Mach Learn (2018) 107:969–1011
https://doi.org/10.1007/s10994-018-5697-1

An incremental off-policy search in a model-free Markov decision process using a single sample path

Ajin George Joseph[1] · **Shalabh Bhatnagar**[1]

Received: 4 October 2016 / Accepted: 27 January 2018 / Published online: 13 February 2018

Abstract In this paper, we consider a modified version of the control problem in a model free Markov decision process (MDP) setting with large state and action spaces. The control problem most commonly addressed in the contemporary literature is to find an optimal policy which maximizes the value function, i.e., the long run discounted reward of the MDP. The current settings also assume access to a generative model of the MDP with the hidden premise that observations of the system behaviour in the form of sample trajectories can be obtained with ease from the model. In this paper, we consider a modified version, where the cost function is the expectation of a non-convex function of the value function without access to the generative model. Rather, we assume that a sample trajectory generated using a priori chosen behaviour policy is made available. In this restricted setting, we solve the modified control problem in its true sense, i.e., to find the best possible policy given this limited information. We propose a stochastic approximation algorithm based on the well-known cross entropy method which is data (sample trajectory) efficient, stable, robust as well as computationally and storage efficient. We provide a proof of convergence of our algorithm to a policy which is globally optimal relative to the behaviour policy. We also present experimental results to corroborate our claims and we demonstrate the superiority of the solution produced by our algorithm compared to the state-of-the-art algorithms under appropriately chosen behaviour policy.

Keywords Markov decision process · Off-policy prediction · Control problem · Stochastic approximation method · Cross entropy method · Linear function approximation · ODE method · Global optimization

Editor: Alan Fern.

✉ Ajin George Joseph
ajin@iisc.ac.in

[1] Indian Institute of Science, Bangalore 560012, India

1 Summary of notation

We use $\mathbf{x}$ for random variable and x for deterministic variable. 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 $f_\theta(\cdot)$ denote the *probability density function* 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 a scalar-valued function J, let $\gamma_\rho(J, \theta)$ denote the $(1 - \rho)$-quantile of $J(\mathbf{x})$ w.r.t. f_θ, i.e.,

$$\gamma_\rho(J, \theta) \triangleq \sup\{l : P_\theta(J(\mathbf{x}) \geq l) \geq \rho\}. \tag{1}$$

Let $supp(f) \triangleq \overline{\{x|f(x) \neq 0\}}$ denote the support of f and $interior(A)$ be the *interior* of set A. Let $\mathcal{N}_d(a, B)$ represent the multivariate Gaussian distribution with mean vector a and covariance matrix B. A function $L : \mathbb{R}^m \to \mathbb{R}$ is *Lipschitz continuous*, if $\exists K \geq 0$ *s.t.* $|L(x)-L(y)| \leq K\|x - y\|, \forall x, y \in \mathbb{R}^m$, where $\|\cdot\|$ is a norm defined on $\mathbb{R}^m$. Also, for a matrix $A = [a_{ij}]_{1\leq i\leq m, 1\leq j\leq n} \in \mathbb{R}^{m\times n}$, we define the norm $\|A\|_\infty \triangleq \max_{1\leq i\leq m} \sum_{1\leq j\leq n} |a_{ij}|$ and for invertible matrices, we define the *condition number* $\kappa(A) \triangleq \|A\|_\infty\|A^{-1}\|_\infty$. Also, $|A| \triangleq [|a_{ij}|]_{1\leq i\leq m, 1\leq j\leq n}$. Similarly, for $x \in \mathbb{R}^m$, the sup norm $\|x\|_\infty$ is defined as $\|x\|_\infty \triangleq \sup_i |x_i|$ and $|x| \triangleq (|x_i|)_{1\leq i\leq m}$.

2 Introduction and preliminaries

A discrete time Markov decision process (MDP) (Sutton and Barto 1998; Bertsekas 1995) is a 4-tuple $(\mathbb{S}, \mathbb{A}, R, P)$, where $\mathbb{S}$ denotes the set of *states* and $\mathbb{A}$ is the set of *actions*. Also, $R : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$ is the *reward function* where $R(s, a, s')$ represents the reward obtained in state s after taking action a and transitioning to state s'. Without loss of generality, we assume that the same choice of actions is available for all the states. We also assume that the reward function is bounded, i.e., $\|R\|_\infty < \infty$. We let $P : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to [0, 1]$ denote the *transition probability kernel*, where $P(s, a, s')$ is the probability of next state being $s^{'}$ conditioned on the fact that the current state is s and action taken is a. We assume that the state and action spaces are finite. A *stationary random policy* (SRP) $\pi(\cdot|s)$ is a probability distribution over the action space $\mathbb{A}$ conditioned on state $s \in \mathbb{S}$. A given policy π along with the transition kernel P determines the state dynamics of the system. For a given policy π, the system behaves as a homogeneous Markov chain with transition probabilities

$$P_\pi(s, s') = \sum_{a\in\mathbb{A}} \pi(a|s)P(s, a, s'), s, s' \in \mathbb{S}. \tag{2}$$

In this paper, we consider only stationary randomized policies. We also assume that given an SRP π, the Markov chain induced by P_π is ergodic, i.e., the Markov chain is irreducible and aperiodic.

The two fundamental questions most commonly addressed in the MDP literature are: 1. *Prediction problem* and 2. *Control problem*.

Prediction problem For a given SRP π and *discount factor* $\gamma \in (0, 1)$, the objective is to evaluate the long-run γ-discounted cost $V^\pi \in \mathbb{R}^{|\mathbb{S}|}$ which is defined as

$$V^\pi(s) \triangleq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R(\mathbf{s}_k, \mathbf{a}_k, \mathbf{s}_{k+1})\middle|\mathbf{s}_0 = s\right], s \in \mathbb{S}, \tag{3}$$

where the random variable $\mathbf{s}_k$ represents the state at instant k, the random variable $\mathbf{a}_k$ represents the action chosen at instant k and the random variable $\mathbf{s}_{k+1}$ represents the transitioned state after instant k, i.e., the state at instant $k+1$. Further, $\mathbb{E}_\pi[\cdot]$ is the expectation w.r.t. the probability distribution induced by P_π with initial state $\mathbf{s}_0 = s$. Note that the cost evaluation in (3) is realistic and prudent. Since MDP is a sequential decision making paradigm, the discount factor γ controls the width of the window of future events to be considered to guide the decision process. For γ close to 0, only the rewards pertaining to the first few transitions count as the effect of the future rewards whose weights are geometric in γ is minimal. However, the case of γ very close to 1 requires a very long window to be considered.

For a given policy π, the value function V^π satisfies the following *Bellman equation* (written in vector-matrix notation):

$$V^\pi = T^\pi V^\pi, \tag{4}$$

where T^π called the *Bellman operator* is defined as $T^\pi V \triangleq R^\pi + \gamma P_\pi V$ and $R^\pi(s) \triangleq \sum_{a\in\mathbb{A}} \pi(a|s) \sum_{s'\in\mathbb{S}} P(s,a,s')R(s,a,s')$. Hence V^π can be directly computed as $V^\pi = (I - \gamma P_\pi)^{-1} R^\pi$. The computational complexity of the above direct computation is $O(|\mathbb{S}|^3)$ and the space complexity is $O(|\mathbb{S}|^2)$. An alternate procedure to solve the prediction problem is *value iteration* that is based on the contraction mapping theorem. It is easy to see that the Bellman operator T^π is a contraction mapping with the contraction constant γ. Hence by the contraction mapping theorem, $(T^\pi)^k V \to V^\pi$ as $k \to \infty$, $\forall V \in \mathrm{I\!R}^{|\mathbb{S}|}$. The computational complexity of this successive approximation procedure is $O(|\mathbb{S}|^2)$ per iteration and the space complexity is $O(|\mathbb{S}|^2)$ as well. The state space $\mathbb{S}$ can be huge, for example, in cases where the state is represented as a high-dimensional vector. The cardinality of the state space in such a case is exponential in the dimension resulting in a corresponding exponential upsurge in computational effort and storage requirement. In such cases, the above method can become well-nigh intractable. This predicament is referred to in the literature as the *curse of dimensionality*. One commonly employed heuristic to circumvent the curse is the *state aggregation* (Bertsekas and Castanon 1989) technique. However, it also suffers dearly when the state space is huge.

Control problem The objective for this problem is to find the optimal stationary policy π^* of the MDP, where

$$\pi^*(s) \in \arg\max_{\pi} V^\pi(s), s \in \mathbb{S}. \tag{5}$$

The existence of an optimal stationary policy is proven in Puterman (2014). The optimal value function $V^*(= V^{\pi^*})$ satisfies the *Bellman optimality equation* given by: $TV^* = V^*$, where the *Bellman optimality operator* T is defined as $TV(s) \triangleq max_{a\in\mathbb{A}} \sum_{s'\in\mathbb{S}} P(s,a,s')(R(s,a,s') + \gamma V(s'))$. The primary numerical methods which solve the control problem are the *value iteration* and *policy iteration*. A detailed description of these methods is available in Puterman (2014). In a nutshell, policy iteration can be characterized as generating a sequence of improving policies $\{\pi_k\}_{k\in\mathbb{N}}$ with π_k converging to π^* after a finite number of steps. Value iteration on the other hand involves repeated application of the Bellman optimality operator, which requires multiple extensive passes over the state space and the convergence is only guaranteed asymptotically. The computational complexities of policy iteration and value iteration are $O(|\mathbb{S}|^2|\mathbb{A}| + |\mathbb{S}|^3)$ and $O(|\mathbb{S}|^2|\mathbb{A}|)$ respectively. The space complexity of both the methods is the same and it is $O(|\mathbb{S}| + |\mathbb{A}|)$. The super-linear dependency of the methods on the size of state space results in the curse of dimensionality. A recently proposed policy iteration method based on stochastic factorization

(Barreto et al. 2014) has reduced the dependency to linear terms. However, when $\mathbb{S}$ is very large, stochastic factorization also becomes intractable.

