma z^\top \phi'_t - z^\top \phi_t)|\mathbf{s}_t\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\phi_t \mathbf{r}_t|\mathbf{s}_t\right]\right] + \mathbb{E}\left[\mathbb{E}\left[\phi_t(\gamma\phi'_t - \phi_t)^\top|\mathbf{s}_t\right]\right]z, \\ &\quad \text{where } \phi_t \triangleq \phi(\mathbf{s}_t) \text{ and } \phi'_t \triangleq \phi(\mathbf{s}'_t). \qquad (21) \\ &\quad \text{Also,} \quad \Phi^\top D^\nu \Phi = \mathbb{E}\left[\phi_t \phi_t^\top\right]. \qquad (22)\end{aligned}$$

Putting all together we get,

$$\begin{aligned}\text{MSPBE}(z) &= \left(\mathbb{E}\left[\mathbb{E}\left[\phi_t \mathbf{r}_t|\mathbf{s}_t\right]\right] + \mathbb{E}\left[\mathbb{E}\left[\phi_t(\gamma\phi'_t - \phi_t)^\top|\mathbf{s}_t\right]\right]z\right)^\top \left(\mathbb{E}\left[\phi_t\phi_t^\top\right]\right)^{-1} \\ &\left(\mathbb{E}\left[\mathbb{E}\left[\phi_t \mathbf{r}_t|\mathbf{s}_t\right]\right] + \mathbb{E}\left[\mathbb{E}\left[\phi_t(\gamma\phi'_t - \phi_t)^\top|\mathbf{s}_t\right]\right]z\right) = \left(\omega_*^{(0)} + \omega_*^{(1)}z\right)^\top \omega_*^{(2)} \left(\omega_*^{(0)} + \omega_*^{(1)}z\right),\end{aligned} \tag{23}$$

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

This is a quadratic function in z. Note that in the above expression, the parameter vector z and the stochastic component involving $\mathbb{E}[\cdot]$ are decoupled. Hence the stochastic component can be estimated or tracked independent of the parameter vector z.

5.2 MSBR

We execute a similar decoupling procedure to the MSBR function. Indeed, from Eq. (8), we have

$$\begin{aligned}\text{MSBR}(z) &= \mathbb{E}\left[\mathbb{E}^2[\delta_t(z)|\mathbf{s}_t]\right] = \mathbb{E}\left[\mathbb{E}^2[\mathbf{r}_t + \gamma z^\top\phi'_t - z^\top\phi_t|\mathbf{s}_t]\right] \\ &= \mathbb{E}\left[\mathbb{E}^2[\mathbf{r}_t + \gamma z^\top\phi'_t|\mathbf{s}_t]\right] - 2\mathbb{E}\left[\mathbb{E}[\mathbf{r}_t + \gamma z^\top\phi'_t|\mathbf{s}_t]\mathbb{E}[z^\top\phi_t|\mathbf{s}_t]\right] + z^\top\mathbb{E}[\phi_t\phi_t^\top]z \\ &= \mathbb{E}\left[\mathbb{E}^2\left[\mathbf{r}_t|\mathbf{s}_t\right]\right] + \gamma^2 z^\top\mathbb{E}\left[\mathbb{E}[\phi'_t|\mathbf{s}_t]\mathbb{E}\left[\phi'_t|\mathbf{s}_t\right]^\top\right]z + 2z^\top\mathbb{E}\left[\mathbb{E}[\mathbf{r}_t|\mathbf{s}_t]\mathbb{E}[\phi'_t|\mathbf{s}_t]\right] \\ &\quad - 2z^\top\mathbb{E}\left[\mathbb{E}\left[\mathbf{r}_t|\mathbf{s}_t\right]\phi_t\right] - 2\gamma z^\top\mathbb{E}\left[\mathbb{E}[\phi'_t|\mathbf{s}_t]\phi_t^\top\right]z + z^\top\mathbb{E}[\phi_t\phi_t^\top]z \\ &= \mathbb{E}\left[\mathbb{E}^2\left[\mathbf{r}_t|\mathbf{s}_t\right]\right] + \gamma^2 z^\top\mathbb{E}\left[\mathbb{E}[\phi'_t|\mathbf{s}_t]\mathbb{E}\left[\phi'_t|\mathbf{s}_t\right]^\top\right]z \\ &\quad + 2z^\top\mathbb{E}\left[\mathbb{E}[\mathbf{r}_t|\mathbf{s}_t]\big(\mathbb{E}[\phi'_t|\mathbf{s}_t] - \phi_t\big)\right] + z^\top\mathbb{E}\left[\big(\phi_t - 2\gamma\mathbb{E}[\phi'_t|\mathbf{s}_t]\big)\phi_t^\top\right]z.\end{aligned}$$

Therefore,

$$\begin{aligned}\text{MSBR}(z) &= \upsilon_*^{(0)} + z^\top\upsilon_*^{(1)}z + 2z^\top\upsilon_*^{(2)} + z^\top\upsilon_*^{(3)}z, \\ &= \upsilon_*^{(0)} + z^\top(\upsilon_*^{(1)} + \upsilon_*^{(3)})z + 2z^\top\upsilon_*^{(2)},\end{aligned} \tag{24}$$

where $\upsilon_*^{(0)} \triangleq \mathbb{E}\left[\mathbb{E}^2\left[\mathbf{r}_t|\mathbf{s}_t\right]\right]$, $\upsilon_*^{(1)} \triangleq \gamma^2\mathbb{E}\left[\mathbb{E}[\phi'_t|\mathbf{s}_t]\mathbb{E}\left[\phi'_t|\mathbf{s}_t\right]^\top\right]$, $\upsilon_*^{(2)} \triangleq \mathbb{E}\left[\mathbb{E}[\mathbf{r}_t|\mathbf{s}_t]\big(\mathbb{E}[\phi'_t|\mathbf{s}_t] - \phi_t\big)\right]$ and $\upsilon_*^{(3)} \triangleq \mathbb{E}\left[\big(\phi_t - 2\gamma\mathbb{E}[\phi'_t|\mathbf{s}_t]\big)\phi_t^\top\right]$.

6 Proposed algorithms

We propose a generalized algorithm to approximate the value function V^{π} *(for a given policy* π*) with linear function approximation by minimizing either MSPBE or MSBR, where the optimization is performed using a multi-timescale stochastic approximation variant of the CE algorithm*. Since the CE method is a maximization algorithm, the objective function in the optimization problem here is the negative of MSPBE and MSBR. To state it more formally: In this paper, we solve the following two optimization problems:

1. $$z_p^* = \mathop{\arg\min}_{z \in \mathcal{Z} \subset \mathbb{R}^k} \text{MSPBE}(z) = \mathop{\arg\max}_{z \in \mathcal{Z} \subset \mathbb{R}^k} \mathcal{J}_p(z), \quad \text{where } \mathcal{J}_p = -\text{MSPBE}. \tag{25}$$

2. $$z_b^* = \mathop{\arg\min}_{z \in \mathcal{Z} \subset \mathbb{R}^k} \text{MSBR}(z) = \mathop{\arg\max}_{z \in \mathcal{Z} \subset \mathbb{R}^k} \mathcal{J}_b(z), \quad \text{where } \mathcal{J}_b = -\text{MSBR}. \tag{26}$$

Here $\mathcal{Z}$ is the solution space, *i.e.*, the space of parameter values of the function approximator. We also define $\mathcal{J}_p^* \triangleq \mathcal{J}_p(z_p^*)$ and $\mathcal{J}_b^* \triangleq \mathcal{J}_b(z_b^*)$.

⊛ ***Assumption (A2)***: The solution space $\mathcal{Z}$ is compact, *i.e.*, it is closed and bounded.

A few annotations about the algorithms are in order:

1. Tracking the objective function $\mathcal{J}_p, \mathcal{J}_b$: Recall that the goal of the paper is to develop an online and incremental prediction algorithm. This implies that the algorithm has to estimate the value function by recalibrating the prediction vector incrementally as new transitions of the sample trajectory are revealed. Note that the sample trajectory is simply a roll-out of an arbitrary realization of the underlying Markovian dynamics in the form of state transitions and their associated rewards and we assume that the sample trajectory satisfies the following assumption:

⊛ ***Assumption (A3):*** A sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^{\infty}$ is given, where $\mathbf{s}_t \sim \nu(\cdot)$, $\mathbf{s}'_t \sim \mathrm{P}^{\pi}(\mathbf{s}_t, \cdot)$ and $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}'_t)$. Let $\nu(s) > 0$, $\forall s \in \mathbb{S}$. Also, let ϕ_t, ϕ'_t, and $\mathbf{r}_t$ have uniformly bounded second moments. And the matrix $\mathbb{E}\left[\phi_t \phi_t^{\top}\right]$ 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 MDPs. However, the non-singularity requirement of the matrix $\mathbb{E}\left[\phi_t \phi_t^{\top}\right]$ is strict and one can ensure this condition by appropriately choosing the feature set.[8]

Now recall that in the analytic closed-form expression (Eq. (23)) of the objective function $\mathcal{J}_p(\cdot)$, we have isolated the stochastic and the deterministic parts. The stochastic part can be identified by the tuple $\omega_* \triangleq (\omega_*^{(0)}, \omega_*^{(1)}, \omega_*^{(2)})^{\top}$. So if we can find ways to track ω_*, then it implies that we can track the objective function $\mathcal{J}_p(\cdot)$. This is the line of thought we follow here. In our algorithm, we track ω_* by maintaining a time indexed 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 $\omega_*^{(i)}$, $0 \leq i \leq 2$. We show here that $\lim_{t \to \infty} \omega_t^{(i)} = \omega_*^{(i)}$, with probability one, $0 \leq i \leq 2$. Now the stochastic recursion to track ω_* is given by

$$\omega_{t+1} = \omega_t + \alpha_{t+1} \Delta\omega_{t+1}. \tag{27}$$

[8] A sufficient condition is the columns of the feature matrix Φ are linearly independent.

The increment term $\Delta\omega_{t+1} \triangleq (\Delta\omega^{(0)}_{t+1}, \Delta\omega^{(1)}_{t+1}, \Delta\omega^{(2)}_{t+1})^\top$ used for this recursion is defined as follows:

$$\left.\begin{aligned} \triangle\omega^{(0)}_{t+1} &\triangleq \mathbf{r}_t\phi_t - \omega^{(0)}_t, \\ \triangle\omega^{(1)}_{t+1} &\triangleq \phi_t(\gamma\phi^{\prime}_t - \phi_t)^\top - \omega^{(1)}_t, \\ \triangle\omega^{(2)}_{t+1} &\triangleq \mathbb{I}_{k\times k} - \phi_t\phi_t^\top\omega^{(2)}_t, \end{aligned}\right\} \tag{28}$$

where $\phi_t \triangleq \phi(\mathbf{s}_t)$ and $\phi^{\prime}_t \triangleq \phi(\mathbf{s}^{\prime}_t)$.
Now we define the estimate of $\mathcal{J}_p(\cdot)$ at time t as follows:
For a given $z \in \mathcal{Z}$,

$$\bar{\mathcal{J}}_p(\omega_t, z) \triangleq -\left(\omega^{(0)}_t + \omega^{(1)}_t z\right)^\top \omega^{(2)}_t \left(\omega^{(0)}_t + \omega^{(1)}_t z\right). \tag{29}$$

Superficially, it is similar to the expression of $\mathcal{J}_p$ in Eq. (23) except for ω_t replacing ω_*. Since ω_t tracks ω_*, it is easy to verify that $\bar{\mathcal{J}}_p(\omega_t, z)$ indeed tracks $\mathcal{J}_p(z)$ for a given $z \in \mathcal{Z}$.

Similarly, in the case of MSBR, we require the following double sampling assumption on the sample trajectory:

⊛ ***Assumption (A3)***$'$: A sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{r}^{\prime}_t, \mathbf{s}^{\prime}_t, \mathbf{s}^{\prime\prime}_t)\}_{t=0}^{\infty}$ is provided, where $\mathbf{s}_t \sim \nu(\cdot)$, $\mathbf{s}^{\prime}_t \sim \mathrm{P}^{\pi}(\mathbf{s}_t, \cdot)$, $\mathbf{s}^{\prime\prime}_t \sim \mathrm{P}^{\pi}(\mathbf{s}_t, \cdot)$ with $\mathbf{s}^{\prime}_t$ and $\mathbf{s}^{\prime\prime}_t$ sampled independently. Also, $\mathbf{r}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}^{\prime}_t)$ and $\mathbf{r}^{\prime}_t = \mathrm{R}(\mathbf{s}_t, \pi(\mathbf{s}_t), \mathbf{s}^{\prime\prime}_t)$. Let $\nu(s) > 0, \forall s \in \mathbb{S}$. Further, let $\phi_t, \phi^{\prime}_t, \phi^{\prime\prime}_t, \mathbf{r}_t$, and $\mathbf{r}^{\prime}_t$ have uniformly bounded second moments (where $\phi_t \triangleq \phi(\mathbf{s}_t)$, $\phi^{\prime}_t \triangleq \phi(\mathbf{s}^{\prime}_t)$, $\phi^{\prime\prime}_t \triangleq \phi(\mathbf{s}^{\prime\prime}_t)$).

Assumption A3$'$ does not contain any non-singularity condition. However, it demands the availability of two independent transitions $(\mathbf{s}^{\prime}_t, \mathbf{r}_t)$ and $(\mathbf{s}^{\prime\prime}_t, \mathbf{r}^{\prime}_t)$ given the current state $\mathbf{s}_t$. This requirement is referred to as the *double sampling*.

We maintain the time indexed variable $\upsilon_t \triangleq (\upsilon^{(0)}_t, \upsilon^{(1)}_t, \upsilon^{(2)}_t, \upsilon^{(3)}_t)^\top$, where $\upsilon^{(0)}_t \in \mathbb{R}$, $\upsilon^{(1)}_t \in \mathbb{R}^{k\times k}$, $\upsilon^{(2)}_t \in \mathbb{R}^{k\times 1}$ and $\upsilon^{(3)}_t \in \mathbb{R}^{k\times k}$. Now the stochastic recursion to track υ_* is given by

$$\upsilon_{t+1} = \upsilon_t + \alpha_{t+1}\Delta\upsilon_{t+1}. \tag{30}$$

The increment term $\Delta\upsilon_{t+1} \triangleq (\upsilon^{(0)}_{t+1}, \upsilon^{(1)}_{t+1}, \upsilon^{(2)}_{t+1}, \upsilon^{(3)}_{t+1})^\top$ used in the above recursion is defined as follows:

$$\left.\begin{aligned} \triangle\upsilon^{(0)}_{t+1} &\triangleq \mathbf{r}_t\mathbf{r}^{\prime}_t - \upsilon^{(0)}_t, \\ \triangle\upsilon^{(1)}_{t+1} &\triangleq \gamma^2\phi^{\prime}_t\phi^{\prime\prime\top}_t - \upsilon^{(1)}_t, \\ \triangle\upsilon^{(2)}_{t+1} &\triangleq \mathbf{r}_t\big(\phi^{\prime}_t - \phi_t\big) - \upsilon^{(2)}_t, \\ \triangle\upsilon^{(3)}_{t+1} &\triangleq \big(\phi_t - 2\gamma\phi^{\prime}_t\big)\phi_t^\top - \upsilon^{(3)}_t. \end{aligned}\right\} \tag{31}$$

We also define the estimate of $\mathcal{J}_b(\cdot)$ at time t as follows:
For a given $z \in \mathcal{Z}$,

$$\bar{\mathcal{J}}_b(\upsilon_t, z) \triangleq -\left(\upsilon^{(0)}_t + z^\top(\upsilon^{(1)}_t + \upsilon^{(3)}_t)z + 2z^\top\upsilon^{(2)}_t\right). \tag{32}$$

2. Tracking the ideal CE method: The ideal CE method defined in Eq. (17) is computationally intractable due to the inherent hardness involved in computing the quantities $\mathbb{E}_{\theta_t}[\cdot]$ and $\gamma_\rho(\cdot, \cdot)$ efficiently (hence the tag name "ideal"). There are multiple ways one can track the ideal CE method. In this paper, we consider the efficient tracking of the ideal CE method using the stochastic approximation (SA) framework proposed in Joseph and Bhatnagar (2016a, b, 2018). The stochastic approximation approach is efficient both computationally and storage wise when compared to the rest of the state-of-the-art CE tracking methods.The SA variant is also shown to exhibit global optimum convergence, *i.e.*, the model sequence $\{\theta_t\}_{t \in \mathbb{N}}$ converges to the degenerate distribution concentrated on any of the global optima of the objective function. The SA version of the CE method consists of three stochastic recursions which are defined as follows:

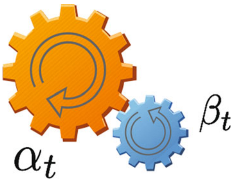
- **Tracking** $\gamma_\rho(\mathcal{J}_p, \theta)$**:** $\gamma_{t+1} = \gamma_t - \beta_{t+1}\Delta\gamma_{t+1}(\mathbf{Z}_{t+1})$, where $\mathbf{Z}_{t+1} \sim f_\theta$ and $\Delta\gamma_{t+1}(x) \triangleq -(1-\rho)\mathbb{I}_{\{\bar{\mathcal{J}}_p(\omega_t, x) \geq \gamma_t\}} + \rho\mathbb{I}_{\{\bar{\mathcal{J}}_p(\omega_t, x) \leq \gamma_t\}}$. (33)
- **Tracking** $\Upsilon_1(\mathcal{J}_p, \theta)$**:** $\xi^{(0)}_{t+1} = \xi^{(0)}_t + \beta_{t+1}\Delta\xi^{(0)}_{t+1}(\mathbf{Z}_{t+1})$, where $\mathbf{Z}_{t+1} \sim f_\theta$ and $\Delta\xi^{(0)}_{t+1}(x) \triangleq \mathbf{g_1}(\bar{\mathcal{J}}_p(\omega_t, x), x, \gamma_t) - \xi^{(0)}_t\mathbf{g_0}(\bar{\mathcal{J}}_p(\omega_t, x), \gamma_t)$. (34)
- **Tracking** $\Upsilon_2(\mathcal{J}_p, \theta)$**:** $\xi^{(1)}_{t+1} = \xi^{(1)}_t + \beta_{t+1}\Delta\xi^{(1)}_t(\mathbf{Z}_{t+1})$, where $\mathbf{Z}_{t+1} \sim f_\theta$ and $\Delta\xi^{(1)}_{t+1}(x) \triangleq \mathbf{g_2}(\bar{\mathcal{J}}_p(\omega_t, x), x, \gamma_t, \xi^{(0)}_t) - \xi^{(1)}_t\mathbf{g_0}(\bar{\mathcal{J}}_p(\omega_t, x), \gamma_t)$. (35)

Note that the above recursions are defined for the objective function $\mathcal{J}_p$. However, in the case of $\mathcal{J}_b$, the recursions are similar except for $\mathcal{J}_b$ replacing $\mathcal{J}_p$ and υ_t replacing ω_t wherever required.