2.1 Model free algorithms

In the above section, the prediction and control algorithms are numerical methods that assume that the probability transition function P and the reward function R are available. In most of the practical scenarios, it is unrealistic to assume that accurate knowledge of P and R is realizable. However, the behaviour of the system can be observed and one needs to either predict the value of a given policy or find the optimal control using the available observations. The observations are in the form of a sample trajectory $\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, \dots\}$, where $s_i \in \mathbb{S}$ is the state and $r_i = R(s_i, a_i, s_{i+1})$ is the immediate reward at time instant i. Model free algorithms are basically of three types: (i) Indirect methods, (ii) Direct methods and (iii) Policy search methods. The last of these methods searches in the policy space to find the optimal policy where the performance measure used for comparison is the estimate of the value function induced from the observations. Prominent algorithms in this category are actor-critic (Konda and Tsitsiklis 2003), policy gradient (Baxter and Bartlett 2001), natural actor-critic (Bhatnagar et al. 2009) and fast policy search (Mannor et al. 2003). Indirect methods are based on the certainty equivalence of computing where initially the transition matrix and the expected reward vector are estimated using the observations and subsequently, model based approaches mentioned in the above section are applied on the estimates. A few indirect methods are control learning (Sato et al. 1982, 1988; Kumar and Lin 1982), priority sweeping (Moore and Atkeson 1993), adaptive real-time dynamic programming (ARTDP) (Barto et al. 1995) and PILCO (Deisenroth and Rasmussen 2011). For the case of direct methods which are more appealing, the model is not estimated, rather the control policy is adapted iteratively using a shadow utility function derived from the instantiation of the internal dynamics of the MDP. The algorithms in this class are generally referred to in the literature as the *reinforcement learning* algorithms. Prominent reinforcement learning algorithms include *temporal difference (TD) learning* (Sutton 1988) (prediction method), *Q-learning* (Watkins 1989) and *SARSA* (Singh and Sutton 1996) (control methods). There are two variants of the prediction algorithm depending on how the sample trajectory is generated. They are *on-policy* and *off-policy* algorithms. In the on-policy case, the sample trajectory is generated using the policy π which is being evaluated, i.e., $s_{i+1} \sim P(s_i, a_i, \cdot)$, where $a_i \sim \pi(\cdot|s_i)$ and $r_i = R(s_i, a_i, s_{i+1})$. In the off-policy case, the sample trajectory is generated using a policy π_b which is possibly different from the policy π that is being evaluated, i.e., $s_{i+1} \sim P(s_i, a_i, \cdot)$, where $a_i \sim \pi_b(\cdot|s_i)$ and $r_i = R(s_i, a_i, s_{i+1})$.

Model free algorithms are shown to be robust, stable and exhibit good convergence behaviour under realistic assumptions. However, they suffer from the curse of dimensionality which arises due to the space complexity. Note that the space complexity of the above mentioned learning algorithms is $O(|\mathbb{S}|)$, which becomes unmanageably large with increasing state space.

2.2 Linear function approximation (LFA) methods for model free Markov decision process

To tackle the curse of dimensionality and to achieve tractability, it is imperative to eliminate the dependency both in terms of the computational and storage requirements of the learning methods on the cardinalities of state and action spaces. An efficient approach is to compactly yet effectively represent the system in a lower k_1-dimensional space, where $k_1 \ll |\mathbb{S}|$. A well understood dimensionality reduction technique is the linear function approximation. Here, we choose a collection of *prediction features* $\{\phi_i\}_{i=1}^{k_1}$, where $\phi_i \in \mathbb{R}^{|\mathbb{S}|}$. In this case, the prediction task becomes a projection where

$$\Pi V^{\pi} = \underset{h \in \mathbb{H}^{\Phi}}{\arg\min} \| V^{\pi} - h \|^2, \tag{6}$$

where $\mathbb{H}^{\Phi} \triangleq \{\Phi x | x \in \mathbb{R}^{k_1}\} \subset \mathbb{R}^{|\mathbb{S}|}$ is the space of representable functions with $\Phi \triangleq (\phi_1, \ldots, \phi_{k_1}) \in \mathbb{R}^{|\mathbb{S}| \times k_1}$ and the norm $\| \cdot \|$ is chosen appropriately according to the domain. Note that $\mathbb{H}^{\Phi}$ is a linear function space. Further, we define $\phi(s) \triangleq (\phi_1(s), \ldots, \phi_{k_1}(s))^{\top}$, $s \in \mathbb{S}$. Note that ϕ_i can be viewed as a function from $\mathbb{S}$ to $\mathbb{R}$. Similarly, the control problem becomes $\pi^*(s) \in \underset{\pi}{\arg\max}\, \Pi V^{\pi}(s), \forall s \in \mathbb{S}$. Note that in the case of large and complex MDPs, the features are not hard-coded, instead one employs compact representations in the form of basis functions. Examples of basis functions include radial basis functions and Fourier basis.

To address the computational and storage concerns arising due to large action space, a sagacious approach is to employ a parametrized class of SRPs $\{\pi_w | w \in \mathbb{W} \subset \mathbb{R}^{k_2}\}$, where $k_2 \in \mathbb{N}$, instead of an exact representation. The most commonly used is the Gibbs (or Boltzmann) "soft-max" class of policies. In this case, for a given $w \in \mathbb{W} \subset \mathbb{R}^{k_2}$, the SRP π_w is defined as

$$\pi_w(a|s) = \frac{\exp\left(w^{\top}\psi(s,a)/\tau\right)}{\sum_{b \in \mathbb{A}} \exp\left(w^{\top}\psi(s,b)/\tau\right)}, \tag{7}$$

where $\{\psi(s,a) \in \mathbb{R}^{k_2} | s \in \mathbb{S}, a \in \mathbb{A}\}$ is a given *policy feature set* and $\tau \in \mathbb{R}_+$ is fixed *a priori*.

The accuracy of the function approximation method depends on the representational/expressive ability of $\mathbb{H}^{\Phi}$. For example, when $k_1 = |\mathbb{S}|$, the representational ability is utmost, since $\mathbb{H}^{\Phi} = \mathbb{R}^{|\mathbb{S}|}$. In general, $k_1 \ll |\mathbb{S}|$ and hence $\mathbb{H}^{\Phi} \subset \mathbb{R}^{|\mathbb{S}|}$. So for an arbitrary policy π, where $V^{\pi} \notin \mathbb{H}^{\Phi}$, the prediction of the value function V^{π} shall always incur an unavoidable *approximation error* (e_{appr}) given by $\inf_{h \in \mathbb{H}^{\Phi}} \|V^{\pi} - h\|$. Given $\mathbb{H}^{\Phi}$, one cannot perform no better than e_{appr}. The prediction features $\{\phi_i\}$ are hand-crafted using prior domain knowledge and their choice is critical in approximating the value function. There is an abundance of literature available on the topic. In this paper, we assume that an appropriately chosen feature set is available *a priori*. Also note that the convergence of the prediction methods is in asymptotic sense. But in most practical scenarios, the algorithm has to be terminated after a finite number of steps. This incurs an *estimation error* (e_{est}) which however decays to zero, asymptotically.

Even though LFA produces sub-optimal solutions, since the search is conducted on a restricted subspace of $\mathbb{R}^{|\mathbb{S}|}$, it yields large computational and storage benefits. So some degree of trade-off between accuracy and tractability is indeed unavoidable.

2.3 Off-policy prediction using LFA

Setup Given $w, w_b \in \mathbb{W}$ and an observation of the system dynamics in the form of a sample trajectory $\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, \ldots\}$, where at each instant k, $a_k \sim \pi_{w_b}(\cdot|s_k)$, $s_{k+1} \sim P(s_k, a_k, \cdot)$ and $r_k = R(s_k, a_k, s_{k+1})$, the goal is to estimate the value function V^{π_w} of the target policy π_w (that is possibly different from π_{w_b}). We assume that the Markov chains defined by P_w and P_{w_b} are ergodic. Further, let ν_w and ν_{w_b} be the stationary distributions of the Markov chains with transition probability matrices P_w and P_{w_b} respectively, i.e., $\lim_{k \to \infty} P_w(\mathbf{s}_k = s) = \nu_w(s)$ and $\nu_w^{\top} P_w = \nu_w^{\top}$ and likewise for ν_{w_b}. Note that for brevity the notations have been simplified here, i.e., $P_w \triangleq P_{\pi_w}$ and $P_{w_b} \triangleq P_{\pi_{w_b}}$. We follow the new notation for the rest of the paper. Similarly, $V^w \triangleq V^{\pi_w}$.

In the off-policy learning case, the projection is w.r.t. the norm $\|\cdot\|_{\nu_{w_b}}$, where $\|V\|^2_{\nu_{w_b}} =< V, V >_{\nu_{w_b}}$. The inner product is defined as $< V_1, V_2 >_\nu = V_1^\top D^\nu V_2$, where $V_1, V_2 \in \mathbb{R}^{|\mathbb{S}|}$, $\nu \in [0,1]^{|\mathbb{S}|}$ is a probability mass function over $\mathbb{S}$ and D^ν is a $|\mathbb{S}| \times |\mathbb{S}|$ diagonal matrix with $D^\nu_{ii} = \nu(i)$, $1 \leq i \leq |\mathbb{S}|$. Thus the norm $\|\cdot\|_{\nu_{w_b}}$ is in fact the Euclidean norm weighted with the stationary distribution ν_{w_b} of the behaviour policy π_{w_b}, i.e., $\|V\|_{\nu_{w_b}} \triangleq \sqrt{\sum_{s \in \mathbb{S}} \nu_{w_b}(s) V^2(s)}$. So

$$h_{w|w_b} \triangleq \Pi^{w_b} V^w = \operatorname*{arg\,min}_{h \in \mathbb{H}^\Phi} \|V^w - h\|^2_{\nu_{w_b}}, \tag{8}$$

where Π^{w_b} denotes the projection operator w.r.t. $\|\cdot\|_{\nu_{w_b}}$ whose closed form expression can be derived as follows:

$$\begin{aligned}
\nabla_x \|V^w - h\|^2_{\nu_{w_b}} &= 0 \\
&\Rightarrow \nabla_x (V^w - \Phi x)^\top D^{\nu_{w_b}} (V^w - \Phi x) = 0 \\
&\Rightarrow \Phi^\top D^{\nu_{w_b}} (V^w - \Phi x) = 0 \\
&\Rightarrow \Phi^\top D^{\nu_{w_b}} \Phi x = \Phi^\top D^{\nu_{w_b}} V^w \\
&\Rightarrow x = (\Phi^\top D^{\nu_{w_b}} \Phi)^{-1} \Phi^\top D^{\nu_{w_b}} V^w \\
&\Rightarrow \Phi x = \Phi(\Phi^\top D^{\nu_{w_b}} \Phi)^{-1} \Phi^\top D^{\nu_{w_b}} V^w. \\
&\quad \therefore \Pi^{w_b} = \Phi(\Phi^\top D^{\nu_{w_b}} \Phi)^{-1} \Phi^\top D^{\nu_{w_b}}.
\end{aligned} \tag{9}$$

⊛ **Assumption (A1)** *The prediction features* $\{\phi_i\}_{i=1}^{k_1}$ *are linearly independent.*

Algorithms The evaluation of Π^{w_b} requires knowledge of the stationary distribution ν_{w_b} which can only be derived if the transition matrix P_{w_b} is available. However, in model free learning P_{w_b} is hidden and hence all the state-of-the-art methods can only derive an approximation to the projection. Two pertinent algorithms are *off-policy TD*(λ) and *off-policy LSTD*(λ). The algorithms return a prediction vector $x \in \mathbb{R}^{k_1}$ *s.t.* $\Phi x \approx h_{w|w_b}$. The major technique used in both the algorithms is to correct the discrepancies between the target and behaviour policies using *importance sampling* (Glynn and Iglehart 1989). Here we introduce the sampling ratio at time k to be $\rho_k \triangleq \frac{\pi_w(a_k|s_k)}{\pi_{w_b}(a_k|s_k)}$, where we use the convention $0/0 = 0$.

- **Off-policy TD**(λ)

Off-policy TD(λ) (Yu 2012, 2015), where $\lambda \in [0, 1]$ is one of the fundamental algorithms to approximate value function using linear architecture. The algorithm is defined as follows:

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_{k+1}\delta_{k+1}\mathbf{e}_k, \tag{10a}$$

$$\mathbf{e}_{k+1} := \gamma\lambda\rho_k\mathbf{e}_k + \phi(s_k), \tag{10b}$$

where $\mathbf{e}_k, \mathbf{x}_k \in \mathbb{R}^{k_1}$ and $\delta_{k+1} \triangleq \rho_k r_k + \gamma\rho_k \mathbf{x}_k^\top \phi(s_{k+1}) - \mathbf{x}_k^\top \phi(s_k)$ is called the *temporal difference error*. The learning rate α_k is non-negative, deterministic and satisfies $\sum_k \alpha_k = \infty$, $\sum_k \alpha_k^2 < \infty$. The vector $\mathbf{e}_k \in \mathbb{R}^{k_1}$ is called the *eligibility trace* and is used for variance reduction. Eligibility traces accelerate the learning process by integrating temporal differences from multiple time steps. The convergence analysis of the off-policy TD(λ) method is provided in Yu (2012). However, the analysis assumes that the iterates $\mathbf{x}_k \in \bar{B}_r(0), \forall k \geq 0$, with $r > 0$ being sufficiently large. The convergence of the un-constrained case for λ close to 1 is proved in Yu (2015).

- **Off-policy LSTD(λ)**

Off-policy least squares temporal difference (LSTD) with eligibility traces (Yu 2012) is another relevant algorithm in this category. The procedure is described below:

$$\mathbf{e}_{k+1} := \gamma\lambda\rho_k\mathbf{e}_k + \phi(s_k), \tag{11a}$$

$$\mathbf{A}_{k+1} := \mathbf{A}_k + \frac{1}{k+1}\Big(\mathbf{e}_k(\phi(s_k) - \gamma\rho_k\phi(s_{k+1}))^\top - \mathbf{A}_k\Big), \tag{11b}$$

$$\mathbf{b}_{k+1} := \mathbf{b}_k + \frac{1}{k+1}(\rho_k r_k \mathbf{e}_k - \mathbf{b}_k), \tag{11c}$$

$$\mathbf{x}_{k+1} := \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}, \tag{11d}$$

where $\mathbf{A}_k \in \mathbb{R}^{k_1 \times k_1}$ and $\mathbf{e}_k, \mathbf{b}_k, \mathbf{x}_k \in \mathbb{R}^{k_1}$. In some cases, the matrix $\mathbf{A}_k$ may not be of full rank. To avoid such singularities, initialize $\mathbf{A}_0$ with $\delta\mathbb{1}_{k_1 \times k_1}$, $\delta > 0$.

Contrary to the earlier algorithm, the off-policy LSTD(λ) is shown to be stable with well defined limiting behaviour for all $\lambda \in [0, 1]$ under pragmatic assumptions. The only restriction imposed is that the target policy π_{w_b} is *absolutely continuous* ($\prec$) w.r.t. the behaviour policy π_{w_b}, i.e.,

$$\pi_w \prec \pi_{w_b} \;\Leftrightarrow\; \pi_{w_b}(a|s) = 0 \Rightarrow \pi_w(a|s) = 0, \forall a \in \mathbb{A}, \forall s \in \mathbb{S}. \tag{12}$$

The contrapositive form of the above statement implies that $\pi_w(a|s) \neq 0 \Rightarrow \pi_{w_b}(a|s) \neq 0$, $\forall a \in \mathbb{A}, \forall s \in \mathbb{S}$. This means that for a given state $s \in \mathbb{S}$, every action feasible under the target policy π_w is also feasible under the behaviour policy π_{w_b}. The following result from Yu (2012) characterizes the limiting behaviour of the off-policy LSTD(λ) algorithm:

Theorem 1 *For a given target policy vector $w \in \mathbb{W}$ and a behaviour policy vector $w_b \in \mathbb{W}$, the sequence $\{\mathbf{x}_k\}$ generated by the off-policy LSTD(λ) algorithm defined in Eq. (11) converges to the limit $x_{w|w_b}$ with probability one, where*

$$\begin{aligned} x_{w|w_b} = A^{-1}_{w|w_b} b_{w|w_b}, \textit{ with} & \\ A_{w|w_b} &= \Phi^\top D^{\nu_{w_b}}(I - \gamma\lambda P_w)^{-1}(I - \gamma P_w)\Phi \textit{ and} \\ b_{w|w_b} &= \Phi^\top D^{\nu_{w_b}}(I - \gamma\lambda P_w)^{-1} R^w. \end{aligned} \tag{13}$$

Here $D^{\nu_{w_b}}$ is the diagonal matrix with $D^{\nu_{w_b}}_{ii} = \nu_{w_b}(i)$, $1 \leq i \leq |\mathbb{S}|$, where ν_{w_b} is the stationary distribution of the Markov chain P_{w_b} induced by the behaviour policy π_{w_b}, i.e., ν_{w_b} satisfies $\nu^\top_{w_b} P_{w_b} = \nu^\top_{w_b}$ and $R^w(s) \in \mathbb{R}^{|\mathbb{S}|}$ is the expected reward, i.e., $R^w \triangleq \Sigma_{s' \in \mathbb{S}, a \in \mathbb{A}} \pi_w(a|s) P(s, a, s') R(s, a, s')$.

It is also important to note that in the on-policy LSTD(λ), where both π_w and π_{w_b} are the same, the limit point $x_{w|w}$ is given by $x_{w|w} = A^{-1}_{w|w} b_{w|w}$, where

$$\begin{aligned} A_{w|w} &= \Phi^\top D^{\nu_w}(I - \gamma\lambda P_w)^{-1}(I - \gamma P_w)\Phi \;\text{ and} \\ b_{w|w} &= \Phi^\top D^{\nu_w}(I - \gamma\lambda P_w)^{-1} R^w. \end{aligned} \tag{14}$$

2.4 The control problem of interest

In this section, we define a variant of the control problem which is the topic of interest in this paper.

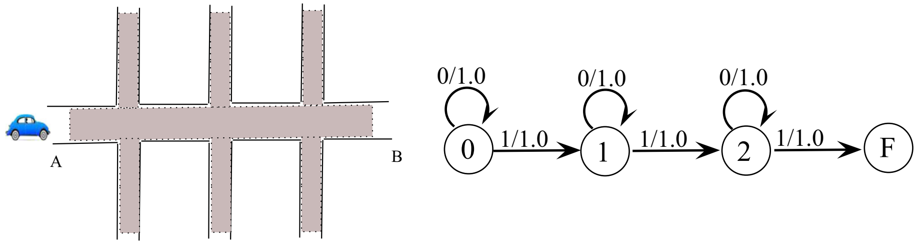


Fig. 1 Self-drive system

Problem Statement

$$\text{Find } w^* \in \underset{w \in \mathbb{W} \subset \mathbb{R}^{k_2}}{\arg\max}\ \mathbb{E}_{\nu_w}\left[L(h_{w|w})\right], \tag{15}$$

where $L : \mathbb{R}^{|\mathbb{S}|} \to \mathbb{R}^{|\mathbb{S}|}$ is a performance function. We assume that L is bounded and continuous. Note that since $h_{w|w} \in \mathbb{R}^{|\mathbb{S}|}$, we have $L(h_{w|w}) \in \mathbb{R}^{|\mathbb{S}|}$, i.e., $L(h_{w|w})$ can be viewed as a mapping from the the state space $\mathbb{S}$ to the scalars. In the case of finite MDP (both $\mathbb{S}$ and $\mathbb{A}$ are finite), we have $\mathbb{E}_{\nu_w}\left[L(h_{w|w})\right] = \sum_{s \in \mathbb{S}} \nu_w(s) L(h_{w|w})(s)$. Thus the objective function in Eq. (15) is scalar-valued and hence the optimization problem defined in Eq. (15) is indeed well-defined.