3. Learning rates and timescales: Our algorithms use two learning rates $\{\alpha_t\}_{t \in \mathbb{N}}$ and $\{\beta_t\}_{t \in \mathbb{N}}$, which are deterministic, positive, non-increasing, predetermined (chosen a priori) and satisfy the following conditions:

$$\sum_{t=1}^{\infty} \alpha_t = \sum_{t=1}^{\infty} \beta_t = \infty, \qquad \sum_{t=1}^{\infty} \left(\alpha_t^2 + \beta_t^2\right) < \infty, \qquad \lim_{t \to \infty} \frac{\alpha_t}{\beta_t} = 0. \tag{36}$$

In a multi-timescale stochastic approximation setting (Borkar 1997), it is important to understand the difference between timescale and learning rate. The timescale of a stochastic recursion is defined by its learning rate (also referred to as step-size). Note that from the conditions imposed on the learning rates $\{\alpha_t\}_{t \in \mathbb{N}}$ and $\{\beta_t\}_{t \in \mathbb{N}}$ in Eq. (36), we have $\frac{\alpha_t}{\beta_t} \to 0$. So α_t decays to 0 relatively faster than β_t. Hence the timescale obtained from $\{\beta_t\}_{t \in \mathbb{N}}$ is considered faster as compared to the other. So in a multi-timescale stochastic recursion scenario, the evolution of the recursion controlled by $\{\alpha_t\}$ (that converges relatively faster to 0) is slower compared to the recursions controlled by $\{\beta_t\}$. This is because the increments are weighted by their learning rates, *i.e.*, the learning rates control the quantity of change that occurs to the variables when the update is executed. When observed from the faster timescale recursion, one can consider the slower timescale recursion to be almost stationary, while when viewed from the slower timescale, the faster timescale recursion appears to have equilibrated. This attribute of the multi-timescale recursions are very important in the analysis of the algorithm. In the analysis, when studying the asymptotic behaviour of a particular stochastic recursion, we can consider the variables of other recursions which are on slower timescales to be constant. In our algorithm, the recursion of ω_t and θ_t proceed along the slowest timescale and so updates of ω_t appear to be quasi-static when viewed from the timescale on which the recursions governed by β_t proceed. The recursions of γ_t, $\xi^{(0)}_t$ and $\xi^{(1)}_t$ proceed along the faster timescale and hence appear equilibrated when viewed from the slower recursion. The coherent behaviour exhibited by the algorithms is primarily attributed to the timescale differences obeyed by the various recursions.



The algorithm SCE-MSPBEM acronym for *stochastic cross entropy-mean squared projected Bellman error minimization* that minimizes the mean squared projected Bellman error (MSPBE) by incorporating a multi-timescale stochastic approximation variant of the cross entropy (CE) method is formally presented in Algorithm 1.

```
Algorithm 1: SCE-MSPBEM
Data: $\alpha_t, \beta_t, c_t \in (0,1), c_t \to 0, \epsilon_1, \lambda, \rho \in (0,1), \quad S(\cdot) : \mathbb{R} \to \mathbb{R}_+$
Initialization: $\gamma_0 = 0, \gamma_0^p = -\infty, \theta_0 = (\mu_0, \Sigma_0)^\top, T_0 = 0, \xi_t^{(0)} = 0_{k \times 1}, \xi_t^{(1)} = 0_{k \times k}$,
$\omega_0^{(0)} = 0_{k \times 1}, \omega_0^{(1)} = 0_{k \times k}, \omega_0^{(2)} = 0_{k \times k}, \theta^p = NULL$
foreach $(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)$ of the sample trajectory do        /*Traj follows (A3)*/
    $\mathbf{Z}_{t+1} \sim \widehat{f}_{\theta_t}$, where $\widehat{f}_{\theta_t} = (1-\lambda) f_{\theta_t} + \lambda f_{\theta_0}$    (37)
    Estimate Objective Function $\mathcal{J}_p$:
    $\boldsymbol{\omega_{t+1} = \omega_t + \alpha_{t+1} \Delta \omega_{t+1}}$    (38)
    $\boldsymbol{\bar{\mathcal{J}}_p(\omega_t, \mathbf{Z}_{t+1}) = -(\omega_t^{(0)} + \omega_t^{(1)} \mathbf{Z}_{t+1})^\top \omega_t^{(2)} (\omega_t^{(0)} + \omega_t^{(1)} \mathbf{Z}_{t+1})}$    (39)
    Track $\gamma_\rho(\mathcal{J}_p, \widehat{\theta}_t)$:    $\gamma_{t+1} = \gamma_t - \beta_{t+1} \Delta \gamma_{t+1}(\mathbf{Z}_{t+1})$    (40)
    Track $\Upsilon_1(\mathcal{J}_p, \widehat{\theta}_t)$:    $\xi_{t+1}^{(0)} = \xi_t^{(0)} + \beta_{t+1} \Delta \xi_{t+1}^{(0)}(\mathbf{Z}_{t+1})$    (41)
    Track $\Upsilon_2(\mathcal{J}_p, \widehat{\theta}_t)$:    $\xi_{t+1}^{(1)} = \xi_t^{(1)} + \beta_{t+1} \Delta \xi_{t+1}^{(1)}(\mathbf{Z}_{t+1})$    (42)
    if $\theta^p \neq NULL$ then
        $\mathbf{Z}_{t+1}^p \sim \widehat{f}_{\theta^p} \triangleq \lambda f_{\theta_0} + (1-\lambda) f_{\theta^p}$
        $\gamma_{t+1}^p = \gamma_t^p - \beta_{t+1} \Delta \gamma_{t+1}^p(\mathbf{Z}_{t+1}^p)$    (43)
    Compare Thresholds: $T_{t+1} = T_t + c(\mathbb{I}_{\{\gamma_{t+1} > \gamma_{t+1}^p\}} - \mathbb{I}_{\{\gamma_{t+1} \leq \gamma_{t+1}^p\}} - T_t)$    (44)
    if $T_{t+1} > \epsilon_1$ then
        Save Old Model: $\gamma_{t+1}^p = \gamma_t; \quad \theta^p = \theta_t$
        Update Model: $\theta_{t+1} = \theta_t + \alpha_{t+1} \left( (\xi_t^{(0)}, \xi_t^{(1)})^\top - \theta_t \right)$    (45)
        Reset Parameters: $T_t = 0; \quad c = c_t$    (46)
    else
        $\gamma_{t+1}^p = \gamma_t^p; \quad \theta_{t+1} = \theta_t$
    $t := t + 1$;
```

The algorithm SCE-MSBRM acronym for *stochastic cross entropy-mean squared Bellman residue minimization* that minimizes the mean squared Bellman residue (MSBR) by incorporating a multi-timescale stochastic approximation variant of the cross entropy (CE) method is formally presented in Algorithm 2.

Algorithm 2: SCE-MSBRM

```
Data: $\alpha_t, \beta_t, c_t \in (0,1), c_t \to 0, \epsilon_1, \lambda, \rho \in (0,1), \quad S(\cdot) : \mathbb{R} \to \mathbb{R}_+$
Initialization: $\gamma_0 = 0, \gamma_0^p = -\infty, \theta_0 = (\mu_0, \Sigma_0)^\top, T_0 = 0, \xi_t^{(0)} = 0_{k\times 1}, \xi_t^{(1)} = 0_{k\times k},$
$\upsilon_0^{(0)} = 0, \upsilon_0^{(1)} = 0_{k\times k}, \upsilon_0^{(2)} = 0_{k\times 1}, \upsilon_0^{(3)} = 0_{k\times k}, \theta^p = NULL$
foreach $(\mathbf{s}_t, \mathbf{r}_t, \mathbf{r}'_t, \mathbf{s}'_t, \mathbf{s}''_t)$ of the sample trajectory do      /*Trajectory follows (A3)'*/
    $\mathbf{Z}_{t+1} \sim \widehat{f}_{\theta_t}$, where $\widehat{f}_{\theta_t} = (1-\lambda) f_{\theta_t} + \lambda f_{\theta_0}$      (47)
    Estimate Objective Function $\mathcal{J}_b$:
    $\boldsymbol{\upsilon}_{t+1} = \boldsymbol{\upsilon}_t + \boldsymbol{\alpha}_{t+1} \boldsymbol{\Delta} \boldsymbol{\upsilon}_{t+1}$      (48)
    $\bar{\mathcal{J}}_b(\boldsymbol{\upsilon}_t, \mathbf{Z}_{t+1}) = -\left(\boldsymbol{\upsilon}_t^{(0)} + \mathbf{Z}_{t+1}^\top (\boldsymbol{\upsilon}_t^{(1)} + \boldsymbol{\upsilon}_t^{(3)}) \mathbf{Z}_{t+1} + 2\mathbf{Z}_{t+1}^\top \boldsymbol{\upsilon}_t^{(2)}\right)$      (49)
    Track $\gamma_\rho(\mathcal{J}_b, \widehat{\theta}_t)$:   $\gamma_{t+1} = \gamma_t - \beta_{t+1} \Delta\gamma_{t+1}(\mathbf{Z}_{t+1})$      (50)
    Track $\Upsilon_1(\mathcal{J}_b, \widehat{\theta}_t)$:   $\xi_{t+1}^{(0)} = \xi_t^{(0)} + \beta_{t+1} \Delta\xi_{t+1}^{(0)}(\mathbf{Z}_{t+1})$      (51)
    Track $\Upsilon_2(\mathcal{J}_b, \widehat{\theta}_t)$:   $\xi_{t+1}^{(1)} = \xi_t^{(1)} + \beta_{t+1} \Delta\xi_{t+1}^{(1)}(\mathbf{Z}_{t+1})$      (52)
    if $\theta^p \neq NULL$ then
        $\left.\begin{array}{l} \mathbf{Z}_{t+1}^p \sim \widehat{f}_{\theta^p} \triangleq \lambda f_{\theta_0} + (1-\lambda) f_{\theta^p} \\ \gamma_{t+1}^p = \gamma_t^p - \beta_{t+1} \Delta\gamma_{t+1}(\mathbf{Z}_{t+1}^p) \end{array}\right\}$      (53)
    Compare Thresholds: $T_{t+1} = T_t + c(\mathbb{I}_{\{\gamma_{t+1} > \gamma_{t+1}^p\}} - \mathbb{I}_{\{\gamma_{t+1} \le \gamma_{t+1}^p\}} - T_t)$
    if $T_{t+1} > \epsilon_1$ then
        Save Old Model: $\gamma_{t+1}^p = \gamma_t; \quad \theta^p = \theta_t$
        Update Model: $\theta_{t+1} = \theta_t + \alpha_{t+1}\left((\xi_t^{(0)}, \xi_t^{(1)})^\top - \theta_t\right)$      (54)
        Reset Parameters: $T_t = 0; \quad c = c_t$      (55)
    else
        $\gamma_{t+1}^p = \gamma_t^p; \quad \theta_{t+1} = \theta_t$
    $t := t + 1;$
```

7 Convergence analysis

Observe that the algorithms are multi-timescale stochastic approximation algorithms (Borkar 1997) involving multiple stochastic recursions piggybacking each other. The primal recursions which typify the algorithms are the stochastic recursions which update the model parameters θ_t (Eq. (45) of Algorithm 1 and Eq. (54) of Algorithm 2), where the model parameters θ_t are calibrated to ensure their evolution towards the degenerate distribution concentrated on the global optimum (z_p^* for Algorithm 1 and z_b^* for Algorithm 2). Nonetheless, not disregarding the relevance of the remaining recursions which are all too vital and should augment each other and the primal recursion in achieving the desideratum. Therefore to analyze the limiting behaviour of the algorithms, one has to study the asymptotic behaviour of the individual recursions, *i.e.*, the effectiveness of the variables involved in tracking the true quantities. For analyzing the asymptotic behaviour of the algorithms, we apply the ODE based analysis from Ljung (1977), Kushner and Clark (1978), Kubrusly and Gravier (1973), Borkar (2008), Benveniste et al. (2012). In this method of analysis, for each individual stochastic recursion, we identify an associated ODE whose asymptotic (limiting) behaviour is similar to that of the stochastic recursion. In other words, the stochastic recursion eventually tracks the associated ODE. Subsequently, a qualitative analysis of the solutions of the associated ODE is performed to study their limiting behaviour and it is argued that the stochastic recursion asymptotically converges almost surely to the set of stable fixed points of the ODE (See Chapter 2 of Borkar (2008) or Chapter 5 of Kushner and Clark (1978) or Chapter 2 of Benveniste et al. 2012).

7.1 Outline of the proof

The roadmap followed in the analysis of the algorithms is as follows:

1. First and foremost, in the case of Algorithm 1, we study the asymptotic behaviour of the stochastic recursion (38). We show in Lemma 1 that the stochastic sequence $\{\omega_t\}$ indeed tracks the true quantity ω_* which defines the true objective function $\mathcal{J}_p$. Note that the recursion (38) is independent of other recursions and hence can be analyzed independently. The composition of the analysis (proof of Lemma 1) apropos of the limiting behaviour of $\{\omega_t\}$ involves mutlitple steps such as analyzing the nature of growth of the stochastic sequence, identifying the character of the implicit noise extant in the stochastic recursion, exploring the existence of finite bounds of the noise sequence (we solicit probabilistic analysis (Borkar 2012) to realize the above steps), ensuring with certainty the stability of the stochastic sequence (we appeal to Borkar–Meyn theorem Borkar 2008) and finally the qualitative analysis of the limit points of the associated ODE of the stochastic recursion (we seek assistance from dynamical systems theory Perko 2013).
2. Similarly, in the case of Algorithm 2, we study the asymptotic behaviour of the stochastic recursion (48). We show in Lemma 2 that the stochastic sequence $\{\upsilon_t\}$ certainly tracks the true quantity υ_* which defines the true objective function $\mathcal{J}_b$. The composition of the proof of Lemma 2 follows similar discourse as that of Lemma 1.
3. Since the proposed algorithms are multi-timescale stochastic approximation algorithms, their asymptotic behaviour depends heavily on the timescale differences induced by the step-size schedules $\{\alpha_t\}_{t\in\mathbb{N}}$ and $\{\beta_t\}_{t\in\mathbb{N}}$. The timescale differences allow the different individual recursions in a multi-timescale setting to learn at different rates. Since $\frac{\alpha_t}{\beta_t} \to 0$, the step-size $\{\beta_t\}_{t\in\mathbb{N}}$ decays to 0 at a relatively slower rate than $\{\alpha_t\}_{t\in\mathbb{N}}$ and therefore the increments in the recursions (40)–(42) which are controlled by β_t are relatively larger and hence appear to converge relatively faster than the recursions (38)–(39) and (45) which are controlled by α_t when viewed from the latter. So, considering a finite, yet sufficiently long time window, the relative evolution of the variables from the slower timescale α_t, *i.e.*, ω_t and θ_t to their steady-state form is indeed slow and in fact can be considered quasi-stationary when viewed from the evolutionary path of the faster timescale β_t. See Chapter 6 of Borkar (2008) for a succinct description on multi-timescale stochastic approximation algorithms. Hence, when viewed from the timescale of the recursions (40)–(42), one may consider ω_t and θ_t to be fixed. This is a standard technique used in analyzing multi-timescale stochastic approximation algorithms. Following this course

of analysis, we obtain Lemma 3 which characterizes the asymptotic behaviour of the stochastic recursions (40)–(42). The original paper (Joseph and Bhatnagar 2018) apropos of the stochastic approximation version of the CE method (proposed for a generalized optimization setting) establishes claims synonymous to Lemma 3 and hence we skip the proof of the lemma, nonetheless, we provide references to the same.