⊛ **Assumption (A2)** *The Markov chain under any SRP* π_w, $w \in \mathbb{R}^{k_2}$ *is ergodic*, i.e., *irreducible and aperiodic.*

⊛ **Assumption (A3)** $\mathbb{W}$ *is a compact subset of* $\mathbb{R}^{k_2}$.

2.5 Motivation

We demonstrate here a practical situation where the optimization problem of the kind (15) arises. We consider here a special case of the self-drive system (Fig. 1). The goal is to propel an automotive (equipped with sensors to detect the vehicular traffic) from source 0 to destination F (where there are multiple intersections in between) in minimum time without any accidents. Here, the collection of junctions represents the state space, i.e., $\mathbb{S} = \{0, 1, 2, 3, F\}$. The automotive travels with a constant velocity between subsequent intersections and the choice of the velocities is restricted to the discrete, finite set $\{1, 2, 3\}$. The velocity is chosen randomly by the automotive from the above set at each intersection. The purpose of the randomness is to capture the uncertainty in the traffic conditions during the subsequent stretch of the trip. At each intersection, the automotive senses the vehicular traffic at the intersection and has to make a choice of whether to halt or not. So the action space is $\mathbb{A} = \{0 \text{ (halt)}, 1 \text{ (proceed)}\}$. Here, the performance of the task is evaluated based on the overall time the automotive takes to cover the distance to the destination. Hence the reward function is taken as the velocity chosen by the automotive to traverse the subsequent stretch. This indeed makes sense since the time is directly dependent on the velocity with distance being constant. This optimization problem can be modeled using a finite horizon cost function. Now, suppose that the task is further rewarded based on the overall time it takes to complete the trip. In this case, the final payoff is dependent on the value function (in this case, the value function is time), then the role of the performance function L is to capture this particular aspect. If the payoffs are further based on the maintenance cost incurred (which cannot be integrated into the reward function due to the presence of multiple operating components and hence considering the net maintenance cost at the end of the episode is more worthwhile), the performance function might not be

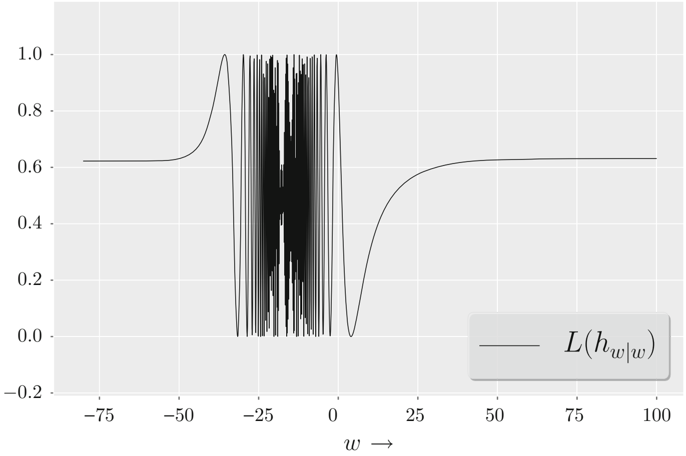


Fig. 2 $\mathbb{S} = \{0, 1, 2, F\}$, $\mathbb{A} = \{0, 1\}$, $k_1 = 1$, $k_2 = 1$, $\gamma = 0.99$, $\tau = 10$, $\lambda = 0.00125$, $\psi(s, a) = s * a$, $\Phi = (1, 0, 1, 0)^\top$, $L(h_{w|w})(s) = \sin^2(\frac{\pi}{2}s)$, $P(0, 0, 0) = P(1, 0, 1) = P(2, 0, 2) = 1.0$, $P(0, 1, 1) = P(1, 1, 2) = P(2, 1, F) = 1.0$. The remaining transition probabilities are zero

unimodal in general. This is further confirmed by Fig. 2, where we provide the plot of the objective function of the self-drive MDP which exhibits a complex landscape with many local optima. This particular problem is more relevant in the context of neural computation, where distinct neural substrates in regions of prefrontal and anterior striatum have been identified with human habitual learning (model free reinforcement learning) (ODoherty et al. 2015; Balleine and Dickinson 1998; Lee and Shimojo 2014). The human brain is a complex network of computing components and one is inclined to believe that the value function obtained through the habitual learning will be further evaluated using a performance function (similar to the activation function found in the artificial neural networks) before relaying to the subsequent level in the network.

The control problem in Eq. (15) is harder due to the application of the performance function L on the approximate value function. Hence we cannot apply the existing direct model free methods like LSPI or off-policy Q-learning (Maei et al. 2010). Note that the LSPI algorithm [Fig. 8 of Lagoudakis and Parr (2003)] is a policy iteration method, where at each iteration an improved policy parameter is deduced from the projected Q-value of the previous policy parameter. So one cannot directly incorporate the operator $\mathbb{E}_{\nu_w}$ into the LSPI iteration. Similar compatibility issues are found with the off-policy Q-learning also (Maei et al. 2010). However, policy search methods are a direct match for this problem. Not all policy search methods can provide quality solutions. The pertinent issue is the non-convexity of $\mathbb{E}_{\nu_w}\left[L(h_{w|w})\right]$ which presents a landscape with many local optima. Any gradient based method like the state-of-the-art simultaneous perturbation stochastic approximation (SPSA) (Spall 1992) algorithm or the policy gradient methods can only provide sub-optimal solutions. In this paper, we try to solve the control problem in its *true sense*, i.e., find a solution close to the global optimum of the optimization problem (15). We employ a stochastic approximation variant of the well known cross entropy (CE) method proposed in Joseph and Bhatnagar (2016b, c, a) to achieve the *true sense* behaviour. The CE method has in fact been applied to the model free control setting before in Mannor et al. (2003), where the algorithm is termed the *fast policy search*. However, the approach in Mannor et al. (2003) has left several practical and computational challenges uncovered. The method in Mannor et al. (2003) assumes access to a generative model, i.e., the real MDP system itself or a simulator/computational model of the MDP under consideration, which can be configured with moderate ease (with time constraints) and the observations recorded. The existence of generative models for extremely complex MDPs is highly unlikely, since it demands accurate knowledge about the transition dynamics of the MDP. Now regarding the computational aspect, the algorithm in Mannor et al. (2003) maintains an evolving $|\mathbb{S}| \times |\mathbb{A}|$ matrix $P^{(t)} \triangleq (P_{sa}^{(t)})_{s \in \mathbb{S}, a \in \mathbb{A}}$, where $P_{sa}^{(t)}$ is the probability of taking action a in state s at time t. At each discrete time instant t, the algorithm generates multiple sample trajectories using $P^{(t)}$, each of finite length, but sufficiently long. For each trajectory, the discounted cost is calculated and then averaged over those multiple trajectories to deduce the subsequent iterate $P^{(t+1)}$. This however is an expensive operation, both computation and storage wise. Another pertinent issue is the number of sample trajectories required at each time instant t. There is no analysis pertaining to finding a bound on the trajectory count. This implies that a brute-force approach has to be adopted which further burdens the algorithm. A more recent global optimization algorithm called the model reference adaptive search (MRAS) has also been applied in the model free control setting (Chang et al. 2013). However, it also suffers from similar issues as the earlier approach.

Here, we illustrate using a real life scenario, the hardness incurred in assuming a generative model. We consider a legacy water delivery system (Feinberg and Shwartz 2012; Fracasso et al. 2014; Ikonen and Bene 2011; Ertin et al. 2001). The legacy water delivery systems in most cases are not electronically controlled, which implies that a manual intervention is required to adjust the various throughput levels. The reservoir operators have to rely on agreed upon rules, their judgement and experience to calibrate the network. Figure 3 shows a water delivery network where there is a web of manual controllers. The state space is the net output (quantity of water delivered) of the delivery system. Intuitively, one might expect the dynamics of the system to be Markovian in character since the immediate future output is indeed dependent on the current quantity of the reservoir and its current consumption rate. So the state variable takes real values and the underlying MDP is continuous. The reward function is a complex function with positive weights on profits from effective utilization (agriculture, drinking purpose, power generation, *etc*) and negative weights on spill overs, kinetic energy losses and factors engendering physical damage to the network like excessive pipe pressure. The objective is to find a configuration for the network of controllers (which is indeed a vector with each co-ordinate deciding the amount of calibration required for the corresponding controller) which provides optimum expected discounted reward. Here the configurations represent the action space and thus are also continuous. The reconfiguration of the whole system as and when demanded by the algorithm requires heavy human labor, which is a luxury one cannot afford. On the other hand, developing a simulator for this system requires understanding all the sources of water for the reservoir which depends on a wide variety of environmental factors and also the consumption statistics of the end users, both of which require observations for a long period of time notwithstanding the human labor incurred. Therefore, it is hard in general to develop a simulator/generative model for MDPs with large state and action spaces with complex, opaque and perplexing transition dynamics. Examples where similar issues arise can be found in manual human control, social sciences, biological systems, unmanned aerial vehicles (Bagnell and Schneider 2001) and mechanical systems which wear out quickly like low-cost robots (Deisenroth and Rasmussen 2011).

A few relevant work in the literature which do not assume the availability of a generative model include Bellman-residual minimization based fitted policy iteration using a single trajectory (Antos et al. 2008) and value-iteration based fitted policy iteration using a single trajectory (Antos et al. 2007). However, those approaches fall prey to the curse of dimensionality arising from large action spaces. Also, they are abstract in the sense that a generic

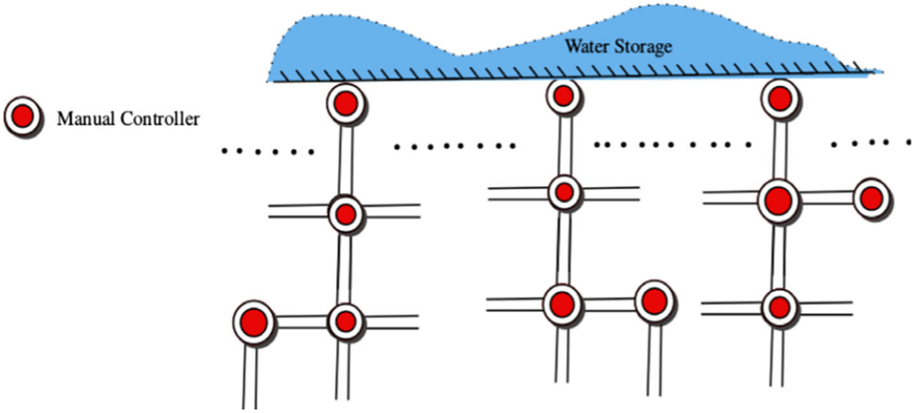


Fig. 3 Water delivery network: the system consists of a water reservoir and a web of manual controllers. The quantity of water in the reservoir is stochastic in nature and so is the consumption of the water by the end users. The end usage of the system includes agriculture, household activities, power generation etc. The reward function is a complex function with positive weights on profits from effective utilization and negative weights on spill overs

function space is considered and the value function approximation step is expressed as a formal optimization problem. In the above methods which are almost similar in their approach, considerable effort is dedicated to addressing the approximation power of the function space and sample complexity.

In this paper, we address the above mentioned practical and computational concerns. We focus on two key objectives:

1. To reduce the total number of policy evaluations.
2. To find a high performing policy without presuming an unlimited access to the generative model.

By accomplishing the above mentioned objectives, we try to chisel down the requirements inherent in most of the reinforcement learning algorithms and thus enable them to operate in real-time scenarios. We provide here a brief narrative of the approach we follow to realize the above objectives.

To accomplish the former objective, the ubiquitous choice is to employ the stochastic approximation (SA) version of the CE method instead of the naive CE method used in Mannor et al. (2003). The SA version of CE is a zero-order optimization method which is incremental, adaptive, robust and stable with the additional attractive attribute of convergence to the global optimum of the objective function. It has been demonstrated empirically in Joseph and Bhatnagar (2016a, b) that the method exhibits efficient utilization of the samples and possesses better rate of convergence than the naive CE method. The effective sample utilization implies that the method requires minimum number of objective function evaluations. These attributes are appealing in the context of the control problem we consider here, especially in effectively addressing the former objective. The adaptive nature of the algorithm apparently eliminates any brute-force approach which has a detrimental impact on the performance of the naive CE method.

The latter objective is achieved by employing the off-policy LSTD(λ) for policy evaluation which is defined in Sect. 2.3. The advantage of this method lies in its ability to approximate the value function of an arbitrary policy (called the target policy) using the observations of the MDP under a possibly different policy (called the behaviour policy), with the only restriction

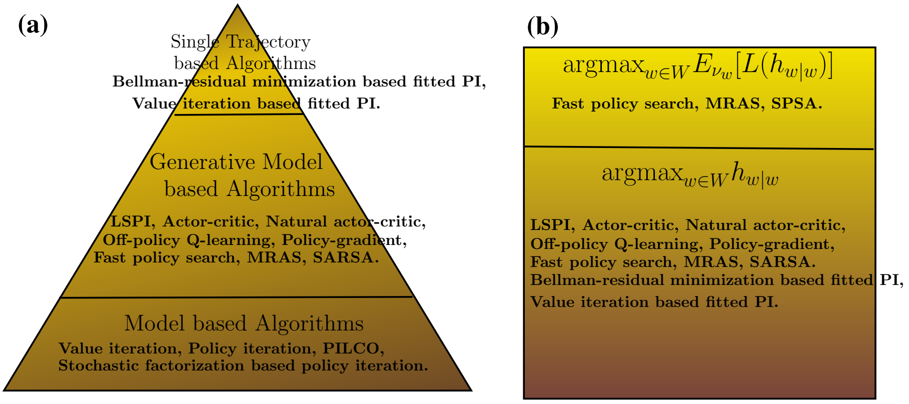

(a)
Single Trajectory based Algorithms
Bellman-residual minimization based fitted PI, Value iteration based fitted PI.
Generative Model based Algorithms
LSPI, Actor-critic, Natural actor-critic, Off-policy Q-learning, Policy-gradient, Fast policy search, MRAS, SARSA.
Model based Algorithms
Value iteration, Policy iteration, PILCO, Stochastic factorization based policy iteration.
(b)
$\mathrm{argmax}_{w \in W} E_{\nu_w}[L(h_{w|w})]$
Fast policy search, MRAS, SPSA.
$\mathrm{argmax}_{w \in W} h_{w|w}$
LSPI, Actor-critic, Natural actor-critic, Off-policy Q-learning, Policy-gradient, Fast policy search, MRAS, SARSA. Bellman-residual minimization based fitted PI, Value iteration based fitted PI.

Fig. 4 **a** Information pyramid. **b** Optimization box

being the absolute continuity between the target and behaviour policies. This implies that we optimize the approximate objective function given by $\mathbb{E}_{\nu_{w_b}}\left[L(\Phi x_{w|w_b})\right]$ (where $x_{w|w_b}$ is the solution generated by the off-policy LSTD(λ)) instead of the true objective function $\mathbb{E}_{\nu_w}\left[L(h_{w|w})\right]$. Here, ν_{w_b} is the steady state distribution of the Markov chain induced by the behaviour policy π_{w_b}. This is the best approximation possible under the absence of the generative model since ν_w is the long-run steady state marginal distribution of the Markov chain induced by the policy π_w and one cannot correct the long-run discrepancies arising due to the restriction that the available sample trajectory is generated using the behaviour policy. However, hidden deep under the appealing characteristic of the single sample trajectory approach is the painful Achilles heel of choice, where one cannot forget that the quality of the solution contrived by the algorithm depends on the choice of the sample trajectory which is directly dependent on the behaviour policy that generates it. The additional approximation error incurred due to this particular information restrictive setting is indeed unavoidable. In order to choose the behaviour policy wisely, it is imperative to provide a quantitative analysis of the cost incurred in the choice of the behaviour policy. In this paper, we provide a bound on the approximation error of the off-policy LSTD(λ) solution of an arbitrary target policy with respect to the deviation of the target policy from the behaviour policy. The practical aspect of the approach can be further improved by reconsidering the same sample trajectory for all value function evaluations. This implies that our algorithm just requires a single sample trajectory to solve the optimization problem defined in Eq. (15). Since the access to the generative model is forbidden, in order to reuse the trajectory, one has to find provisions in terms of memory to store the transition stream.

Goal of the Paper *To solve the control problem defined in Eq.* (15) *without having access to any generative model. Formally stated, given an infinitely long sample trajectory* $\{\mathbf{s}_0, \mathbf{a}_0, \mathbf{r}_0, \mathbf{s}_1, \mathbf{a}_1, \mathbf{r}_1, \mathbf{s}_2, \ldots\}$ *generated using the behaviour policy* π_{w_b} ($w_b \in \mathbb{R}^{k_2}$), *solve the control problem in* (15).

⊛ **Assumption (A4)** *The behaviour policy* π_{w_b}, *where* $w_b \in \mathbb{W}$, *satisfies the following condition*: $\pi_{w_b}(a|s) > 0, \forall s \in \mathbb{S}, \forall a \in \mathbb{A}$.

A few remarks are in order: We can classify the reinforcement learning algorithms based on the information made available to the algorithm in order to seek the optimal policy. We graphically illustrate this classification as a pyramid in Fig. 4. The bottom of the pyramid contains the classical methods, where the entire model information, i.e., both P and R are available, while in the middle, we have the model free algorithms, where both P and R are assumed hidden, however an access to the generative model/simulator is presumed. In the top of the pyramid, we have the single trajectory approaches, where a single sample trajectory generated using a behaviour policy is made available, however, the algorithms have no access to the model information or simulator. Observe that as one goes up the pyramid, the mass of the information vested upon the algorithm reduces considerably. The algorithm we propose in this paper belongs to the top of the information pyramid and to the upper half of the optimization box which makes it a unique combination.

3 Proposed algorithm

In this section, we propose an algorithm to solve the control problem defined in Eq. (15). We employ a stochastic approximation variant of the Gaussian based cross entropy method to find the optimal policy. We delay the discussion of the algorithm until the next subsection. We now focus on the objective function estimation. The objective function values $\mathbb{E}_{\nu_w}\left[L(h_{w|w})\right]$ which are required to efficiently guide the search for w^* are estimated using the off-policy LSTD(λ) method. In LFA, given $w \in \mathbb{W}$, the best approximation of V^w one can hope for is the projection $\Pi^w V^w$. Theorem 1 of Tsitsiklis and Roy (1997) shows that the on-policy LSTD(λ) solution $\Phi x_{w|w}$ is indeed an approximation of the projection $\Pi^w V^w$. Using Babylonian–Pythagorean theorem and Theorem 1 of Tsitsiklis and Roy (1997) along with a little arithmetic, we obtain $\|\Phi x_{w|w} - \Pi^w V^w\|_{\nu_w} \leq \frac{\sqrt{(1-\lambda)\gamma(\gamma+\gamma\lambda+2)}}{1-\gamma}\|\Pi^w V^w - V^w\|_{\nu_w}$. Hence for $\lambda = 1$, we have $\Phi x_{w|w} = \Pi^w V^w$, i.e., the on-policy LSTD(1) provides the exact projection. However for $\lambda < 1$, only approximations to it are obtained. Now when off-policy LSTD(λ) is applied, it adds one more level of approximation, i.e., $\Phi x_{w|w}$ is approximated by $\Phi x_{w|w_b}$. Hence to evaluate the performance of the off-policy approximation, we must quantify the errors incurred in the approximation procedure and we believe a capacious analysis had been far overdue.