The results in Lemma 3 attest to validate that under the quasi-stationary hypothesis of $\omega_t \equiv \omega$ and $\theta_t \equiv \theta$, the stochastic sequence $\{\gamma_t\}$ tracks the true quantile $\gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})$ ((1) of Lemma 3), while the stochastic sequences $\{\xi_t^{(0)}\}$ and $\{\xi_t^{(1)}\}$ track the ideal CE model parameters $\Upsilon_1(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})$ and $\Upsilon_2(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})$ respectively ((2–3) of Lemma 3) with probability one. Certainly, these results establish that the stochastic recursions (40–42) track the ideal CE method and ergo, they provide a stable and proximal optimization gadget to minimize the error functions MSPBE (or MSBR). The rationale behind the pertinence of the stochastic recursion (44) is provided in Joseph and Bhatnagar (2018). Ostensibly, the purpose is as follows: The threshold sequence $\{\gamma_\rho(\mathcal{J}_p, \theta_t)\}$ (where θ_t is generated by Eq. (17)) of the ideal CE method is monotonically increasing (Proposition 2 of Joseph and Bhatnagar 2018). However, when stochastic approximation iterates are employed to track the ideal model parameters, the monotonicity may not hold always. The purpose of the stochastic recursion (44) is to ensure that the monotonicity of the threshold sequence is maintained and therefore (4–5) of Lemma 3 along with an appropriate choice of $\epsilon_1 \in [0, 1)$ (Algorithm 1) ensure that the model sequence $\{\theta_t\}$ is updated infinitely often.

4. Finally, we state our main results regarding the convergence of MSPBE and MSBR in Theorems 1 and 2, respectively. The theorems analyze the asymptotic behaviour of the model sequence $\{\theta_t\}_{t\in\mathbb{N}}$ for Algorithms 1 and 2 respectively. The theorems claim that the model sequence $\{\theta_t\}$ generated by Algorithm 1 (Algorithm 2) almost surely converges to $\theta_p^* = (z_p^*, 0_{k\times k})^\top$ ($\theta_b^* = (z_b^*, 0_{k\times k})^\top$), the degenerate distribution concentrated at z_p^* (z_b^*), where z_p^* (z_b^*) is the solution to the optimization problem (25) ((26)) which minimizes the error function MSPBE (MSBR).

7.2 The proof of convergence

For the stochastic recursion (38), we have the following result:
As a proviso, we define the filtration[9] $\{\mathcal{F}_t\}_{t\in\mathbb{N}}$, where the σ-field $\mathcal{F}_t \triangleq \sigma(\omega_i, \gamma_i, \gamma_i^p, \xi_i^{(0)}, \xi_i^{(1)}, \theta_i, 0 \leq i \leq t; \mathbf{Z}_i, 1 \leq i \leq t; \mathbf{s}_i, \mathbf{r}_i, \mathbf{s}'_i, 0 \leq i < t)$, $t \in \mathbb{N}$, is the σ-field generated by the specified random variables in the definition.

Lemma 1 *Let the step-size sequences $\{\alpha_t\}_{t\in\mathbb{N}}$ and $\{\beta_t\}_{t\in\mathbb{N}}$ satisfy Eq.* (36). *For the sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^\infty$, we let Assumption (A3) hold. Then, for a given $z \in \mathcal{Z}$, the sequence $\{\omega_t\}_{t\in\mathbb{N}}$ defined in Eq.* (38) *satisfies with probability one,*

$$\lim_{t\to\infty} \omega_t^{(0)} = \omega_*^{(0)}, \qquad \lim_{t\to\infty} \omega_t^{(1)} = \omega_*^{(1)},$$
$$\lim_{t\to\infty} \omega_t^{(2)} = \omega_*^{(2)} \quad \text{and} \quad \lim_{t\to\infty} \bar{\mathcal{J}}_p(\omega_t, z) = -MSPBE(z),$$

where $\omega_*^{(0)}$, $\omega_*^{(1)}$, $\omega_*^{(2)}$ and MSPBE are defined in Eq. (23)*, while $\bar{\mathcal{J}}_p(\omega_t, z)$ is defined in Eq.* (39).

[9] For detailed technical information pertaining to filtration and σ-field, refer Borkar (2012).

Proof By rearranging equations in (38), for $t \in \mathbb{N}$, we get

$$\omega_{t+1}^{(0)} = \omega_t^{(0)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,0)} + h^{(0,0)}(\omega_t^{(0)})\big), \tag{56}$$

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

$$\omega_{t+1}^{(1)} = \omega_t^{(1)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,1)} + h^{(0,1)}(\omega_t^{(1)})\big), \tag{57}$$

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

$$\omega_{t+1}^{(2)} = \omega_t^{(2)} + \alpha_{t+1}\big(\mathbb{M}_{t+1}^{(0,2)} + h^{(0,2)}(\omega_t^{(2)})\big), \tag{58}$$

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 $h^{(0,2)}(x) = \mathbb{I}_{k\times k} - \mathbb{E}\left[\phi_t\phi_t^\top x\right]$.

To apply the ODE based analysis, certain necessary conditions on the structural decorum are in order:

(B1) $h^{(0,j)}$, $0 \leq j \leq 2$ are Lipschitz continuous (easy to verify).

(B2) $\{\mathbb{M}_{t+1}^{(0,j)}\}_{t\in\mathbb{N}}$, $0 \leq j \leq 2$ are martingale difference noise sequences, *i.e.*, for each j, $\mathbb{M}_t^{(0,j)}$ is $\mathcal{F}_t$-measurable, integrable and $\mathbb{E}\left[\mathbb{M}_{t+1}^{(0,j)}|\mathcal{F}_t\right] = 0$, $t \in \mathbb{N}$, $0 \leq j \leq 2$.

(B3) Since ϕ_t, $\phi_t^{\prime}$ and $\mathbf{r}_t$ have uniformly bounded second moments, the noise sequences $\{\mathbb{M}_{t+1}^{(0,j)}\}_{t\in\mathbb{N}}$ have uniformly bounded second moments as well for each $0 \leq j \leq 2$ and hence $\exists K_{0,0}, K_{0,1}, K_{0,2} > 0$ *s.t.*

$$\mathbb{E}\left[\|\mathbb{M}_{t+1}^{(0,0)}\|^2\big|\mathcal{F}_t\right] \leq K_{0,0}(1 + \|\omega_t^{(0)}\|^2), \quad t \in \mathbb{N}. \tag{59}$$

$$\mathbb{E}\left[\|\mathbb{M}_{t+1}^{(0,1)}\|^2\big|\mathcal{F}_t\right] \leq K_{0,1}(1 + \|\omega_t^{(1)}\|^2), \quad t \in \mathbb{N}. \tag{60}$$

$$\mathbb{E}\left[\|\mathbb{M}_{t+1}^{(0,2)}\|^2\big|\mathcal{F}_t\right] \leq K_{0,2}(1 + \|\omega_t^{(2)}\|^2), \quad t \in \mathbb{N}. \tag{61}$$

(B4) To establish the stability (boundedness) condition, *i.e.*, $\sup_{t\in\mathbb{N}}\|\omega_t^{(j)}\| < \infty$ *a.s.*, for each $0 \leq j \leq 2$, we appeal to the Borkar–Meyn theorem (Theorem 2.1 of Borkar and Meyn (2000) or Theorem 7, Chapter 3 of Borkar 2008). Particularly, in order to prove $\sup_{t\in\mathbb{N}}\|\omega_t^{(0)}\| < \infty$ *a.s.*, we study the qualitative behaviour of the dynamical system defined by the following limiting ODE:

$$\frac{d}{dt}\omega^{(0)}(t) = h_\infty^{(0,0)}(\omega^{(0)}(t)), \quad t \in \mathbb{R}_+, \tag{62}$$

where

$$h_\infty^{(0,0)}(x) \triangleq \lim_{c\to\infty}\frac{h^{(0,0)}(cx)}{c} = \lim_{c\to\infty}\frac{\mathbb{E}\left[\mathbf{r}_t\phi_t\right] - cx}{c} = \lim_{c\to\infty}\frac{\mathbb{E}\left[\mathbf{r}_t\phi_t\right]}{c} - x = -x.$$

According to the Borkar–Meyn theorem, the global asymptotic stability of the above limiting system to the origin is sufficient to warrant the stability of the sequence $\{\omega_t^{(0)}\}_{t\in\mathbb{N}}$. Now, note that the ODE (62) is a linear, first-order ODE with negative rate of change and hence qualitatively the flow induced by the ODE is globally asymptotically stable to the origin. Therefore, we obtain the following:

$$\sup_{t\in\mathbb{N}}\|\omega_t^{(0)}\| < \infty \quad a.s. \tag{63}$$

Similarly we can show that

$$\sup_{t \in \mathbb{N}} \|\omega_t^{(1)}\| < \infty \quad a.s. \tag{64}$$

Now, regarding the stability of the sequence $\{\omega_t^{(2)}\}_{t \in \mathbb{N}}$, we consider the following limiting ODE:

$$\frac{d}{dt}\omega^{(2)}(t) = h_{\infty}^{(0,2)}(\omega^{(2)}(t)), \quad t \in \mathbb{R}_+, \tag{65}$$

where

$$\begin{aligned} h_{\infty}^{(0,2)}(x) \triangleq \lim_{c \to \infty} \frac{h^{(0,2)}(cx)}{c} &= \lim_{c \to \infty} \frac{\mathbb{I}_{k \times k} - \mathbb{E}\left[\phi_t \phi_t^\top cx\right]}{c} \\ &= \lim_{c \to \infty} \frac{\mathbb{I}_{k \times k}}{c} - x\mathbb{E}\left[\phi_t \phi_t^\top\right] = -x\mathbb{E}\left[\phi_t \phi_t^\top\right]. \end{aligned}$$

The system defined by the limiting ODE (65) is globally asymptotically stable to the origin since $\mathbb{E}[\phi_t \phi_t^\top]$ is positive definite (as it is positive semi-definite (easy to verify) and non-singular (from Assumption A3)). Therefore, by Borkar–Meyn theorem, we obtain the following:

$$\sup_{t \in \mathbb{N}} \|\omega_t^{(2)}\| < \infty \quad a.s. \tag{66}$$

Since we have hitherto established the necessary conditions (B1–B4), now by appealing to Theorem 2, Chapter 2 of Borkar (2008), we can directly establish the asymptotic equivalence between the individual stochastic recursions (56)–(58) and the following associated ODEs respectively.

$$\frac{d}{dt}\omega^{(0)}(t) = \mathbb{E}\left[\mathbf{r}_t \phi_t\right] - \omega^{(0)}(t), \quad t \in \mathbb{R}_+, \tag{67}$$

$$\frac{d}{dt}\omega^{(1)}(t) = \mathbb{E}\left[\phi_t (\gamma \phi_t' - \phi_t)^\top\right] - \omega^{(1)}(t)), \quad t \in \mathbb{R}_+, \tag{68}$$

$$\frac{d}{dt}\omega^{(2)}(t) = \mathbb{I}_{k \times k} - \mathbb{E}\left[\phi_t \phi_t^\top\right]\omega^{(2)}(t), \quad t \in \mathbb{R}_+. \tag{69}$$

Now we study the qualitative behaviour of the above system of first-order, linear ODEs. A simple examination of the trajectories of the ODEs reveals that the point $\mathbb{E}\left[\mathbf{r}_t \phi_t\right]$ is a globally asymptotically stable equilibrium point of the ODE (67). Similarly for the ODE (68), the point $\mathbb{E}\left[\phi_t (\gamma \phi_t' - \phi_t)^\top\right]$ is a globally asymptotically stable equilibrium. Finally, regarding the limiting behaviour of the ODE (69), we find that the point $\mathbb{E}\left[\phi_t \phi_t^\top\right]^{-1}$ is a globally asymptotically stable equilibrium. This follows since $\mathbb{E}\left[\phi_t \phi_t^\top\right]$ is positive semi-definite (easy to verify) and non-singular (from Assumption A3). Formally,

$$\lim_{t \to \infty} \omega^{(0)}(t) = \mathbb{E}\left[\mathbf{r}_t \phi_t\right], \tag{70}$$

$$\lim_{t \to \infty} \omega^{(1)}(t) = \mathbb{E}\left[\phi_t (\gamma \phi_t' - \phi_t)^\top\right], \tag{71}$$

$$\lim_{t \to \infty} \omega^{(2)}(t) = \mathbb{E}\left[\phi_t \phi_t^\top\right]^{-1}, \tag{72}$$

where the above convergence is achieved independent of the initial values $\omega^{(0)}(0)$, $\omega^{(1)}(0)$ and $\omega^{(2)}(0)$.

Therefore, by the employing the asymptotic equivalence of the stochastic recursions (56)–(58) and their associated ODEs (67)–(69) we obtain the following:

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

Putting all the above together, we get, for $z \in \mathcal{Z}$, $\lim_{t\to\infty}\bar{\mathcal{J}}_p(\omega_t, z) = \bar{\mathcal{J}}_p(\omega_*, z) = \mathcal{J}_p(z)$ *a.s.* □

For the stochastic recursion (48), we have the following result:
Again, as a proviso, we define the filtration $\{\mathcal{F}_t\}_{t\in\mathbb{N}}$ where the σ-field $\mathcal{F}_t \triangleq \sigma(\upsilon_i, \gamma_i, \gamma_i^p, \xi_i^{(0)}, \xi_i^{(1)}, \theta_i, 0 \le i \le t; \mathbf{Z}_i, 1 \le i \le t; \mathbf{s}_i, \mathbf{r}_i, \mathbf{r}_i', \mathbf{s}_i', \mathbf{s}_i'', 0 \le i < t)$, $t \in \mathbb{N}$.

Lemma 2 *Let the step-size sequences $\{\alpha_t\}_{t\in\mathbb{N}}$ and $\{\beta_t\}_{t\in\mathbb{N}}$ satisfy Eq.* (36)*. For the sample trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{r}_t', \mathbf{s}_t', \mathbf{s}_t'')\}_{t=0}^{\infty}$, we let Assumption $(A3)'$ hold. Then, for a given $z \in \mathcal{Z}$, the sequence $\{\upsilon_t\}_{t\in\mathbb{N}}$ defined in Eq.* (48) *satisfies with probability one,*

$$\begin{aligned}
&\lim_{t\to\infty}\upsilon_t^{(0)} = \upsilon_*^{(0)}, \quad \lim_{t\to\infty}\upsilon_t^{(1)} = \upsilon_*^{(1)}, \quad \lim_{t\to\infty}\upsilon_t^{(2)} = \upsilon_*^{(2)},\\
&\lim_{t\to\infty}\bar{\mathcal{J}}_b(\upsilon_t, z) = -MSBR(z)
\end{aligned}$$

where $\upsilon_*^{(0)}$, $\upsilon_*^{(1)}$, $\upsilon_*^{(2)}$, $\upsilon_*^{(3)}$ and MSBR are defined in Eq. (24)*, while $\bar{\mathcal{J}}_b(\upsilon_t, z)$ is defined in Eq.* (49).

Proof By rearranging equations in (48), for $t \in \mathbb{N}$, we get

$$\upsilon_{t+1}^{(0)} = \upsilon_t^{(0)} + \alpha_{t+1}\left(\mathbb{M}_{t+1}^{(1,0)} + h^{(1,0)}(\upsilon_t^{(0)})\right), \tag{73}$$

where $\mathbb{M}_{t+1}^{(1,0)} = \mathbf{r}_t\mathbf{r}_t' - \mathbb{E}^2\left[\mathbf{r}_t\right]$ and $h^{(1,0)}(x) = \mathbb{E}^2\left[\mathbf{r}_t\right] - x$.
Similarly,

$$\upsilon_{t+1}^{(1)} = \upsilon_t^{(1)} + \alpha_{t+1}\left(\mathbb{M}_{t+1}^{(1,1)} + h^{(1,1)}(\upsilon_t^{(1)})\right), \tag{74}$$

where $\mathbb{M}_{t+1}^{(1,1)} = \gamma^2\phi_t'\phi_t''^\top - \gamma^2\mathbb{E}\left[\phi_t'\right]\mathbb{E}\left[\phi_t'\right]^\top$ and $h^{(1,1)}(x) = \gamma^2\mathbb{E}\left[\phi_t'\right]\mathbb{E}\left[\phi_t'\right]^\top - x$. Also,

$$\upsilon_{t+1}^{(2)} = \upsilon_t^{(2)} + \alpha_{t+1}\left(\mathbb{M}_{t+1}^{(1,2)} + h^{(1,2)}(\upsilon_t^{(2)})\right), \tag{75}$$

where $\mathbb{M}_{t+1}^{(1,2)} = \mathbf{r}_t\left(\phi_t' - \phi_t\right) - \mathbb{E}\left[\mathbf{r}_t\left(\phi_t' - \phi_t\right)\right]$ and $h^{(1,2)}(x) = \mathbb{E}\left[\mathbf{r}_t\left(\phi_t' - \phi_t\right)\right] - x$. Finally,

$$\upsilon_{t+1}^{(3)} = \upsilon_t^{(3)} + \alpha_{t+1}\left(\mathbb{M}_{t+1}^{(1,3)} + h^{(1,3)}(\upsilon_t^{(3)})\right), \tag{76}$$

where $\mathbb{M}_{t+1}^{(1,3)} = \left(\phi_t - 2\gamma\phi_t'\right)\phi_t^\top - \mathbb{E}\left[\left(\phi_t - 2\gamma\phi_t'\right)\phi_t^\top\right]$ and $h^{(1,3)}(x) = \mathbb{E}\left[\left(\phi_t - 2\gamma\phi_t'\right)\phi_t^\top\right] - x$.
To apply the ODE based analysis, certain necessary conditions on the structural decorum are in order:

(C1) $h^{(1,j)}, 0 \le j \le 3$ are Lipschitz continuous (easy to verify).