3.1 Choice of the behaviour policy

The behaviour policy is often an exploration policy which promotes the exploration of the state and action spaces of the MDP. Efficient exploration is a precondition for effective learning. In this paper, we operate in a minimalistic MDP setting, where the only information available for inference is the single stream of transitions and payoffs generated using the behaviour policy. So the choice of the behaviour policy is vital for a sound inductive reasoning. The following theorem will provide a bound on the approximation error incurred in the off-policy LSTD(λ) method. The provided bound can be beneficial in choosing a good behaviour policy and also supplements in understanding the stability and usefulness of the proposed algorithm.

Theorem 2 *For a given $w \in \mathbb{W}$, the target policy vector, and $w_b \in \mathbb{W}$, the behaviour policy vector, let $x_{w|w}$ and $x_{w|w_b}$ be the solutions of the on-policy and off-policy versions of LSTD(λ), respectively, with $\lambda \in [0, 1]$.*

$$\textit{If} \sup_{s\in\mathbb{S}, a\in\mathbb{A}} \left|\frac{\pi_w(a|s)}{\pi_{w_b}(a|s)} - 1\right| < \epsilon_2, \ \textit{then} \ \frac{\left\|x_{w|w} - x_{w|w_b}\right\|_\infty}{\|x_{w|w}\|_\infty} \leq O\Big((|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)\frac{(1+\gamma)(1+\gamma\lambda)}{(1-\gamma)(1-\gamma\lambda)}\|D^{\nu_{w_b}}\|_\infty\|(D^{\nu_{w_b}})^{-1}\|_\infty\Big). \tag{16}$$

$$\textit{Also,} \qquad \|\Phi x_{w|w_b} - V^w\|_{\nu_{w_b}} \leq \tfrac{\gamma-2\gamma\lambda+1}{1-\gamma}\|V^w - V^{w_b}\|_{\nu_{w_b}} + \tfrac{\epsilon_2(1-\gamma\lambda)\|R\|_\infty}{(1-\gamma)^2} + \tfrac{1-\gamma\lambda}{1-\gamma}\|\Pi^{w_b}V^w - V^w\|_{\nu_{w_b}}, \tag{17}$$

where V^w and V^{w_b} are the true value functions corresponding to the SRPs π_w and π_{w_b} respectively. Also, ν_{w_b} is the stationary distribution of the Markov chain defined by P_{w_b} and $D^{\nu_{w_b}}$ is the diagonal matrix defined in Theorem 1.

Proof Given $w \in \mathbb{W}$, we have

$$P_w(s, s') = \sum_{a \in \mathbb{A}} \pi_w(a|s)P(s, a, s'), s, s' \in \mathbb{S},$$
$$P_{w_b}(s, s') = \sum_{a \in \mathbb{A}} \pi_{w_b}(a|s)P(s, a, s'), s, s' \in \mathbb{S}.$$

Therefore,

$$P_w = P_{w_b} + F, \text{ where } F = P_w - P_{w_b}.$$

Hence, for $s, s' \in \mathbb{S}$,

$$\begin{aligned} |F(s, s')| &= \Big|\sum_{a \in \mathbb{A}} \big(\pi_w(a|s) - \pi_{w_b}(a|s)\big) P(s, a, s')\Big|, \\ &= \Big|\sum_{a \in \mathbb{A}} \left(\frac{\pi_w(a|s)}{\pi_{w_b}(a|s)} - 1\right) \pi_{w_b}(a|s)P(s, a, s')\Big|, \\ &\leq \sum_{a \in \mathbb{A}} \epsilon_2 \pi_{w_b}(a|s)P(s, a, s'), \\ &= \epsilon_2 P_{w_b}(s, s'). \end{aligned} \tag{18}$$

The above bound of the deviation matrix F in terms of P_{w_b} compels us to apply the result from Xue (1997), which provides a sensitivity analysis of the stationary distribution of a Markov chain w.r.t. its probability transition matrix. In particular, by appealing to Theorem 1 of Xue (1997) along with Eq. (18), we obtain the following:

$$\begin{aligned} &\left|\frac{\nu_w(s) - \nu_{w_b}(s)}{\nu_{w_b}(s)}\right| \leq 2(|\mathbb{S}| - 1)\epsilon_2 + O(\epsilon_2^2), s \in \mathbb{S}. \\ &\implies \left|\frac{\nu_w(s) - \nu_{w_b}(s)}{\nu_{w_b}(s)}\right| \leq O(|\mathbb{S}|\epsilon_2), s \in \mathbb{S}. \end{aligned} \tag{19}$$

Let $\epsilon_3 = O(|\mathbb{S}|\epsilon_2)$. Then from (19), we get

$$\begin{aligned} &|\nu_w(s) - \nu_{w_b}(s)| \leq \epsilon_3|\nu_{w_b}(s)| \leq \epsilon_3(|\nu_w(s) - \nu_{w_b}(s)| + |\nu_w(s)|) \\ \implies &\frac{|\nu_w(s) - \nu_{w_b}(s)|}{|\nu_w(s)|} \leq \frac{\epsilon_3}{1 - \epsilon_3} = O(\epsilon_3 + \epsilon_3^2) = O(|\mathbb{S}|\epsilon_2 + |\mathbb{S}|^2\epsilon_2^2). \end{aligned} \tag{20}$$

For the policy π_w, recall that the on-policy approximation is $\Phi x_{w|w}$, where $x_{w|w}$ is the unique solution to the linear system $A_{w|w}x = b_{w|w}$. Analogously, the off-policy approximation is given by $\Phi x_{w|w_b}$, where $x_{w|w_b}$ is the unique solution to the linear system $A_{w|w_b}x = b_{w|w_b}$. Now using the bound in (20) and the definitions of $A_{w|w}$, $A_{w|w_b}$, $b_{w|w}$ and $b_{w|w_b}$ in (14) and (13), it is easy to verify that

$$|A_{w|w_b} - A_{w|w}| \leq O(|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)|A_{w|w}| \text{ and}$$

$$|b_{w|w_b} - b_{w|w}| \leq O(|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)|b_{w|w}|.$$

Hence the off-policy linear system $A_{w|w_b}x = b_{w|w_b}$ can be viewed as a perturbed version of the on-policy system $A_{w|w}x = b_{w|w}$. Let $\epsilon_4 = O(|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)$. Now we make use of the norm bound on the solutions of perturbed linear system of equations provided in Theorem 2.2 of Higham (1994). In particular, using the remark following Theorem 2.2 of Higham (1994), we have

$$\frac{\left\|x_{w|w} - x_{w|w_b}\right\|_\infty}{\|x_{w|w}\|_\infty} \leq \frac{2\epsilon_4\kappa(A_{w|w})}{1 - \epsilon_4\kappa(A_{w|w})}, \tag{21}$$

where $\kappa(A_{w|w}) = \|A_{w|w}\|_\infty\|A_{w|w}^{-1}\|_\infty$ (condition number $\kappa(\cdot)$ is defined in Sect. 1). Using the definition of $A_{w|w}$ in (14), we obtain $A_{w|w}^{-1} = \Phi^{-1}(I-\gamma P_w)^{-1}(I-\gamma\lambda P_w)(D^{\nu_w})^{-1}\Phi^{-\top}$, where Φ^{-1} is the left inverse of Φ and $\Phi^{-\top}$ is the right inverse of $\Phi^\top$. Therefore $\|A_{w|w}^{-1}\|_\infty \leq \|\Phi^{-1}\|_\infty\|(I-\gamma P_w)^{-1}\|_\infty\|I-\gamma\lambda P_w\|_\infty\|(D^{\nu_w})^{-1}\|_\infty\|\Phi^{-\top}\|_\infty$. Now by arguing along the same lines as (31), one can show that $\|(I-\gamma P_w)^{-1}\|_\infty \leq \frac{1}{1-\gamma}$. Also $\|I-\gamma\lambda P_w\|_\infty = 1+\gamma\lambda$. And the feature matrix Φ is presumed to be constant. A forteriori, $\|A_{w|w}^{-1}\|_\infty = O(\frac{1+\gamma\lambda}{1-\gamma}\|(D^{\nu_w})^{-1}\|_\infty)$. Also from (19), we have $\nu_w(s) \geq (1-\epsilon_3)\nu_{w_b}(s), s \in \mathbb{S}$. Henceforth, $\|A_{w|w}^{-1}\|_\infty = O(\frac{1+\gamma\lambda}{(1-\gamma)(1-\epsilon_3)}\|(D^{\nu_{w_b}})^{-1}\|_\infty)$. Similarly, one can show that

$$\|A_{w|w}\|_\infty = O(\frac{(1+\gamma)(1+\epsilon_3)}{(1-\gamma\lambda)}\|D^{\nu_{w_b}}\|_\infty).$$

Hence

$$\begin{aligned}\kappa(A_{w|w}) &= O\big(\frac{(1+\epsilon_3)(1+\gamma)(1+\gamma\lambda)}{(1-\gamma)(1-\gamma\lambda)(1-\epsilon_3)}\|D^{\nu_{w_b}}\|_\infty\|(D^{\nu_{w_b}})^{-1}\|_\infty\big),\\ &= O\big(\frac{(1+\epsilon_3)^2(1+\gamma)(1+\gamma\lambda)}{(1-\gamma)(1-\gamma\lambda)}\|D^{\nu_{w_b}}\|_\infty\|(D^{\nu_{w_b}})^{-1}\|_\infty\big).\end{aligned}$$