(C2) $\{\mathbb{M}_{t+1}^{(1,j)}\}_{t\in\mathbb{N}}, 0 \le j \le 3$ are martingale difference noise sequences, *i.e.*, for each $t \in \mathbb{N}$, $\mathbb{M}_t^{(1,j)}$ is $\mathcal{F}_t$-measurable, integrable and $\mathbb{E}\left[\mathbb{M}_{t+1}^{(1,j)}|\mathcal{F}_t\right] = 0$, $t \in \mathbb{N}$, $0 \le j \le 3$.

(C3) Since ϕ_t, ϕ'_t, ϕ''_t, $\mathbf{r}_t$ and $\mathbf{r}'_t$ have uniformly bounded second moments, the noise sequences $\{\mathbb{M}^{(1,j)}_{t+1}\}_{t\in\mathbb{N}}$, $0 \leq j \leq 3$ have uniformly bounded second moments as well and hence $\exists K_{1,0}, K_{1,1}, K_{1,2}, K_{1,3} > 0$ *s.t.*

$$\mathbb{E}\left[\|\mathbb{M}^{(1,0)}_{t+1}\|^2|\mathcal{F}_t\right] \leq K_{1,0}(1 + \|\upsilon^{(0)}_t\|^2), t \in \mathbb{N}. \tag{77}$$

$$\mathbb{E}\left[\|\mathbb{M}^{(1,1)}_{t+1}\|^2|\mathcal{F}_t\right] \leq K_{1,1}(1 + \|\upsilon^{(1)}_t\|^2), t \in \mathbb{N}. \tag{78}$$

$$\mathbb{E}\left[\|\mathbb{M}^{(1,2)}_{t+1}\|^2|\mathcal{F}_t\right] \leq K_{1,2}(1 + \|\upsilon^{(2)}_t\|^2), t \in \mathbb{N}. \tag{79}$$

$$\mathbb{E}\left[\|\mathbb{M}^{(1,3)}_{t+1}\|^2|\mathcal{F}_t\right] \leq K_{1,3}(1 + \|\upsilon^{(3)}_t\|^2), t \in \mathbb{N}. \tag{80}$$

(C4) To establish the stability condition, *i.e.*, $\sup_{t\in\mathbb{N}} \|\upsilon^{(j)}_t\| < \infty\ a.s.$, for each $0 \leq j \leq 3$, we appeal to the Borkar–Meyn theorem (Theorem 2.1 of Borkar and Meyn (2000) or Theorem 7, Chapter 3 of Borkar 2008). Indeed to prove $\sup_{t\in\mathbb{N}} \|\upsilon^{(0)}_t\| < \infty\ a.s.$, we consider the dynamical system defined by the following ∞-system ODE:

$$\frac{d}{dt}\upsilon^{(0)}(t) = h^{(1,0)}_{\infty}(\upsilon^{(0)}(t)), \tag{81}$$

where

$$h^{(1,0)}_{\infty}(x) \triangleq \lim_{c\to\infty} \frac{h^{(1,0)}(cx)}{c} = \lim_{c\to\infty} \frac{\mathbb{E}^2\left[\mathbf{r}_t\right] - cx}{c} = \lim_{c\to\infty} \frac{\mathbb{E}^2\left[\mathbf{r}_t\right]}{c} - x = -x.$$

It is easy to verify that the above flow (81) is globally asymptotically stable to the origin. Therefore, by appealing to the Borkar–Meyn theorem, we obtain that the iterates $\{\upsilon^{(0)}_t\}_{t\in\mathbb{N}}$ are almost surely stable, *i.e.*,

$$\sup_{t\in\mathbb{N}} \|\upsilon^{(0)}_t\| < \infty \quad a.s. \tag{82}$$

Similarly we can show that

$$\sup_{t\in\mathbb{N}} \|\upsilon^{(1)}_t\| < \infty \quad a.s. \tag{83}$$

$$\sup_{t\in\mathbb{N}} \|\upsilon^{(2)}_t\| < \infty \quad a.s. \tag{84}$$

$$\sup_{t\in\mathbb{N}} \|\upsilon^{(3)}_t\| < \infty \quad a.s. \tag{85}$$

Since we have hitherto established the necessary conditions (C1–C4), now by appealing to Theorem 2, Chapter 2 of Borkar (2008), we can forthwith guarantee the asymptotic equivalence between the recursion (73) and the following ODE (*i.e.*, the recursion (73) asymptotically tracks the following ODE):

$$\frac{d}{dt}\upsilon^{(0)}(t) = \mathbb{E}^2\left[\mathbf{r}_t\right] - \upsilon^{(0)}(t), \quad t \in \mathbb{R}_+. \tag{86}$$

Similarly, we can guarantee the independent asymptotic equivalences between the recursions (74)–(76) and the ODEs (87)–(89) respectively.

$$\frac{d}{dt}\upsilon^{(1)}(t) = \gamma^2\mathbb{E}\left[\phi'_t\right]\mathbb{E}\left[\phi'_t\right]^\top - \upsilon^{(1)}(t), \quad t \in \mathbb{R}_+, \tag{87}$$

$$\frac{d}{dt}\upsilon^{(2)}(t) = \mathbb{E}\left[\mathbf{r}_t\left(\phi'_t - \phi_t\right)\right] - \upsilon^{(2)}(t), \quad t \in \mathbb{R}_+, \tag{88}$$

$$\frac{d}{dt}\upsilon^{(3)}(t) = \mathbb{E}\left[\left(\phi_t - 2\gamma\phi'_t\right)\phi_t^\top\right] - \upsilon^{(3)}(t), \quad t \in \mathbb{R}_+. \tag{89}$$

Note that all the above ODEs (86)–(89) are linear, first-order ODEs and further qualitative analysis reveals that the individual flows defined by the various ODEs are globally asymptotically stable. An examination of the trajectories of the ODEs attests that the limiting behaviour of the individual flows defined by the ODEs (86)–(89) satisfies the following:

$$\left.\begin{aligned}
&\upsilon^{(0)}(t) \to \mathbb{E}^2\left[\mathbf{r}_t\right] \quad \text{as } t \to \infty.\\
&\upsilon^{(1)}(t) \to \gamma^2\mathbb{E}\left[\phi'_t\right]\mathbb{E}\left[\phi'_t\right]^\top \quad \text{as } t \to \infty.\\
&\upsilon^{(2)}(t) \to \mathbb{E}\left[\mathbf{r}_t\left(\phi'_t - \phi_t\right)\right] \quad \text{as } t \to \infty.\\
&\upsilon^{(3)}(t) \to \mathbb{E}\left[\left(\phi_t - 2\gamma\phi'_t\right)\phi_t^\top\right] \quad \text{as } t \to \infty.
\end{aligned}\right\} \tag{90}$$

Finally, Eq. (90) and the previously established asymptotic equivalence between the recursions (73)–(76) and their respective associated ODEs (86)–(89) ascertains the following:

$$\begin{aligned}
&\lim_{t\to\infty}\upsilon_t^{(0)} = \mathbb{E}^2\left[\mathbf{r}_t\right] \; a.s. \; = \upsilon_*^{(0)}.\\
&\lim_{t\to\infty}\upsilon_t^{(1)} = \gamma^2\mathbb{E}\left[\phi'_t\right]\mathbb{E}\left[\phi'_t\right]^\top \; a.s. \; = \upsilon_*^{(1)}.\\
&\lim_{t\to\infty}\upsilon_t^{(2)} = \mathbb{E}\left[\mathbf{r}_t\left(\phi'_t - \phi_t\right)\right] \; a.s. \; = \upsilon_*^{(2)}.\\
&\lim_{t\to\infty}\upsilon_t^{(3)} = \mathbb{E}\left[\left(\phi_t - 2\gamma\phi'_t\right)\phi_t^\top\right] \; a.s. \; = \upsilon_*^{(3)}.
\end{aligned}$$

Putting all the above together, we get, for $z \in \mathcal{Z}$,

$$\lim_{t\to\infty}\bar{\mathcal{J}}_b(\upsilon_t, z) = \bar{\mathcal{J}}_b(\upsilon_*, z) = \mathcal{J}_b(z) \quad a.s.$$

□

Notation: We denote by $\mathbb{E}_{\widehat{\theta}}[\cdot]$ the expectation *w.r.t.* the mixture PDF $\widehat{f_\theta}$ and $\mathbb{P}_{\widehat{\theta}}$ denotes its induced probability measure. Also, $\gamma_\rho(\cdot, \widehat{\theta})$ represents the $(1-\rho)$-quantile *w.r.t.* the mixture PDF $\widehat{f_\theta}$.

The following result characterizes the asymptotic behaviour of the stochastic recursions (40-42):

Lemma 3 *Assume $\omega_t \equiv \omega$, $\theta_t \equiv \theta$, $\forall t \in \mathbb{N}$. Let Assumption A2 hold. Also, let the step-size sequences $\{\alpha_t\}_{t\in\mathbb{N}}$ and $\{\beta_t\}_{t\in\mathbb{N}}$ satisfy Eq. (36). Then,*

1. *The sequence $\{\gamma_t\}_{t\in\mathbb{N}}$ generated by Eq. (40) satisfies*

$$\lim_{t\to\infty}\gamma_t = \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta}) \quad a.s.$$

2. *The sequence $\{\xi_t^{(0)}\}_{t\in\mathbb{N}}$ generated by Eq. (41) satisfies*

$$\lim_{t\to\infty}\xi_t^{(0)} = \xi_{\omega,\theta}^{(0)} = \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_1\left(\bar{\mathcal{J}}_p(\omega, \mathbf{Z}), \mathbf{Z}, \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_0\left(\bar{\mathcal{J}}_p(\omega, \mathbf{Z}), \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})\right)\right]} \; a.s.$$

3. *The sequence $\{\xi_t^{(1)}\}_{t\in\mathbb{N}}$ generated by Eq. (42) satisfies*

$$\lim_{t\to\infty}\xi_t^{(1)} = \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_2\left(\bar{\mathcal{J}}_p(\omega, \mathbf{Z}), \mathbf{Z}, \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta}), \xi_{\omega,\theta}^{(0)}\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_0\left(\bar{\mathcal{J}}_p(\omega, \mathbf{Z}), \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta})\right)\right]} \; a.s.$$

4. *For any $T_0 \in (0, 1)$, $\{T_t\}_{t \in \mathbb{N}}$ generated by Eq.* (44) *satisfies $T_t \in (-1, 1)$, $\forall t \in \mathbb{N}$.*
5. *If $\gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta}) > \gamma_\rho(\bar{\mathcal{J}}_p(\omega, \cdot), \widehat{\theta^p})$, then $\{T_t\}_{t \in \mathbb{N}}$ generated by Eq.* (44) *satisfies $\lim_{t \to \infty} T_t = 1$ a.s.*

Proof Please refer to the proofs of Proposition 1, Lemmas 2 and 3 in Joseph and Bhatnagar (2018). □

Remark 3 Similar results can also be obtained for Algorithm 2 with $\bar{\mathcal{J}}_p$ replaced by $\bar{\mathcal{J}}_b$ and ω replaced by υ.

Finally, we analyze the asymptotic behaviour of the model sequence $\{\theta_t\}_{t \in \mathbb{N}}$. As a preliminary requirement, we define $\Psi_p(\omega, \theta) = (\Psi_p^{(0)}(\omega, \theta), \Psi_p^{(1)}(\omega, \theta))^\top$, where

$$\Psi_p^{(0)}(\omega, \theta) \triangleq \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_1\left(\mathcal{J}_p(\omega, \mathbf{Z}), \mathbf{Z}, \gamma_\rho(\mathcal{J}_p(\omega, \cdot), \widehat{\theta})\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_0\left(\mathcal{J}_p(\omega, \mathbf{Z}), \gamma_\rho(\mathcal{J}_p(\omega, \cdot), \widehat{\theta})\right)\right]}, \tag{91}$$

$$\Psi_p^{(1)}(\omega, \theta) \triangleq \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_2\left(\mathcal{J}_p(\omega, \mathbf{Z}), \mathbf{Z}, \gamma_\rho(\mathcal{J}_p(\omega, \cdot), \widehat{\theta}), \Psi_p^{((0)}(\omega, \theta)\right)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g}_0\left(\mathcal{J}_p(\omega, \mathbf{Z}), \gamma_\rho(\mathcal{J}_p(\omega, \cdot), \widehat{\theta})\right)\right]}. \tag{92}$$

Similarly, we define Ψ_b with $\mathcal{J}_p$ replaced by $\mathcal{J}_b$ and ω replaced by υ.

We now state our main theorems. The first theorem states that the model sequence $\{\theta_t\}_{t \in \mathbb{N}}$ generated by Algorithm 1 almost surely converges to $\theta^{p*} = (z_p^*, 0_{k \times k})^\top$, the degenerate distribution concentrated at z_p^*, where z_p^* is the solution to the optimization problem (25) which minimizes the error function MSPBE.

Theorem 1 (MSPBE Convergence) *Let $S(z) = exp(rz)$, $r \in \mathbb{R}_+$. Let $\rho \in (0, 1)$ and $\lambda \in (0, 1)$. Let $\theta_0 = (\mu_0, qI_{k \times k})^\top$, where $q \in \mathbb{R}_+$. Let the step-size sequences $\{\alpha_t\}_{t \in \mathbb{N}}$, $\{\beta_t\}_{t \in \mathbb{N}}$ satisfy Eq.* (36)*. 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 $\theta_t \in \Theta$, $\forall t \in \mathbb{N}$. Also, let the Assumptions (A1), (A2) and (A3) hold. Further, we assume that there exists a continuously differentiable function $V : U \to \mathbb{R}_+$, where $U \subseteq \Theta$ is an open neighbourhood of θ^{p*} with $\nabla V(\theta)^\top \Psi_p(\omega_*, \theta) < 0$, $\forall \theta \in U \setminus \{\theta^{p*}\}$ and $\nabla V(\theta^{p*})^\top \Psi_p(\omega_*, \theta^{p*}) = 0$. Then, there exists $q^* \in \mathbb{R}_+$ and $r^* \in \mathbb{R}_+$ s.t. $\forall q > q^*$ and $\forall r > r^*$,*

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

where $\mathcal{J}_p^*$ and z_p^* are defined in Eq. (25)*. Further, since $\mathcal{J}_p = -MSPBE$, the algorithm SCE-MSPBEM converges to the global minimum of MSPBE a.s.*

Proof Please refer to the proof of Theorem 1 in Joseph and Bhatnagar (2018). □

Similarly for Algorithm 2, the following theorem states that the model sequence $\{\theta_t\}_{t \in \mathbb{N}}$ generated by Algorithm 2 almost surely converges to $\theta^{b*} = (z_b^*, 0_{k \times k})^\top$, the degenerate distribution concentrated at z_b^*, where z_b^* is the solution to the optimization problem (26) which minimizes the error function MSBR.

Theorem 2 (MSBR Convergence) *Let $S(z) = exp(rz)$, $r \in \mathbb{R}_+$. Let $\rho \in (0, 1)$ and $\lambda \in (0, 1)$. Let $\theta_0 = (\mu_0, qI_{k \times k})^\top$, where $q \in \mathbb{R}_+$. Let the step-size sequences $\{\alpha_t\}_{t \in \mathbb{N}}$, $\{\beta_t\}_{t \in \mathbb{N}}$ satisfy Eq.* (36)*. Also let $c_t \to 0$. Suppose $\{\theta_t = (\mu_t, \Sigma_t)^\top\}_{t \in \mathbb{N}}$ is the sequence generated by Algorithm* 2 *and assume $\theta_t \in \Theta$, $\forall t \in \mathbb{N}$. Also, let the Assumptions (A1), (A2) and (A3)′ hold. Further, we assume that there exists a continuously differentiable function $V : U \to \mathbb{R}_+$, where $U \subseteq \Theta$ is an open neighbourhood of θ^{b*} with $\nabla V(\theta)^\top \Psi_b(\upsilon_*, \theta) < 0$, $\forall \theta \in U \setminus \{\theta^{b*}\}$ and $\nabla V(\theta^{b*})^\top \Psi_b(\upsilon_*, \theta^{b*}) = 0$. Then, there exists $q^* \in \mathbb{R}_+$ and $r^* \in \mathbb{R}_+$ s.t. $\forall q > q^*$ and $\forall r > r^*$,*

$$\lim_{t\to\infty} \bar{\mathcal{J}}_b(\upsilon_t, \mu_t) = \mathcal{J}_b^* \quad \textit{and} \quad \lim_{t\to\infty} \theta_t = \theta^{b*} = (z_b^*, 0_{k\times k})^\top a.s.,$$

where $\mathcal{J}_b^*$ and z_b^* are defined in Eq. (26). Further, since $\mathcal{J}_b = -MSBR$, the algorithm SCE-MSBRM converges to the global minimum of MSBR a.s.

Proof Please refer to the proof of Theorem 1 in Joseph and Bhatnagar (2018). □

7.3 Discussion of the proposed algorithms

The computational load of the algorithms SCE-MSPBEM and SCE-MSBRM is $\Theta(k^2)$ per iteration which is primarily attributed to the computation of Eqs. (38) and (48) respectively. Least squares algorithms like LSTD and LSPE also require $\Theta(k^2)$ per iteration. However, LSTD requires an extra operation of inverting the $k \times k$ matrix A_T (Algorithm 7) which requires an extra computational effort of $\Theta(k^3)$. (Note that LSPE also requires a $k \times k$ matrix inversion.) This makes the *overall complexity of LSTD and LSPE to be $\Theta(k^3)$*. Further in some cases the matrix A_T may not be invertible. In that case, the pseudo inverse of A_T needs to be obtained in LSTD and LSPE which is computationally even more expensive. Our algorithm does not require such an inversion procedure. Also *even though the complexity of the first order temporal difference algorithms such as TD(λ) and GTD2 is $\Theta(k)$, the approximations they produced in the experiments we conducted turned out to be inferior to ours and also showed a slower rate of convergence than our algorithm.* Another noteworthy characteristic exhibited by our algorithm is *stability*. Recall that the convergence of TD(0) is guaranteed by the requirements that the Markov chain of P^π should be ergodic with the sampling distribution ν as its stationary distribution. The classic example of Baird's 7-star (Baird 1995) violates those restrictions and hence TD(0) is seen to diverge. However, our algorithm does not impose such restrictions and shows stable behaviour even in non-ergodic cases such as the Baird's example.

But the significant feature of SCE-MSPBEM/SCE-MSBRM is its ability to find the global optimum. This particular characteristic of the algorithm enables it to produce high quality solutions when applied to non-linear function approximation, where the convexity of the objective function does not hold in general. Also note that SCE-MSPBEM/SCE-MSBRM is a gradient-free technique and hence does not require strong structural restrictions on the objective function.

8 Experimental results

We present here a numerical comparison of our algorithms with various state-of-the-art algorithms in the literature on some benchmark reinforcement learning problems. In each of the experiments, a random trajectory $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t)\}_{t=0}^{\infty}$ is chosen and all the algorithms are updated using it. Each $\mathbf{s}_t$ in $\{(\mathbf{s}_t, \mathbf{r}_t, \mathbf{s}'_t), t \geq 0\}$ is sampled using an arbitrary distribution ν over $\mathbb{S}$. The algorithms are run on 10 independent trajectories and the average of the results obtained is 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. In all the test cases, the evolution of the model sequence $\{\theta_t\}$ across independent trials was almost homogeneous and hence we omit the standard error bars from our plots.

We evaluated the performance of our algorithms on the following benchmark problems:

1. Linearized cart-pole balancing (Dann et al. 2014).
2. 5-Link actuated pendulum balancing (Dann et al. 2014).

3. Baird's 7-star MDP (Baird 1995).
4. 10-state ring MDP (Kveton et al. 2006).
5. MDPs with radial basis functions and Fourier basis functions (Konidaris et al. 2011).
6. Settings involving non-linear function approximation (Tsitsiklis and Roy 1997).

8.1 Experiment 1: linearized cart-pole balancing (Dann et al. 2014)

- **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 (Fig. 2).
- **Goal:** To balance the pole upright and the cart at the centre of the track.
- **State space:** The 4-tuple $(x, \dot{x}, \psi, \dot{\psi})^\top \in \mathbb{R}^4$, where ψ is the angle of the pendulum with respect to 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 $a \in \mathbb{R}$ on the cart parallel to the track. The stochastic policy used in this setting corresponds to $\pi(a|s) = \mathcal{N}(a|\beta_1^\top s, \sigma_1^2)$, where $\beta_1 \in \mathbb{R}^4$ and $\sigma_1 \in \mathbb{R}$.
- **System dynamics:** The dynamical equations of the system are given by

$$\ddot{\psi} = \frac{-3ml\dot{\psi}^2 \sin\psi \cos\psi + (6M + m)g \sin\psi - 6(a - b\dot{\psi})\cos\psi}{4l(M + m) - 3ml \cos\psi}, \tag{93}$$

$$\ddot{x} = \frac{-2ml\dot{\psi}^2 \sin\psi + 3mg \sin\psi \cos\psi + 4a - 4b\dot{\psi}}{4(M + m) - 3m \cos\psi}. \tag{94}$$

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

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \\ \psi_{t+1} \\ \dot{\psi}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ \dot{x}_t \\ \psi_t \\ \dot{\psi}_t \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\psi}_t \\ \frac{3(M+m)\psi_t - 3a + 3b\dot{\psi}_t}{4Ml - ml} \\ \dot{x}_t \\ \frac{3mg\psi_t + 4a - 4b\dot{\psi}_t}{4M - m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{z} \end{bmatrix}, \tag{95}$$

  where Δt is the integration time step, *i.e.*, the time difference between two transitions and $\mathbf{z}$ is a Gaussian noise on the velocity of the cart with standard deviation σ_2.
- **Reward function:** $\mathrm{R}(s, a) = \mathrm{R}(\psi, \dot{\psi}, x, \dot{x}, a) = -100\psi^2 - x^2 - \frac{1}{10}a^2$.
- **Feature vectors:** $\phi(s \in \mathbb{R}^4) = (1, s_1^2, s_2^2 \ldots, s_1 s_2, s_1 s_3, \ldots, s_3 s_4)^\top \in \mathbb{R}^{11}$.
- **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.*, $V^{\pi^*} \in \{\Phi z | z \in \mathbb{R}^k\}$.

Here the sample trajectory is obtained by a continuous roll-out of a particular realization of the underlying Markov chain and hence it is of on-policy nature. Therefore, the sampling distribution is the stationary distribution (steady-state distribution) of the Markov chain induced by the policy being evaluated (see Remark 1). The various parameter values we used in our experiment are provided in Table 3 of "Appendix". The results of the experiments are shown in Fig. 3.

8.2 Experiment 2: 5-link actuated pendulum balancing (Dann et al. 2014)

- **Setup:** 5 independent poles each with mass m and length l with the top pole being a pendulum connected using 5 rotational joints (Fig. 4).

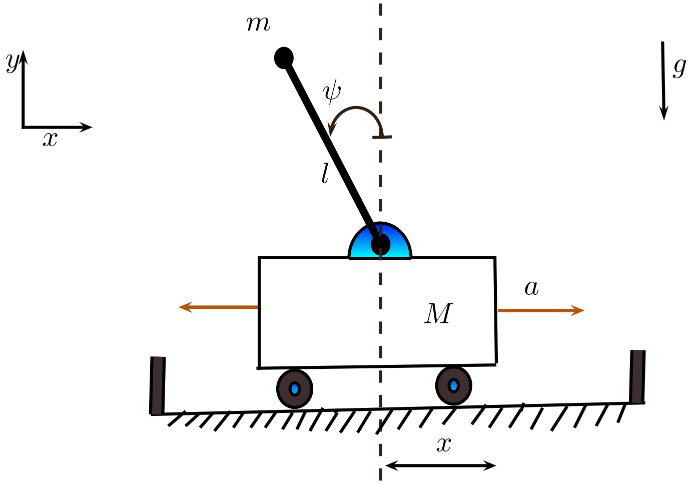


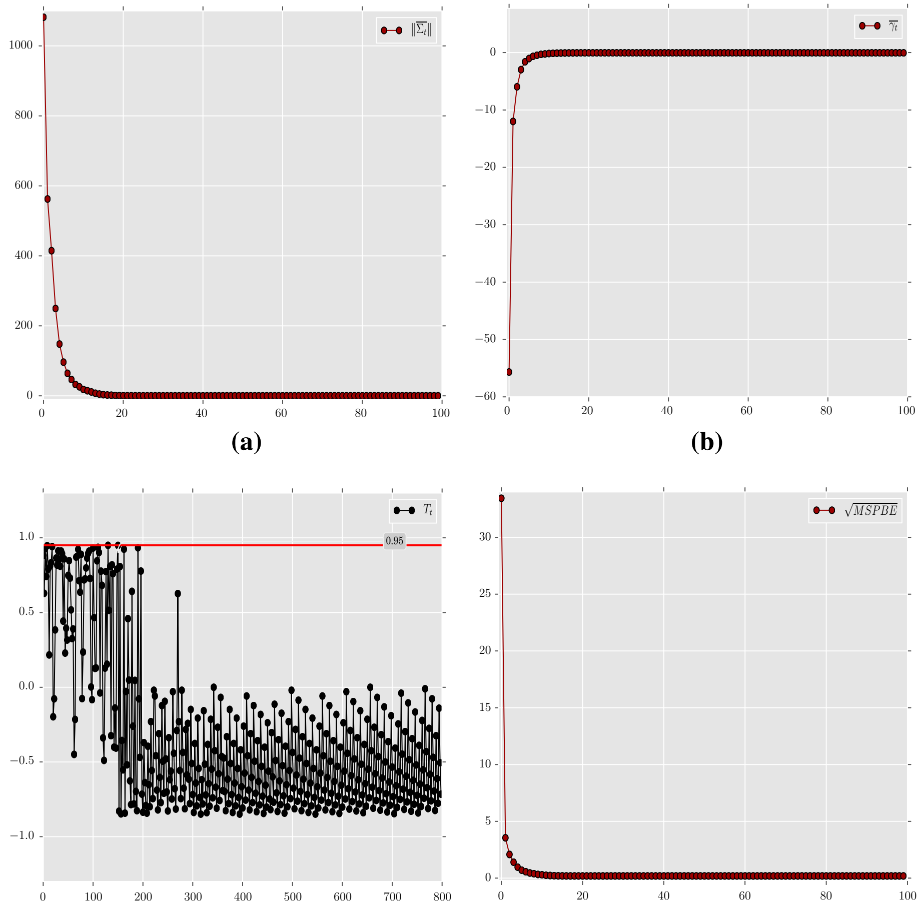
Fig. 2 The cart-pole system. The goal is to keep the pole in the upright position and the cart at the center of the track by pushing the cart with a force a either to the left or the right. The system is parametrized by the position x of the cart, the angle of the pole ψ, the velocity $\dot{x}$ and the angular velocity $\dot{\psi}$

- **Goal:** To keep all the poles in the upright position by applying independent torques at each joint.
- **State space:** The state $s = (q, \dot{q})^\top \in \mathbb{R}^{10}$, 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 being the angle of the pole i with respect to the vertical axis and $\dot{\psi}_i$ 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 $\pi(a|s) = \mathcal{N}_5(a|\beta_1^\top s, \sigma_1^2)$, where $\beta_1 \in \mathbb{R}^{10\times 5}$ and $\sigma_1 \in \mathbb{R}^{5\times 5}$.
- **System dynamics:** The approximate linear system dynamics is given by

$$\begin{bmatrix} q_{t+1} \\ \dot{q}_{t+1} \end{bmatrix} = \begin{bmatrix} I & \Delta t \, I \\ -\Delta t \, M^{-1} U & I \end{bmatrix} \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} + \Delta t \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} a + \mathbf{z}, \tag{96}$$

  where Δt is the integration time step, *i.e.*, the time difference between two transitions, M is the mass matrix in the upright position where $M_{ij} = l^2(6 - max(i, j))m$ and U is a diagonal matrix with $U_{ii} = -gl(6 - i)m$. Each component of $\mathbf{z}$ is a Gaussian noise.
- **Reward function:** $\mathrm{R}(q, \dot{q}, a) = -q^\top q$.
- **Feature vectors:** $\phi(s \in \mathbb{R}^{10}) = (1, s_1^2, s_2^2 \ldots, s_1 s_2, s_1 s_3, \ldots, s_9 s_{10})^\top \in \mathbb{R}^{46}$.
- **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.*, $V^{\pi^*} \in \{\Phi z | z \in \mathbb{R}^k\}$.

Similar to the earlier experiment, here also the sample trajectory is of on-policy nature and therefore the sampling distribution is the steady-state distribution of the Markov chain induced by the policy being evaluated (see Remark 1). The various parameter values we used in our experiment are provided in Table 4 of Appendix. Note that we have used constant step-sizes in this experiment. The results of the experiment are shown in Fig. 5.



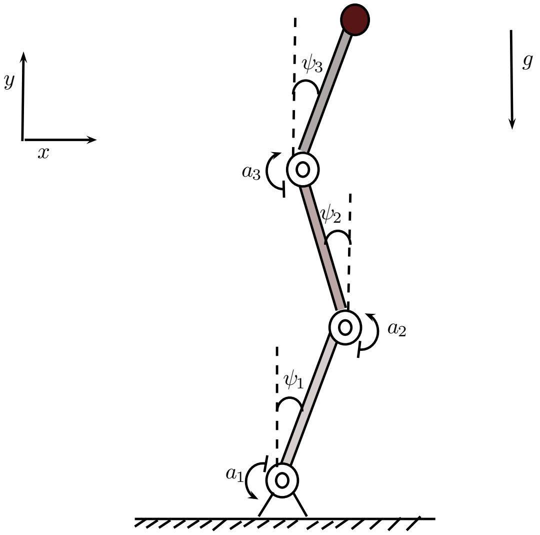
(a) **(b)** **(c)** **(d)**

Fig. 3 The cart-pole setting. The evolutionary trajectory of the variables $\|\Sigma_t\|_F$ (where $\|\cdot\|_F$ is the Frobenius norm), γ_t^p, T_t and $\sqrt{\text{MSPBE}(\mu_t)}$. Note that both γ_t^p and $\sqrt{\text{MSPBE}(\mu_t)}$ 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^*. The evolutionary track of T_t shows that T_t does not cross the $\epsilon_1 = 0.95$ line after the model $\theta_t = (\mu_t, \Sigma_t)^\top$ reaches a close neighbourhood of its limit. **a** Plot of $\|\Sigma_t\|_F$ (where $\|\cdot\|_F$ is the Frobenius norm). **b** Plot of γ_t^p. **c** Plot of T_t. **d** Plot of $\sqrt{\text{MSPBE}(\mu_t)}$

8.3 Experiment 3: Baird's 7-star MDP (Baird 1995)

Our algorithm was also tested on Baird's star problem (Baird 1995). 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 (Baird 1995). We consider here an MDP with $|\mathbb{S}| = 7$, $|\mathbb{A}| = 2$ and $k = 8$. The setting is illustrated in Fig. 6. We let the sampling distribution ν to be the uniform distribution over $\mathbb{S}$. The feature matrix Φ and the transition matrix P^π are given by

Fig. 4 A 3-link actuated pendulum setting. Each rotational joint i, $1 \leq i \leq 3$ is actuated by a torque a_i. The system is parametrized by the angle ψ_i against the vertical direction 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 as close to 0 as possible. The 5-link actuated pendulum setting that we actually consider in the experiments is similar to this but with two additional links



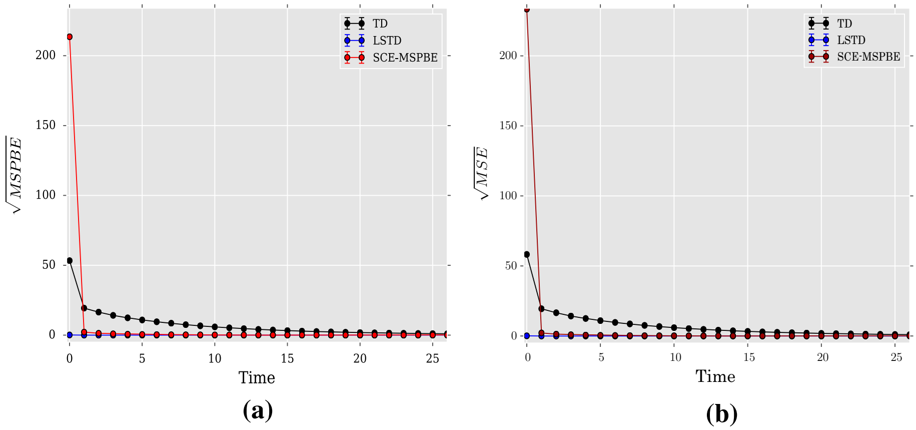


Fig. 5 5-link actuated pendulum setting. The respective trajectories of the $\sqrt{\text{MSPBE}}$ and $\sqrt{\text{MSE}}$ generated by TD(0), LSTD(0) and SCE-MSPBEM algorithms are plotted. The graph on the left is for $\sqrt{\text{MSPBE}}$, while on the right is that of $\sqrt{\text{MSE}}$. Note that $\sqrt{\text{MSE}}$ also converges to 0 since the feature set is perfect. **a** $\sqrt{\text{MSPBE}(\mu_t)}$. **b** $\sqrt{\text{MSE}(\mu_t)}$

$$\Phi = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{P}^{\pi} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{97}$$

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(0) with SCE-MSPBEM is shown in Fig. 7. Here, the

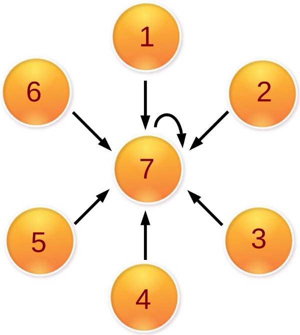


Fig. 6 Baird's 7-star MDP

performance metric used for comparison is the $\sqrt{\text{MSE}(\cdot)}$ of the prediction vector generated by the corresponding algorithm at time t.