Consequently from (21), we get

$$\begin{aligned}\frac{\left\|x_{w|w} - x_{w|w_b}\right\|_\infty}{\|x_{w|w}\|_\infty} &\leq O(\epsilon_4\kappa(A_{w|w}) + \epsilon_4^2\kappa^2(A_{w|w}))\\ &= O\ \big((|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)\tfrac{(1+\gamma)(1+\gamma\lambda)}{(1-\gamma)(1-\gamma\lambda)}\|D^{\nu_{w_b}}\|_\infty\|(D^{\nu_{w_b}})^{-1}\|_\infty\big).\end{aligned}$$

This completes the proof of (16). □

Now to prove (17), here we define an operator $T_{w|w_b}^{(\lambda)}$ [referred to as the TD(λ] operator in Tsitsiklis and Roy (1997)) as follows:

$$T_{w|w_b}^{(\lambda)}V = (1-\lambda)\sum_{i=0}^{\infty}\lambda^i\left(\sum_{j=0}^{i}(\gamma P_{w_b})^j R^w(s_j) + (\gamma P_{w_b})^{i+1}V\right) \tag{22}$$

$$\text{with } P_{w_b}(s,s') \triangleq \sum_{a\in\mathbb{A}}\pi_{w_b}(a|s)P(s,a,s')$$

$$\text{and } R^w(s) \triangleq \sum_{s'\in\mathbb{S}}\sum_{a\in\mathbb{A}}\pi_w(a|s)P(s,a,s')R(s,a,s'). \tag{23}$$

Before we proceed any further, a few observations are in order:

Observation 1 For $V \in \mathbb{R}^{|\mathbb{S}|}$ and $w \in \mathbb{W}$, we have

$$\|\Pi^w V\|_{\nu_w} \leq \|V\|_{\nu_w}. \tag{24}$$

Proof Using$< \Pi^w V - V, \Pi^w V >_{\nu_w} = 0$ and by the Babylonian–Pythagorean theorem, we have $\|V\|^2_{\nu_w} = \|\Pi^w V - V\|^2_{\nu_w} + \|\Pi^w V\|^2_{\nu_w}, \Rightarrow \|\Pi^w V\|_{\nu_w} \leq \|V\|_{\nu_w}$. This proves (24). □

Observation 2 For $w \in \mathbb{W}, s \in \mathbb{S}$,

$$\text{if } \sup_{a \in \mathbb{S}} \left| \frac{\pi_w(a|s)}{\pi_{w_b}(a|s)} - 1 \right| < \epsilon_2 \text{ then } |R^w(s) - R^{w_b}(s)| \leq \epsilon_2 \|R\|_\infty. \tag{25}$$

Proof From (23), we have,

$$\begin{aligned} |R^w(s) - R^{w_b}(s)| &= \Big| \sum_{s' \in \mathbb{S}} \sum_{a \in \mathbb{A}} \big(\pi_w(a|s) - \pi_{w_b}(a|s)\big) P(s,a,s') R(s,a,s') \Big|, \\ &\leq \sum_{s' \in \mathbb{S}} \sum_{a \in \mathbb{A}} \big|\pi_w(a|s) - \pi_{w_b}(a|s))\big| P(s,a,s') R(s,a,s'), \\ &\leq \sum_{s' \in \mathbb{S}} \epsilon_2 P_{w_b}(s,s') \|R\|_\infty, \\ &\leq \epsilon_2 \|R\|_\infty. \end{aligned} \tag{26}$$

This proves (25). □

Observation 3 For $V_1, V_2 \in \mathbb{R}^{|\mathbb{S}|}$,

$$\|T^{(\lambda)}_{w|w_b} V_1 - T^{(\lambda)}_{w|w_b} V_2\|_{\nu_{w_b}} \leq \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|V_1 - V_2\|_{\nu_{w_b}}.$$

Proof Refer Lemma 4 of Tsitsiklis and Roy (1997).

Observation 4

$$\big|T^{(\lambda)}_{w|w_b} V(s) - T^{(\lambda)}_{w_b|w_b} V(s)\big| \leq \frac{\epsilon_2 \|R\|_\infty}{1-\gamma}. \tag{27}$$

Proof From (22) and observation 2, we have

$$\begin{aligned} &\big|T^{(\lambda)}_{w|w_b} V(s) - T^{(\lambda)}_{w_b|w_b} V(s)\big| \\ &= \Big|(1-\lambda) \sum_{i=0}^{\infty} \lambda^i \sum_{j=0}^{i} \gamma^j \sum_{s' \in \mathbb{S}} P^j_{w_b}(s, s'\Big(R^w(s') - R^{w_b}(s')\Big)\Big|, \\ &\leq (1-\lambda) \sum_{i=0}^{\infty} \lambda^i \sum_{j=0}^{i} \gamma^j \sum_{s' \in \mathbb{S}} P^j_{w_b}(s,s') \|R\|_\infty \epsilon_2, \\ &= (1-\lambda) \sum_{i=0}^{\infty} \lambda^i \sum_{j=0}^{i} \gamma^j \epsilon_2 \|R\|_\infty, \\ &\leq \frac{\epsilon_2 \|R\|_\infty}{1-\gamma}. \end{aligned}$$

This proves (27).

Observation 5 $\Phi x_{w|w_b} = \Pi^{w_b} T^{(\lambda)}_{w|w_b} \Phi x_{w|w_b}$. This is the *off-policy projected Bellman equation*. Detailed discussion is available in Yu (2012). For the on-policy case, similar equation exists which is as follows: $\Phi x_{w|w} = \Pi^{w} T^{(\lambda)}_{w|w} \Phi x_{w|w}$. For the proof of the above equation, refer Theorem 1 of Tsitsiklis and Roy (1997). A few other relevant fixed point equations are $T^{(\lambda)}_{w|w} V^w = V^w$ and $T^{(\lambda)}_{w_b|w_b} V^{w_b} = V^{w_b}$. The proof of the above equations is provided in Lemma 5 of Tsitsiklis and Roy (1997).

This completes the observations. Now we will prove (17). Using the triangle inequality and the above observations, we have

$$\begin{aligned}
&\|\Phi x_{w|w_b} - V^w\|_{\nu_{w_b}} \leq \|\Phi x_{w|w_b} - \Pi^{w_b} V^{w_b}\|_{\nu_{w_b}} + \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&=_1 \|\Pi^{w_b} T^{(\lambda)}_{w|w_b} \Phi x_{w|w_b} - \Pi^{w_b} T^{(\lambda)}_{w_b|w_b} V^{w_b}\|_{\nu_{w_b}} + \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&\leq_2 \|T^{(\lambda)}_{w|w_b} \Phi x_{w|w_b} - T^{(\lambda)}_{w_b|w_b} V^{w_b}\|_{\nu_{w_b}} + \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&\leq_3 \|T^{(\lambda)}_{w|w_b} \Phi x_{w|w_b} - T^{(\lambda)}_{w|w_b} V^{w_b}\|_{\nu_{w_b}} + \|T^{(\lambda)}_{w|w_b} V^{w_b} - T^{(\lambda)}_{w_b|w_b} V^{w_b}\|_{\nu_{w_b}} +\\
&\|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&\leq_4 \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|\Phi x_{w|w_b} - V^{w_b}\|_{\nu_{w_b}} + \|T^{(\lambda)}_{w|w_b} V^{w_b} - T^{(\lambda)}_{w_b|w_b} V^{w_b}\|_{\nu_{w_b}}\\
&+ \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&\leq_5 \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|\Phi x_{w|w_b} - V^{w}\|_{\nu_{w_b}} + \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|V^w - V^{w_b}\|_{\nu_{w_b}} +\\
&\frac{\epsilon_2 \|R\|_\infty}{1-\gamma} + \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},
\end{aligned}$$

Note that $=_1$ follows from Observation 5; $\leq_2$ follows from Observation 1; $\leq_3$ follows from the triangle inequality; $\leq_4$ follows from Observation 3; $\leq_5$ follows from Observation 4 and the triangle inequality. This further implies

$$\begin{aligned}
&\frac{1-\gamma}{1-\gamma\lambda} \|\Phi x_{w|w_b} - V^w\|_{\nu_{w_b}}\\
&\quad \leq \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|V^w - V^{w_b}\|_{\nu_{w_b}} + \frac{\epsilon_2 \|R\|_\infty}{1-\gamma} + \|\Pi^{w_b} V^{w_b} - V^w\|_{\nu_{w_b}},\\
&\quad \leq \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|V^w - V^{w_b}\|_{\nu_{w_b}} + \frac{\epsilon_2 \|R\|_\infty}{1-\gamma} + \|\Pi^{w_b} V^{w_b} - \Pi^{w_b} V^w\|_{\nu_{w_b}} +\\
&\|\Pi^{w_b} V^w - V^w\|_{\nu_{w_b}},\\
&\quad \leq \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|V^w - V^{w_b}\|_{\nu_{w_b}} + \frac{\epsilon_2 \|R\|_\infty}{1-\gamma} + \|V^{w_b} - V^w\|_{\nu_{w_b}}\\
&\qquad + \|\Pi^{w_b} V^w - V^w\|_{\nu_{w_b}}.
\end{aligned}$$

Therefore

$$\|\Phi x_{w|w_b} - V^w\|_{\nu_{w_b}} \leq \frac{\gamma - 2\gamma\lambda + 1}{1-\gamma} \|V^w - V^{w_b}\|_{\nu_{w_b}} + \frac{\epsilon_2 (1-\gamma\lambda) \|R\|_\infty}{(1-\gamma)^2} + \frac{1-\gamma\lambda}{1-\gamma} \|\Pi^{w_b} V^w - V^w\|_{\nu_{w_b}}.$$