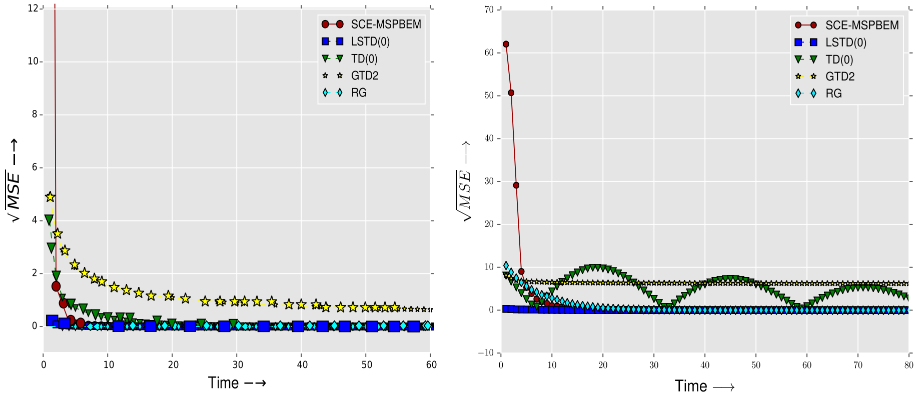
The algorithm parameter values used in the experiment are provided in Table 5 of Appendix.

A careful analysis in Schoknecht and Merke (2002) has shown that when the discount factor $\gamma \leq 0.88$, with appropriate learning rate, 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 learning rate. This is explicitly demonstrated in Fig. 7. However our algorithm SCE-MSPBEM converges in both cases, which demonstrates the stable behaviour exhibited by our algorithm.

The algorithms were also compared on the same Baird's 7-star, but with a different feature matrix $\varPhi_1$ as under.

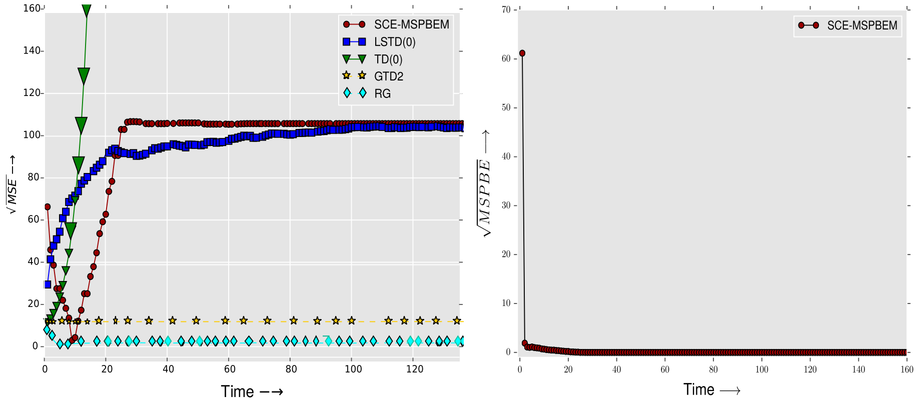
$$\varPhi_1 = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

In this case, the reward function is given by $\mathrm{R}(s, s') = 2.0, \forall s, s' \in \mathbb{S}$. Note that $\varPhi_1$ gives an imperfect feature set. The algorithm parameter values used are same as earlier. The results are shown in Fig. 8. In this case also, TD(0) diverges. However, SCE-MSPBEM is seen to exhibit good stable behaviour.



(a) **(b)**

Fig. 7 Baird's 7-star MDP with perfect feature set. For $\gamma = 0.1$, all the algorithms converge with SCE-MSPBEM converging faster compared to a few algorithms, while being on par with the remaining algorithms. Note that this performance is obtained despite the fact that the initial value of SCE-MSPBEM is far from the solution compared to the rest of the algorithms. For $\gamma = 0.9$, TD(0) does not converge (which is in compliance with the observations made in Schoknecht and Merke 2002), while the performance of GTD2 is slow. However, SCE-MSPBEM exhibits good convergence behaviour which demonstrates the stable nature of the algorithm. **a** Discount factor $\gamma = 0.1$. **b** Discount factor $\gamma = 0.9$



(a) **(b)**

Fig. 8 Baird's 7-Star MDP with imperfect feature set. Here the discount factor $\gamma = 0.99$. In this case, TD(0) diverges. However, $\sqrt{\text{MSE}}$ of SCE-MSPBEM and LSTD(0) converge to the same limit point (=103.0) with SCE-MSPBEM converging faster than LSTD. Also note that the RG method converges to a different limit (= 1.6919). This is because the feature set is imperfect and also the fact that RG minimizes MSBR, while SCE-MSPBEM and LSTD minimize MSPBE. To verify this fact, note that in **b**, $\sqrt{\text{MSPBE}(\mu_t)}$ of SCE-MSPBEM converges to 0 which is indeed the minimum of MSPBE. **a** $\sqrt{\text{MSE}}$. **b** $\sqrt{\text{MSPBE}}$

8.4 Experiment 4: 10-state ring MDP (Kveton et al. 2006)

Next, we studied the performance comparisons of the algorithms on a 10-ring MDP with $|\mathbb{S}| = 10$ and $k = 8$. The setting is illustrated in Fig. 9. We let the sampling distribution ν to be the uniform distribution over $\mathbb{S}$. The transition matrix P^{π} and the feature matrix Φ are given by

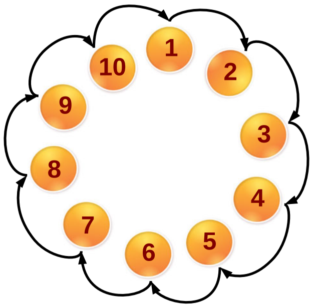


Fig. 9 10-Ring MDP

$$\mathrm{P}^{\pi} = \begin{pmatrix} 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0&0&0 \end{pmatrix}, \quad \Phi = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&1&0&0 \end{pmatrix}. \tag{98}$$

The reward function is $\mathrm{R}(s, s') = 1.0, \forall s, s' \in \mathbb{S}$.

The performance comparisons of the algorithms GTD2, TD(0) and LSTD(0) with SCE-MSPBEM are shown in Fig. 10. The performance metric used here is the $\sqrt{\mathrm{MSE}(\cdot)}$ of the prediction vector generated by the corresponding algorithm at time t. The Markov chain in this case is ergodic and the uniform distribution over $\mathbb{S}$ is indeed the stationary distribution of the Markov chain. So theoretically all the algorithms should converge and the results in Fig. 10 confirm this. However, there is a significant difference in the rate of convergence of the various algorithms for large values of the discount factor γ. For $\gamma = 0.99$, the results show that GTD2 and RG trail behind other methods, while our method is only behind LSTD and outperforms TD(0), RG and GTD2. The algorithm parameter values used in the experiment are provided in Table 6 of "Appendix".

8.5 Experiment 5: random MDP with radial basis functions and fourier basis

These toy experiments are designed by us. Here, the tests are performed using standard basis functions to demonstrate that the algorithm is not dependent on any particular feature set. Two types of feature sets are considered here: Fourier basis functions and radial basis functions (RBF).

The Fourier basis functions Konidaris et al. (2011) are defined as follows:

$$\phi_i(s) = \begin{cases} 1 & \text{if } i = 1, \\ \cos \frac{(i+1)\pi s}{2} & \text{if } i \text{ is odd}, \\ \sin \frac{i\pi s}{2} & \text{if } i \text{ is even}. \end{cases} \tag{99}$$

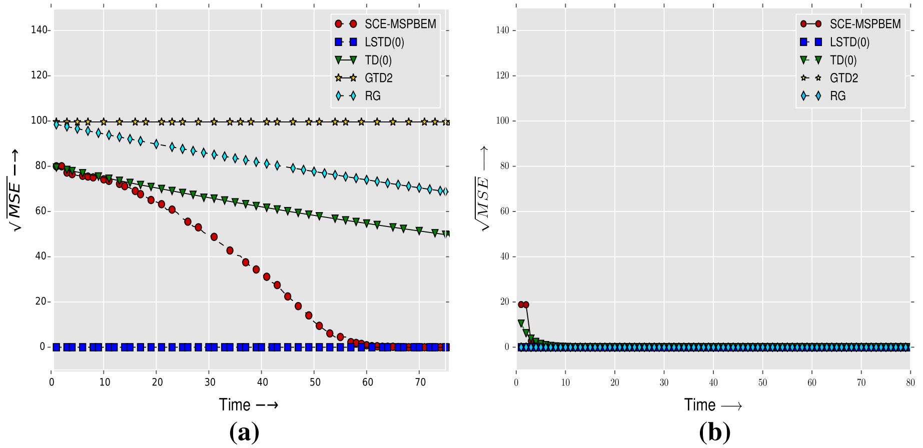


Fig. 10 10-Ring MDP with perfect feature set: For $\gamma = 0.1$, all the algorithms exhibit almost the same rate of convergence. For $\gamma = 0.99$, SCE-MSPBEM converges faster than TD(0), GTD2 and RG. **a** Discount factor $\gamma = 0.99$. **b** Discount factor $\gamma = 0.1$

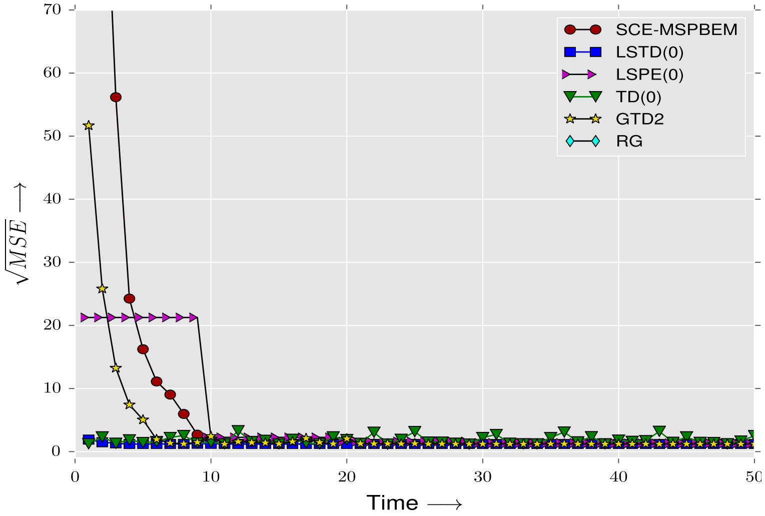


Fig. 11 Fourier basis function: here, $|\mathbb{S}| = 1000$, $|\mathbb{A}| = 200$, $k = 50$ and $\gamma = 0.9$. In this case, SCE-MSPBEM shows good convergence behaviour

The radial basis functions are defined as follows:

$$\phi_i(s) = e^{-\frac{(s-m_i)^2}{2.0v_i^2}} \tag{100}$$

with m_i and v_i fixed *a priori*.

In both the cases, the reward function is given by

$$\mathrm{R}(s, s') = G(s)G(s')\left(\frac{1}{(1.0 + s')^{0.25}}\right), \ \forall s, s' \in \mathbb{S}, \tag{101}$$

where the vector $G \in (0, 1)^{|\mathbb{S}|}$ is initialized for the algorithm with $G(s) \sim U(0, 1)$, $\forall s \in \mathbb{S}$.

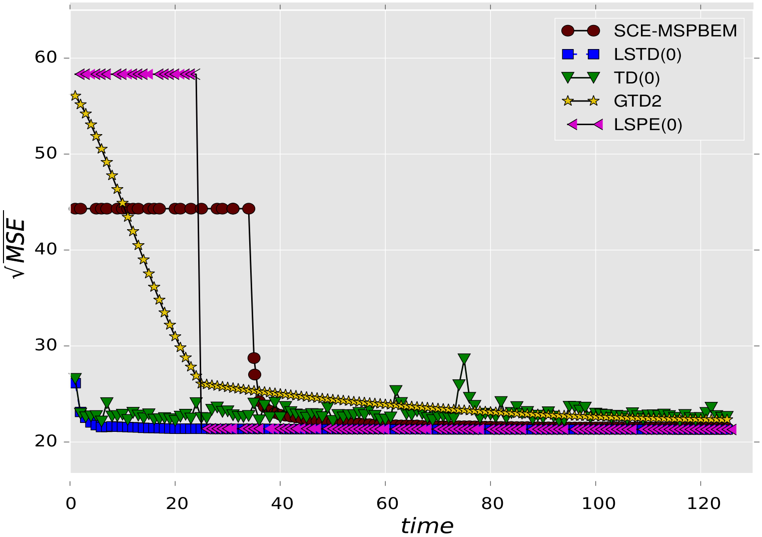


Fig. 12 Radial basis function. Here, $|\mathbb{S}| = 1000$, $|\mathbb{A}| = 200$, $k = 50$ and $\gamma = 0.01$. In this case, SCE-MSPBEM converges to the same limit point as other algorithms

Also in both the cases, the transition probability matrix P^{π} is generated as follows:

$$P^{\pi}(s, s')\binom{|\mathbb{S}|}{s'}b(s)^{s'}(1.0 - b(s))^{|\mathbb{S}|-s'}, \ \forall s, s' \in \mathbb{S}, \tag{102}$$

where the vector $b \in (0, 1)^{|\mathbb{S}|}$ is initialized for the algorithm with $b(s) \sim U(0, 1), \forall s \in \mathbb{S}$. It is easy to verify that the Markov chain defined by P_{π} is ergodic in nature.

In the case of RBF, we let $|\mathbb{S}| = 1000$, $|\mathbb{A}| = 200$, $k = 50$, $m_i = 10 + 20(i - 1)$ and $v_i = 10$, while for Fourier basis functions, we let $|\mathbb{S}| = 1000$, $|\mathbb{A}| = 200$, $k = 50$. In both the cases, the distribution ν is the stationary distribution of the Markov chain. The simulation is run sufficiently long to ensure that the chain achieves its steady state behaviour, *i.e.*, the states appear with the stationary distribution. The algorithm parameter values used in the experiment are provided in Table 7 of "Appendix" and the results obtained are provided in Figs. 11 and 12.

Also note that when Fourier basis is used, the discount factor $\gamma = 0.9$ and for RBFs, $\gamma = 0.01$. SCE-MSPBEM exhibits good convergence behaviour in both cases, which shows the non-dependence of SCE-MSPBEM on the discount factor γ. This is important because in Schoknecht and Merke (2003), the performance of TD methods is shown to be dependent on the discount factor γ.

To measure how well our algorithm scales with respect to the size of the state space, we applied it on a medium sized MDP,[10] where $|\mathbb{S}| = 2^{15}$, $|\mathbb{A}| = 50$, $k = 100$ and $\gamma = 0.9$. This is the stress test. The reward function R and the transition probability matrix P_{π} are generated using Eqs. (101) and (102) respectively. RBFs are used as the features in this case. Since the MDP is huge, the algorithms were run on Amazon cloud servers. The true value function V^{π} was computed and the $\sqrt{\text{MSE}}$s of the prediction vectors generated by the different algorithms were compared. The performance results are shown in Table 2. The

[10] This is the biggest MDP we could deploy on a machine with 3.2GHz processor and 8GB of memory.

Table 2 Performance comparison of various algorithms on a medium sized MDP. Here $|\mathbb{S}| = 2^{15}$, $|\mathbb{A}| = 50$, $k = 100$, and $\gamma = 0.9$. RBF is used as the feature set. The feature set is imperfect. The entries in the table correspond to the $\sqrt{\text{MSE}}$ values obtained from the respective algorithms on 7 different random MDPs. While the entries of SCE-MSPBEM, and LSTD(0) appear to be similar, they actually differed in decimal digits that are not shown here for lack of space

Ex#	SCE-MSPBEM	LSTD(0)	TD(0)	LSPE(0)	GTD2
1	23.339	23.339	24.581	23.354	24.932
2	23.142	23.142	24.372	23.178	24.755
3	23.332	23.332	24.537	23.446	24.881
4	22.978	22.978	24.194	22.987	24.532
5	22.950	22.950	24.203	22.965	24.554
6	23.060	23.060	24.253	23.084	24.607
7	23.228	23.228	24.481	23.244	24.835

results show that the performance of our algorithm does not seem affected by the complexity of the MDP.

8.6 Experiment 6: non-linear function approximation of value function (Tsitsiklis and Roy 1997)

To demonstrate the flexibility and robustness of our approach, we also consider a few non-linear function approximation RL settings. The landscape in the non-linear setting is mostly non-convex and therefore multiple local optima exist. The stable non-linear function approximation extension of GTD2 is only shown to converge to the local optima (Maei et al. 2009). We believe that the non-linear setting offers the perfect scaffolding to demonstrate the global convergence property of our approach.

8.6.1 Experiment 6.1: Van Roy and Tsitsiklis MDP (Tsitsiklis and Roy 1997)

This particular setting is designed in Tsitsiklis and Roy (1997) to show the divergence of the standard TD(0) algorithm in reinforcement learning under a non-linear approximation architecture.

We consider here a discrete time Markov chain with state space $\mathbb{S} = \{1, 2, 3\}$, discount factor $\gamma = 0.9$, the reward function $\mathrm{R}(s, s') = 0, \forall s, s' \in \mathbb{S}$ and the transition probability matrix as under:

$$\mathrm{P} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix}.$$

Note that the Markov chain is ergodic and hence the sample trajectory is obtained by following the dynamics of the Markov chain. Therefore the steady-state distribution of the Markov chain is indeed the sampling distribution. Here, we minimize the MSBR error function and it demands double sampling as prescribed in Assumption $A3'$. The optimization is as follows:

$$\eta^* \in \operatorname*{arg\,min}_{\eta \in \mathcal{Q} \subseteq \mathbb{R}} \mathbb{E}\left[\mathbb{E}^2\left[\delta_t\right] |\mathbf{s}_t\right], \tag{103}$$

where $\delta_t = \mathbf{r}_t + \gamma \psi_\eta(\mathbf{s}'_t) - \psi_\eta(\mathbf{s}_t)$. We also have

$$\psi_\eta(s) = (a(s)\cos(\tau\eta) - b(s)\sin(\tau\eta))e^{\epsilon\eta}, \tag{104}$$

where $a = [100, -70, -30]^\top$, $b = [23.094, -98.15, 75.056]^\top$, $\tau = 0.01$ and $\epsilon = 0.001$. Here ψ_η defines the projected non-linear manifold. The true value function of this particular setting is $V = (0, 0, 0)^\top$.

Now the challenge here is to best approximate the true value function V using the family of non-linear functions ψ_η parametrized by $\eta \in \mathbb{R}$ by solving the optimization problem (103). It is easy to see that $\psi_{-\infty} = V$ and hence is a degenerate setting.

The objective function in Eq. (103) can be rearranged as

$$\begin{aligned}&[\upsilon_{11}, \upsilon_{21}, \omega_{31}]\left[1.0, 2.0e^{\epsilon\eta}\cos(\tau\eta), -2.0e^{\epsilon\eta}\sin(\tau\eta)\right]^\top \\ &\quad + \left[e^{\epsilon\eta}\cos(\tau\eta), -e^{\epsilon\eta}\sin(\tau\eta)\right]\begin{bmatrix}\upsilon_{22} & \upsilon_{23}\\ \upsilon_{32} & \upsilon_{33}\end{bmatrix}\left[e^{\epsilon\eta}\cos(\tau\eta), -e^{\epsilon\eta}\sin(\tau\eta)\right]^\top,\end{aligned}$$

where $\upsilon \in \mathbb{R}^{3\times 3}$ with $\upsilon = (\upsilon_{ij})_{1\le i,j\le 3} \triangleq \mathbb{E}[h_t]\mathbb{E}[h'_t]^\top$. Here $h_t = [\mathbf{r}_t, a(\mathbf{s}'_t) - a(\mathbf{s}_t), b(\mathbf{s}'_t) - b(\mathbf{s}_t)]^\top$ and $h'_t = [\mathbf{r}'_t, a(\mathbf{s}''_t) - a(\mathbf{s}_t), b(\mathbf{s}''_t) - b(\mathbf{s}_t)]^\top$.

Now we maintain the time indexed random vector $\upsilon^{(t)} \in \mathbb{R}^{3\times 3}$ with $\upsilon^{(t)} = (\upsilon^{(t)}_{ij})_{1\le i,j\le 3}$ and employ the following recursion to track υ:

$$\upsilon^{(t+1)} = \upsilon^{(t)} + \alpha_{t+1}(h_t {h'_t}^\top - \upsilon^{(t)}). \tag{105}$$

Also, we define

$$\begin{aligned}\bar{\mathcal{J}}_b(\upsilon^{(t)}, \eta) = &\left[\upsilon^{(t)}_{11}, \upsilon^{(t)}_{21}, \upsilon^{(t)}_{31}\right]\left[1.0, 2.0e^{\epsilon\eta}\cos(\tau\eta), -2.0e^{\epsilon\eta}\sin(\tau\eta)\right]^\top \\ &+ \left[e^{\epsilon\eta}\cos(\tau\eta), -e^{\epsilon\eta}\sin(\tau\eta)\right]\begin{bmatrix}\upsilon^{(t)}_{22} & \upsilon^{(t)}_{23}\\ \upsilon^{(t)}_{32} & \upsilon^{(t)}_{33}\end{bmatrix}\left[e^{\epsilon\eta}\cos(\tau\eta), -e^{\epsilon\eta}\sin(\tau\eta)\right]^\top.\end{aligned} \tag{106}$$

Now we solve the optimization problem (103) using Algorithm 2 with the objective function defined in Eq. (106) (*i.e.*, using Eq. (105) instead of Eq. (48) and Eq. (106) instead of Eq. (49) respectively).

The various parameter values used in the experiment are provided in Table 8 of Appendix. The results of the experiment are shown in Fig. 13. The x-axis is the iteration number t. The performance measure considered here is the mean squared error (MSE) which is defined in Eq. (5). Algorithm 2 is seen to clearly outperform TD(0) and GTD2 here.

8.6.2 Experiment 6.2: Baird's 7-star MDP using non-linear function approximation

Here, we consider the Baird's 7-star MDP defined in Sect. 8.3 with discount factor $\gamma = 0.9$, $k = 8$ and the sampling distribution to be the uniform distribution over $\mathbb{S}$. To perform the non-linear function approximation, we consider the non-linear manifold given by $\{\Phi h(z) | z \in \mathbb{R}^8\}$, where $h(z) \triangleq (\cos^2(z_1)\exp(0.01z_1), \cos^2(z_2)\exp(0.01z_2), \ldots, \cos^2(z_8)\exp(0.01z_8))^\top$ and Φ is defined in Eq. (97). The reward function is given by $\mathrm{R}(s, s') = 0, \forall s, s' \in \mathbb{S}$ and hence the true value function is $(0, 0, \ldots, 0)^\top_{7\times 1}$. Due to the unique nature of the non-linear manifold, one can directly apply SCE-MSBRM (Algorithm 2) with $h(z)$ replacing z in Eq. (49). This setting presents a hard and challenging task for TD(λ) since we already experienced the erratic and unstable behaviour of TD(λ) in the linear function approximation version of Baird's 7-star. This setting also proves to be a litmus test for determining

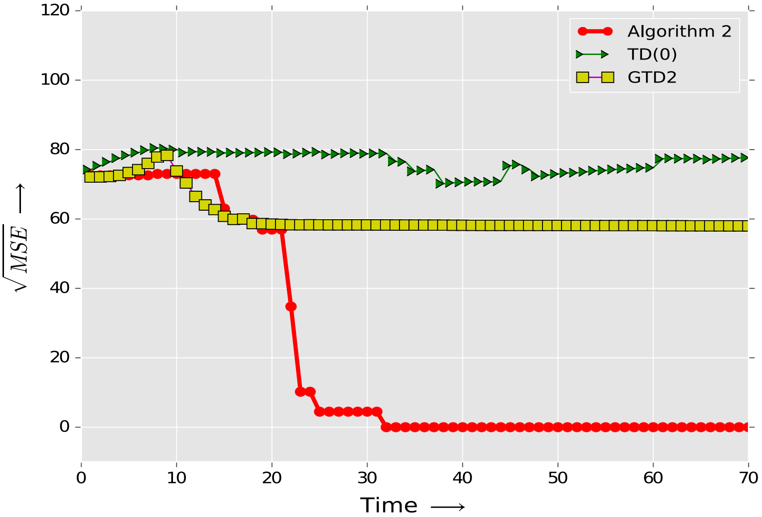


Fig. 13 Non-linear function approximation on Van Roy and Tsitsiklis MDP. The plot shows the trajectory of $\sqrt{\text{MSE}}$ generated by TD(0), GTD2 and our algorithm against the iteration number t. Algorithm 2 (SCE-MSBRM) predicts the true value function V which follows from the observation that $\sqrt{\text{MSE}}$ converges to 0. TD(0) slowly diverges, while GTD2 converges to a sub-optimal solution. This experiment further demonstrates the effectiveness of our proposed scheme to non-convex settings

the confines of the stable characteristic of the non-linear function approximation version of GTD2. The results obtained are provided in Fig. 14. The various parameter values used in the experiment are provided in Table 9 of Appendix. It can be seen that whereas both TD(0) and GTD2 diverge here, SCE-MSBRM is converging to the true value.

8.6.3 Experiment 6.3: 10*-ring MDP using non-linear function approximation*

Here, we consider the 10-ring MDP defined in Sect. 8.4 with discount factor $\gamma = 0.99$, $k = 8$ and the sampling distribution as the stationary distribution of the underlying Markov chain. Here, we consider the non-linear manifold given by $\{\Phi h(z) | z \in \mathbb{R}^8\}$, where $h(z) \triangleq (\cos^2(z_1)\exp(0.1z_1), \cos^2(z_2)\exp(0.1z_2), \ldots, \cos^2(z_8)\exp(0.1z_8))^\top$ and Φ is defined in Eq. (98). The reward function is given by $\mathrm{R}(s, s') = 0, \forall s, s' \in \mathbb{S}$ and hence the true value function is $(0, 0, \ldots, 0)^\top_{10\times 1}$. Similar to the previous experiment, here also one can directly apply SCE-MSBRM (Algorithm 2) with $h(z)$ replacing z in Eq. (49). The results obtained are provided in Fig. 15. The various parameter values we used are provided in Table 10 of "Appendix". GTD2 does not converge to the true value here while both SCE-MSBRM and TD(0) do, with TD(0) marginally better.

9 Conclusion

We proposed, for the first time, an application of the cross entropy (CE) method to the prediction problem in reinforcement learning (RL) under the linear function approximation architecture. This task is accomplished by employing the multi-timescale stochastic approximation variant of the cross entropy optimization method to minimize the mean squared projected Bellman error (MSPBE) and mean squared Bellman error (MSBR) objectives. The

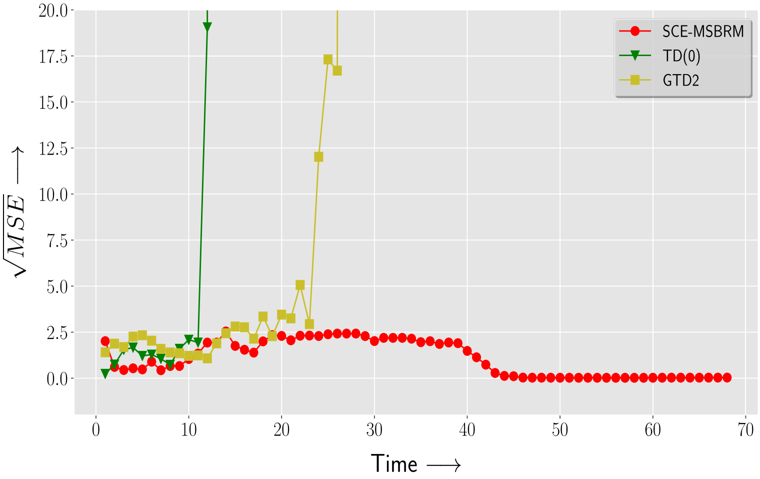


Fig. 14 Non-linear function approximation on Baird's 7-star MDP: The plot shows the trajectory of $\sqrt{\text{MSE}}$ generated by TD(0), GTD2 and SCE-MSBRM against $t/100$, where t is the iteration number. SCE-MSBRM predicts the true value function V which follows from the observation that $\sqrt{\text{MSE}}$ converges to 0. Note that both TD(0) and GTD2 diverge. The divergence of TD(0) is expected since it also diverges in the linear case (see Fig. 7). However, the divergence of the stable non-linear GTD2 is not unforeseen, but subjective. The rationale behind this erratic behaviour is ascribed to the absence of the projection (Π_C defined in Algorithm 9) of the iterates in the experiment conducted. The projection operator Π_C is necessary to ensure the stability of the algorithm, however its computation is hard and hence the omission

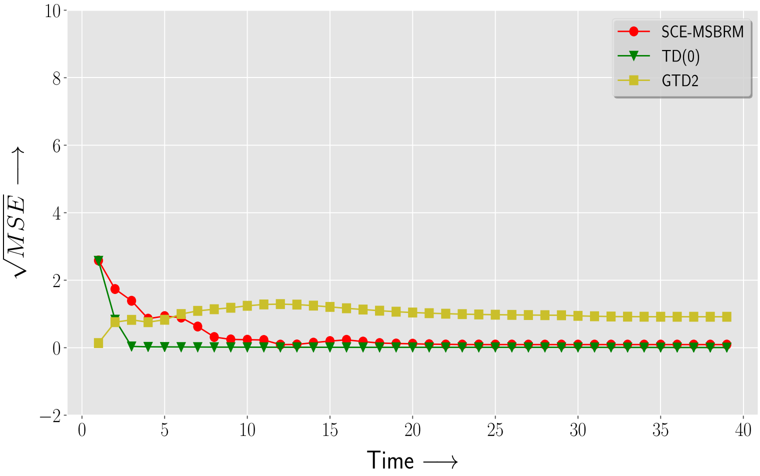


Fig. 15 Non-linear function approximation on 10-ring MDP: The plot shows the trajectory of $\sqrt{\text{MSE}}$ generated by TD(0), GTD2 and SCE-MSBRM against $t/150$, where t is the iteration number. SCE-MSBRM predicts the true value function V which follows from the observation that $\sqrt{\text{MSE}}$ converges to 0. Note that TD(0) also converges to the true value function while GTD2 could only find sub-optimal solution here

proofs of convergence of the algorithms to the optimum values using the ODE method are also provided. The theoretical analysis is supplemented by extensive experimental evaluation which is shown to corroborate the claims. Experimental comparisons with the state-of-the-art algorithms show the superiority in terms of stability and accuracy while being competitive enough with regard to computational efficiency and rate of convergence. As future work, one may design similar cross entropy approaches for both prediction and control problems. More numerical experiments involving other non-linear function approximators, delayed rewards *etc.* may be tried as well.

Appendices

A Linear function approximation (LFA) based prediction algorithms

```
Algorithm 3: TD(λ) LFA
$\delta_t = \mathbf{r}_t + \gamma \mathbf{z}_t^\top \phi(\mathbf{s}'_t) - \mathbf{z}_t^\top \phi(\mathbf{s}_t)$;
$\mathbf{e}_{t+1} = \phi(\mathbf{s}_t) + \gamma\lambda\mathbf{e}_t$;
$\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_{t+1}\delta_t\mathbf{e}_{t+1}$.
$\alpha_t > 0$ satisfies $\sum_{t=1}^{\infty}\alpha_t = \infty$, $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$.
```

```
Algorithm 4: RG LFA
$\delta_t = \mathbf{r}_t + \gamma \mathbf{z}_t^\top \phi(\mathbf{s}'_t) - \mathbf{z}_t^\top \phi(\mathbf{s}_t)$;
$\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_{t+1}\delta_t \left(\phi(\mathbf{s}_t) - \gamma\phi(\mathbf{s}'_{t+1})\right)$;
$\alpha_t > 0$ satisfies $\sum_{t=1}^{\infty}\alpha_t = \infty$, $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$.
```

```
Algorithm 5: GTD2 LFA
    $\delta_t = \mathbf{r}_t + \gamma \mathbf{z}_t^\top \phi(\mathbf{s}'_t) - \mathbf{z}_t^\top \phi(\mathbf{s}_t)$;
    $\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_{t+1}\left(\phi(\mathbf{s}_t) - \gamma\phi(\mathbf{s}'_t)\right)\left(\phi(\mathbf{s}_t)^\top\mathbf{v}_t\right)$;
    $\mathbf{v}_{t+1} = \mathbf{v}_t + \beta_{t+1}(\delta_t - \phi(\mathbf{s}_t)^\top\mathbf{v}_t)\phi(\mathbf{s}_t)$;
$\alpha_t, \beta_t > 0$ satisfy $\sum_{t=1}^{\infty}\alpha_t = \infty$, $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$ and $\beta_t = \eta\alpha_t$, where $\eta > 0$.
```

```
Algorithm 6: TDC LFA
    $\delta_t = \mathbf{r}_t + \gamma \mathbf{z}_t^\top \phi(\mathbf{s}'_t) - \mathbf{z}_t^\top \phi(\mathbf{s}_t)$;
    $\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_{t+1}\delta_t\phi(\mathbf{s}_t) - \gamma\phi(\mathbf{s}'_t)(\phi(\mathbf{s}_t)^\top\mathbf{v}_t)$;
    $\mathbf{v}_{t+1} = \mathbf{v}_t + \beta_{t+1}(\delta_t - \phi(\mathbf{s}_t)^\top\mathbf{v}_t)\phi(\mathbf{s}_t)$;
$\alpha_t, \beta_t > 0$ satisfy $\sum_{t=1}^{\infty}\alpha_t = \sum_{t=1}^{\infty}\beta_t = \infty$, $\sum_{t=1}^{\infty}\left(\alpha_t^2 + \beta_t^2\right) < \infty$ and $\frac{\alpha_t}{\beta_t} \to 0$.
```

```
Algorithm 7: LSTD(λ) LFA
    $\mathbf{A}_0 = \epsilon\mathbb{I}_{k_1 \times k_1}, \epsilon > 0, t = 0, \mathbf{b}_0 = \mathbf{e}_0 = 0_{k \times 1}$;
    while stopping criteria not satisfied do
        $\mathbf{e}_{t+1} = \phi(\mathbf{s}_t) + \gamma\lambda\mathbf{e}_t$;
        $\mathbf{A}_{t+1} = \mathbf{A}_t + \mathbf{e}_{t+1}\left(\phi(\mathbf{s}_t) - \gamma\phi(\mathbf{s}'_t)\right)^\top$;
        $\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{e}_{t+1}\mathbf{r}_t$;
    return $\mathbf{A}_T^{-1}\mathbf{b}_T$;
```

```
Algorithm 8: LSPE(λ) LFA
  $\mathbf{A}_0 = \mathbf{B}_0 = \epsilon\mathbb{I}_{k\times k}, \epsilon > 0, t = 0, \mathbf{b}_0 = \mathbf{e}_0 = 0_{k\times 1}$;
  while stopping criteria not satisfied do
      $\mathbf{B}_{t+1} = \mathbf{B}_t + \phi(\mathbf{s}_t)\phi(\mathbf{s}_t)^\top$;
      $\mathbf{e}_{t+1} = \phi(\mathbf{s}_t) + \gamma\lambda\mathbf{e}_t$;
      $\mathbf{A}_{t+1} = \mathbf{A}_t + \mathbf{e}_{t+1}\left(\phi(\mathbf{s}_t) - \gamma\phi(\mathbf{s}'_t)\right)^\top$;
      $\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{e}_{t+1}\mathbf{r}_t$;
  return $\mathbf{A}_T^{-1}\mathbf{B}_T\mathbf{b}_T$;
```

B Non-linear function approximation (NLFA) based prediction algorithms

```
Algorithm 9: GTD2 NLFA
  $\delta_t = \mathbf{r}_t + \gamma V_{\theta_t}(\mathbf{s}'_t) - V_{\theta_t}(\mathbf{s}_t)$;
  $\mathbf{w}_{t+1} = \mathbf{w}_t + \beta_{t+1}\left(\delta_t - \nabla V_{\theta_t}(\mathbf{s}_t)^\top\mathbf{w}_t\right)\nabla V_{\theta_t}(\mathbf{s}_t)$;
  $h_t = (\delta_t - \nabla V_{\theta_t}(\mathbf{s}_t)^\top\mathbf{w}_t)\nabla^2 V_{\theta_t}(\mathbf{s}_t)\mathbf{w}_t$;
  $\theta_{t+1} = \Pi_C\Big(\theta_t + \alpha_{t+1}\{\delta_t\nabla V_{\theta_t}(\mathbf{s}_t) -$
          $\gamma\nabla V_{\theta_t}(\mathbf{s}'_t)(\nabla V_{\theta_t}(\mathbf{s}_t)^\top\mathbf{w}_t) - h_{t+1}\}\Big)$;
```

$\alpha_t, \beta_t > 0$ satisfy $\sum_{t=1}^{\infty}\alpha_t = \infty$, $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$ and $\frac{\alpha_t}{\beta_t} \to 0$.
Here $\{V_\theta \in \mathbb{R}^{|\mathbb{S}|} | \theta \in \mathbb{R}^n\}$ is the differentiable sub-manifold of $\mathbb{R}^{|\mathbb{S}|}$.
C is a predetermined compact subset of $\mathbb{R}^n$ and Π_C is the projection operator on C w.r.t. some appropriate norm.

C Parameter values used in various experiments

See Tables 3, 4, 5, 6, 7, 8, 9 and 10

Table 3 The experiment parameter values and the algorithm parameter values used in the cart-pole experiment (Experiment 1)

Gravitational acceleration (g)	$9.8\,\frac{\mathrm{m}}{\mathrm{s}^2}$
Mass of the pole (m)	0.5 kg
Mass of the cart (M)	0.5 kg
Length of the pole (l)	0.6 m
Friction coefficient (b)	$0.1\,\mathrm{N(ms)}^{-1}$
Integration time step (Δt)	0.1 s
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 4 The experiment parameter values and the algorithm parameter values used in the 5-link actuated pendulum experiment (Experiment 2)

Gravitational acceleration (g)	$9.8\,\frac{\mathrm{m}}{\mathrm{s}^2}$
Mass of the pole (m)	1.0 kg
Length of the pole (l)	1.0 m
Integration time step (Δt)	0.1 s
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

Table 5 Algorithm parameter values used in the Baird's 7-star experiment (Experiment 3)

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

Table 6 Algorithm parameter values used in the 10-state ring experiment (Experiment 4)

α_t	0.001
β_t	0.05
c_t	0.075
λ	0.001
ϵ_1	0.85
ρ	0.1

Table 7 Algorithm parameter values used in the random MDP experiment (Experiment 5)

Both RBF & Fourier Basis	
α_t	0.001
β_t	0.05
c_t	0.075
λ	0.001
ϵ_1	0.85
ρ	0.1

Table 8 Algorithm parameter values used in the Van Roy and Tsitsiklis non-linear function approximation experiment (Experiment 6.1)

$S(\cdot)$	α_t	β_t	λ	c_t	ϵ_1	ρ
$\exp(rx), r = 10^{-6}$	$\frac{1}{t}$	0.9	0.01	0.03	0.95	0.1

Table 9 Algorithm parameter values used in the Baird's 7-star non-linear function approximation experiment (Experiment 6.2)

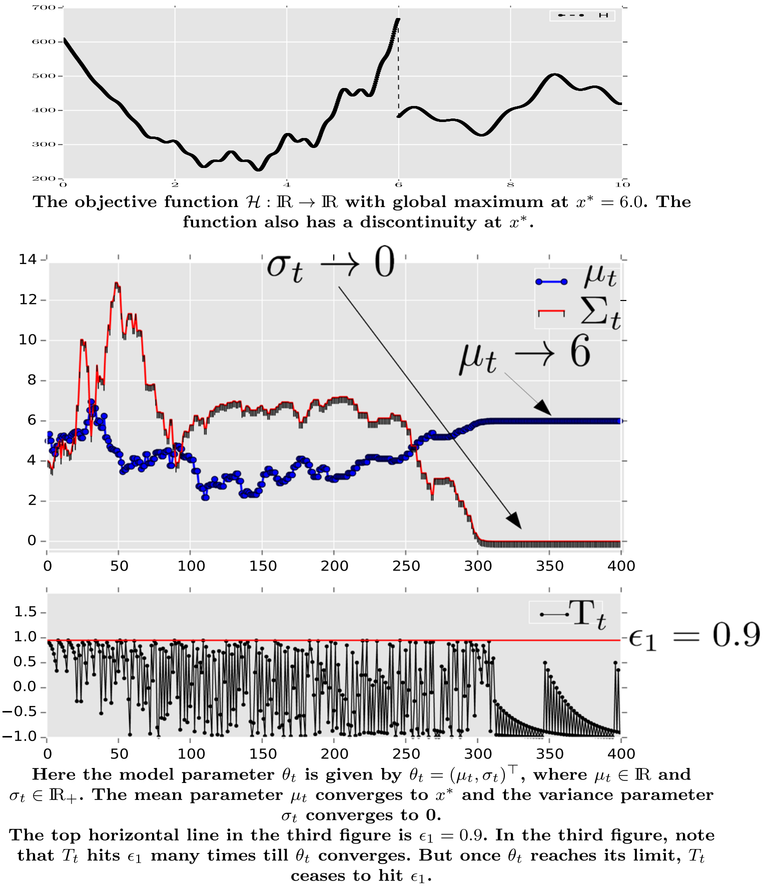
$S(\cdot)$	α_t	β_t	λ	c_t	ϵ_1	ρ
$\exp(rx), r = 0.2$	0.02	0.1	0.001	0.05	0.8	0.1

Table 10 Algorithm parameter values used in the 10-ring MDP non-linear function approximation experiment (Experiment 6.3)

$S(\cdot)$	α_t	β_t	λ	c_t	ϵ_1	ρ
$\exp(rx), r = 0.05$	0.04	0.2	0.001	0.08	0.8	0.1

D Illustration of CE optimization procedure

See Fig. 16



The objective function $\mathcal{H} : \mathbb{R} \to \mathbb{R}$ with global maximum at $x^* = 6.0$. The function also has a discontinuity at x^*.

Here the model parameter θ_t is given by $\theta_t = (\mu_t, \sigma_t)^\top$, where $\mu_t \in \mathbb{R}$ and $\sigma_t \in \mathbb{R}_+$. The mean parameter μ_t converges to x^* and the variance parameter σ_t converges to 0.

The top horizontal line in the third figure is $\epsilon_1 = 0.9$. In the third figure, note that T_t hits ϵ_1 many times till θ_t converges. But once θ_t reaches its limit, T_t ceases to hit ϵ_1.

Fig. 16 Illustration of the CE method on a deterministic optimization problem

E Borkar–Meyn theorem (Theorem 2.1 of Borkar and Meyn 2000)

Theorem 3 *For the stochastic recursion of* $x_n \in \mathbb{R}^d$ *given by*

$$x_{n+1} = x_n + a_n \left(h(x_n) + \mathbb{M}_{n+1}\right), \quad n \in \mathbb{N}, \tag{107}$$

if the following assumptions are satisfied:

- *The map* $h : \mathbb{R}^d \to \mathbb{R}^d$ *is Lipschitz, i.e.,* $\|h(x) - h(y)\| \leq L\|x - y\|$, *for some* $0 < L < \infty$.
- *Step-sizes* $\{a_n\}$ *are positive scalars satisfying*

  $$\sum_n a_n = \infty, \quad \sum_n a_n^2 < \infty.$$

- $\{\mathbb{M}_{n+1}\}_{n\in\mathbb{N}}$ *is a martingale difference noise w.r.t. the increasing family of* σ*-fields*

  $$\mathcal{F}_n \triangleq \sigma(x_m, \mathbb{M}_m, m \leq n), \quad n \in \mathbb{N}.$$

  That is,

  $$\mathbb{E}\left[\mathbb{M}_{n+1} | \mathcal{F}_n\right] = 0 \ \ a.s., \quad n \in \mathbb{N}.$$

  Furthermore, $\{\mathbb{M}_{n+1}\}_{n\in\mathbb{N}}$ *are square-integrable with*

  $$\mathbb{E}\left[\|\mathbb{M}_{n+1}\|^2 | \mathcal{F}_n\right] \leq K(1 + \|x_n\|^2) \ \ a.s., \quad n \in \mathbb{N},$$

  for some constant $K > 0$.
- *The functions* $h_c(x) \triangleq \frac{h(cx)}{x}$, $c \geq 1$, $x \in \mathbb{R}^d$, *satisfy* $h_c(x) \to h_\infty(x)$ *as* $c \to \infty$, *uniformly on compacts for some* $h_\infty \in C(\mathbb{R}^d)$. *Furthermore, the ODE*

  $$\dot{x}(t) = h_\infty(x(t)) \tag{108}$$

  has the origin as its unique globally asymptotically stable equilibrium,

then

$$\sup_{n\in\mathbb{N}} \|x_n\| < \infty \ \ a.s.$$

References

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the twelfth international conference on machine learning* (pp. 30–37).

Benveniste, A., Métivier, M., & Priouret, P. (2012). *Adaptive Algorithms and Stochastic Approximations* (Vol. 22). Berlin: Springer.

Bertsekas, D. P. (2013). *Dynamic programming and optimal control* (Vol. 2). Belmont: Athena Scientific.

Borkar, V. S. (1997). Stochastic approximation with two time scales. *Systems & Control Letters*, *29*(5), 291–294.

Borkar, V. S. (2008). *Stochastic approximation: A dynamical systems viewpoint*. Cambridge: Cambridge University Press.

Borkar, V. S. (2012). *Probability theory: An advanced course*. Berlin: Springer.

Borkar, V. S., & Meyn, S. P. (2000). The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, *38*(2), 447–469.

Boyan, J. A. (2002). Technical update: Least-squares temporal difference learning. *Machine Learning*, *49*(2–3), 233–246.

Bradtke, S. J., & Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, *22*(1–3), 33–57.

Busoniu, L., Ernst, D., De Schutter, B., & Babuska, R. (2009). Policy search with cross-entropy optimization of basis functions. In *IEEE symposium on adaptive dynamic programming and reinforcement learning, 2009. ADPRL'09* (pp. 153–160). IEEE.

Crites, R. H., & Barto, A. G. (1996). Improving elevator performance using reinforcement learning. In *Advances in neural information processing systems* (pp. 1017–1023).

Dann, C., Neumann, G., & Peters, J. (2014). Policy evaluation with temporal differences: A survey and comparison. *The Journal of Machine Learning Research*, *15*(1), 809–883.

De Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, *134*(1), 19–67.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1), 53–66.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, *12*(1), 219–245.

Eldracher, M., Staller, A., & Pompl, R. (1994). *Function approximation with continuous valued activation functions in CMAC*. Inst. für Informatik.

Hu, J., Fu, M. C., & Marcus, S. I. (2007). A model reference adaptive search method for global optimization. *Operations Research*, *55*(3), 549–568.

Hu, J., Fu, M. C., & Marcus, S. I. (2008). A model reference adaptive search method for stochastic global optimization. *Communications in Information & Systems*, *8*(3), 245–276.

Hu, J., & Hu, P. (2009). On the performance of the cross-entropy method. In *Proceedings of the 2009 winter simulation conference (WSC)* (pp. 459–468). IEEE.

Joseph, A. G., & Bhatnagar, S. (2016). A randomized algorithm for continuous optimization. In *Winter simulation conference, WSC 2016*, Washington, DC, USA, (pp. 907–918).

Joseph, A. G., & Bhatnagar, S. (2016). Revisiting the cross entropy method with applications in stochastic global optimization and reinforcement learning. *Frontiers in artificial intelligence and applications* (ECAI 2016) (Vol. 285, pp. 1026–1034). https://doi.org/10.3233/978-1-61499-672-9-1026

Joseph, A. G., & Bhatnagar, S. (2018). A cross entropy based optimization algorithm with global convergence guarantees. CoRR (arXiv:1801.10291).

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274.

Konidaris, G., Osentoski, S., & Thomas, P. S. (2011). Value function approximation in reinforcement learning using the Fourier basis. In *Twenty-fifth AAAI conference on artificial intelligence*.

Kubrusly, C., & Gravier, J. (1973). Stochastic approximation algorithms and applications. In *1973 IEEE conference on decision and control including the 12th symposium on adaptive processes* (Vol. 12, pp. 763–766).

Kullback, S. (1959). *Statistics and information theory*. New York: Wiley.

Kushner, H. J., & Clark, D. S. (1978). *Stochastic approximation for constrained and unconstrained systems*. New York: Springer.

Kveton, B., Hauskrecht, M., & Guestrin, C. (2006). Solving factored mdps with hybrid state and action variables. *Journal of Artificial Intelligence Research (JAIR)*, *27*, 153–201.

Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *The Journal of Machine Learning Research*, *4*, 1107–1149.

Ljung, L. (1977). Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, *22*(4), 551–575.

Maei, H. R., Szepesvári, C., Bhatnagar, S., Precup, D., Silver, D., & Sutton, R. S. (2009). Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in neural information processing systems* (pp. 1204–1212).

Mannor, S., Rubinstein, R. Y., & Gat, Y. (2003). The cross entropy method for fast policy search. In *International conference on machine learning-ICML 2003* (pp. 512–519).

Menache, I., Mannor, S., & Shimkin, N. (2005). Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, *134*(1), 215–238.

Morris, C. N. (1982). Natural exponential families with quadratic variance functions. *The Annals of Statistics* 65–80.

Mühlenbein, H., & Paass, G. (1996). From recombination of genes to the estimation of distributions i. binary parameters. In *Parallel problem solving from naturePPSN IV* (pp. 178–187). Springer.

Nedić, A., & Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, *13*(1–2), 79–110.

Perko, L. (2013). *Differential equations and dynamical systems* (Vol. 7). Berlin: Springer.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* 400–407.

Rubinstein, R. Y., & Kroese, D. P. (2013). *The cross-entropy method: A unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Berlin: Springer.

Scherrer, B. (2010). Should one compute the temporal difference fix point or minimize the Bellman residual? the unified oblique projection view. In *27th International conference on machine learning-ICML 2010*.

Schoknecht, R. (2002). Optimality of reinforcement learning algorithms with linear function approximation. In *Advances in neural information processing systems* (pp. 1555–1562).

Schoknecht, R., & Merke, A. (2002) Convergent combinations of reinforcement learning with linear function approximation. In *Advances in neural information processing systems* (pp. 1579–1586).

Schoknecht, R., & Merke, A. (2003). TD(0) converges provably faster than the residual gradient algorithm. In *International conference on machine learning-ICML 2003* (pp. 680–687).

Silver, D., Sutton, R. S., & Müller, M. (2007). Reinforcement learning of local shape in the game of go. In *International joint conference on artificial intelligence (IJCAI)* (Vol. 7, pp. 1053–1058).

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning*. New York: MIT Press.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., & Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th annual international conference on machine learning* (pp. 993–1000). ACM.

Sutton, R. S., Maei, H. R., & Szepesvári, C. (2009). A convergent O(n) temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems* (pp. 1609–1616).

Tesauro, G. (1995). Td-gammon: A self-teaching backgammon program. In *Applications of neural networks* (pp. 267–285). Springer.

Tsitsiklis, J. N., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, *42*(5), 674–690.

White, D. J. (1993). A survey of applications of Markov decision processes. *Journal of the Operational Research Society*, *44*, 1073–1096.

Williams, R. J., & Baird, L. C. (1993 Nov 24). Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Techical report NU-CCS-93-14, Northeastern University, College of Computer Science, Boston, MA.

Zhou, E., Bhatnagar, S., & Chen, X. (2014). Simulation optimization via gradient-based stochastic search. In *Winter simulation conference (WSC), 2014* (pp. 3869–3879.) IEEE.

Zlochin, M., Birattari, M., Meuleau, N., & Dorigo, M. (2004). Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research*, *131*(1–4), 373–395.