This completes the proof of (17). □

The implications of the bounds given in Theorem 2 are indeed significant. The quantity $\sup_{s\in\mathbb{S},a\in\mathbb{A}}\left|\frac{\pi_w(a|s)}{\pi_{w_b}(a|s)}-1\right|$ given in the hypothesis of the theorem can ostensibly be viewed as a measure of the closeness of the SRPs π_w and π_{w_b}, with the minimum value of 0 being achieved in the on-policy case. Under the hypothesis that $\sup_{s\in\mathbb{S},a\in\mathbb{A}}\left|\frac{\pi_w(a|s)}{\pi_{w_b}(a|s)}-1\right| < \epsilon_2$, we obtain in (16) an upper bound on the relative error of the on-policy and off-policy solutions. The bound is predominantly dominated by the hypothesis bound ϵ_2, the eligibility factor λ, the discount factor γ and $\|(D^{\nu_{w_b}})^{-1}\|_\infty\|D^{\nu_{w_b}}\|_\infty$. Note that $\|D^{\nu_{w_b}}\|_\infty = \max_s \nu_{w_b}(s)$ and $\|(D^{\nu_{w_b}})^{-1}\|_\infty = (\min_s \nu_{w_b}(s))^{-1}$. If the behaviour policy is chosen in such a way that all the states are equally likely under its stationary distribution, then $\|(D^{\nu_{w_b}})^{-1}\|_\infty\|D^{\nu_{w_b}}\|_\infty \approx 1$. Consequently, the upper bound can be reduced to $O\big((|\mathbb{S}|^2\epsilon_2^2 + |\mathbb{S}|\epsilon_2)\frac{(1+\gamma)(1+\gamma\lambda)}{(1-\gamma)(1-\gamma\lambda)}\big)$.

Now regarding the latter bound provided in Eq. (17), given $w \in \mathbb{W}$, by using triangle inequality and Eq. (17), we obtain a proper quantification of the distance between the solution of the off-policy LSTD(λ), i.e., $\Phi x_{w|w_b}$ and the projection $\Pi^{w_b}V^w$ in terms of $\|\cdot\|_{\nu_{w_b}}$ and ϵ_2. The above bound can be further improved by obtaining an expedient bound for $\|V^w - V^{w_b}\|_{\nu_{w_b}}$ as follows:

Corollary 1 *Let $w \in \mathbb{W}$, $\lambda \in [0,1]$ and $\gamma \in (0,1)$. Let the assumptions of Theorem 2 hold. Also, assume that ϵ_2 which is defined in Theorem 2 satisfy $\epsilon_2\frac{1+\gamma}{1-\gamma} < 1$. Then $\exists K_1 > 0$, s.t.*

$$\|\Phi x_{w|w_b} - V^w\|_{\nu_{w_b}} \le \frac{K_1(\gamma - 2\gamma\lambda + 1)(1+\gamma)\epsilon_2}{(1-\gamma)(1-\gamma-\epsilon_2(1+\gamma))} + \frac{\epsilon_2(1-\gamma\lambda)\|R\|_\infty}{(1-\gamma)^2} +$$
$$\frac{1-\gamma\lambda}{1-\gamma}\|\Pi^{w_b}V^w - V^w\|_{\nu_{w_b}},$$

Proof Given $w \in \mathbb{W}$, the value function V^w satisfies the linear system given by the Bellman equation as shown in Eq. (4), i.e.,

$$(I - \gamma P_w)V^w = R^w. \tag{28}$$

Similarly, for the behaviour policy w_b, we have

$$(I - \gamma P_{w_b})V^{w_b} = R^{w_b}. \tag{29}$$

Now, note that

$$(I - \gamma P_w) = (I - \gamma P_{w_b}) + F, \text{ where } F = \gamma(P_{w_b} - P_w).$$
$$R^w = R^{w_b} + b, \text{ where } b = R^w - R^{w_b}.$$

By arguing along the same lines as (26), one can show that $|b(s)| \le \epsilon_2|R^{w_b}(s)|$, $\forall s \in \mathbb{S}$. Similarly, $|F(s,s')| \le \epsilon_2\gamma|P_{w_b}(s,s')| \le \epsilon_2|(I-\gamma P_{w_b})(s,s')|$, $\forall s, s' \in \mathbb{S}$. [The proof is similar to that of (18)]. Hence the on-policy linear system given by (28) can be viewed as a perturbed version of the linear system (29) of the behaviour policy. So, using the remark following Theorem 2.2 of Higham (1994), we obtain the following:

$$\frac{\|V^w - V^{w_b}\|_{\nu_{w_b}}}{\|V^{w_b}\|_{\nu_{w_b}}} \le \frac{2\epsilon_2\kappa(I-\gamma P_{w_b})}{1-\epsilon_2\kappa(I-\gamma P_{w_b})}. \tag{30}$$

where $\kappa(I-\gamma P_{w_b}) = \|I-\gamma P_{w_b}\|_\infty\|(I-\gamma P_{w_b})^{-1}\|_\infty$ (condition number $\kappa(\cdot)$ is defined in Sect. 1). It is also easy to verify that $\|I-\gamma P_{w_b}\|_\infty = 1+\gamma$. Now to bound $\|(I-\gamma P_{w_b})^{-1}\|_\infty$, we use the Ahlberg–Nilson–Varah bound from Varga (1976). In particular, by using Theorem A of Varga (1976), we have

$$\|(I-\gamma P_{w_b})^{-1}\|_\infty \leq \frac{1}{\min_{1\leq i\leq|\mathbb{S}|}\left\{|(I-\gamma P_{w_b})_{ii}|-\sum_{j=1,j\neq i}^{|\mathbb{S}|}|(I-\gamma P_{w_b})_{ij}|\right\}},$$
$$= \tfrac{1}{1-\gamma}, \tag{31}$$

where $(\cdot)_{ij}$ is the (i, j) entry of the matrix.

By putting together the above facts, we get $\kappa(I-\gamma P_{w_b}) \leq \frac{1+\gamma}{1-\gamma}$. Consequently from Eq. (30) and the assumption that $\epsilon_2\frac{1+\gamma}{1-\gamma} < 1$, we obtain

$$\frac{\|V^w - V^{w_b}\|_{\nu_{w_b}}}{\|V^{w_b}\|_{\nu_{w_b}}} \leq \frac{2\epsilon_2(1+\gamma)}{1-\gamma-\epsilon_2(1+\gamma)}.$$

Therefore $\|V^w - V^{w_b}\|_{\nu_{w_b}} \leq K_1\epsilon_2(1+\gamma)(1-\gamma-\epsilon_2(1+\gamma))^{-1}$, $K_1 > 0$. The corollary now easily follows from the above bound and from (17) of Theorem 2. □

The note worthy result on the upper bound of the approximation error of the on-policy LSTD(λ) provided in Tsitsiklis and Roy (1997) can be easily derived from the above result as follows:

Corollary 2 *For $w \in \mathbb{W}$, $\lambda \in [0, 1]$ and $\gamma \in (0, 1)$,*

$$\|\Phi x_{w|w} - V^w\|_{\nu_w} \leq \frac{1-\gamma\lambda}{1-\gamma}\|\Pi^w V^w - V^w\|_{\nu_w}.$$

Proof In the on-policy case, $w_b = w$. Hence $\epsilon_2 = 0$. The corollary directly follows from direct substitution of these values in (17). □

3.2 Estimation of the objective function

The objective function of the control problem defined in Eq. (15) is

$$J(w) = \mathbb{E}_{\nu_w}\left[L(h_{w|w})\right]. \tag{32}$$

In this paper, we employ off-policy LSTD(λ) to approximate $h_{w|w}$ for a given policy parameter $w \in \mathbb{W}$. A sample trajectory $\{\mathbf{s}_0, \mathbf{a}_0, \mathbf{r}_0, \mathbf{s}_1, \mathbf{a}_1, \mathbf{r}_1, \mathbf{s}_2, \ldots\}$ (fixed for the algorithm) generated using the behaviour policy π_{w_b} is provided.

The procedure to estimate the objective function J is formally defined in Algorithm 1. The *Predict* procedure in Algorithm 1 is almost the same as the off-policy LSTD algorithm. The additional recursion (step 10) estimates the objective function defined in Eq. (32) as follows:

$$\ell^w_{k+1} = \ell^w_k + \alpha_{k+1}\Big(L(\mathbf{x}_k^\top\phi(\mathbf{s}_{k+1})) - \ell^w_k\Big), \tag{33}$$

where $\alpha_k = 1/k$. The above choice of α_k is merely a recommendation and not a strict requirement. This, however, alleviates the extra burden of deciding α_k during implementation.

For a given $w \in \mathbb{W}$, ℓ^w_k attempts to find an approximate value of the objective function $J(w)$. The following lemma formally characterizes the limiting behaviour of the iterates ℓ^w_k.

Lemma 1 *For a given $w \in \mathbb{W}$,*

$$\ell^w_k \to \ell^w_* = \mathbb{E}_{\nu_{w_b}}\left[L(x^\top_{w|w_b}\phi(\mathbf{s}))\right] \text{ as } k \to \infty \text{ w.p. 1.} \tag{34}$$

Proof We begin the proof by defining the filtration $\{\mathcal{F}_k\}_{k\in\mathbb{N}}$, where the σ-field $\mathcal{F}_k \triangleq \sigma(\{\mathbf{x}_i, \ell^w_i, \mathbf{s}_i, \mathbf{a}_i, \mathbf{r}_i, 0 \leq i \leq k\})$.

Now recalling the recursion (33),

$$\begin{aligned}\ell^w_{k+1} &:= \ell^w_k + \alpha_{k+1}\Big(L(\mathbf{x}_k^\top \phi(\mathbf{s}_{k+1})) - \ell^w_k\Big)\\ &:= \ell^w_k + \alpha_{k+1}\Big(h(\ell^w_k) + \mathbb{M}_{k+1} + c_k\Big),\end{aligned}$$

where $\mathbb{M}_{k+1} \triangleq L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1})) - \mathbb{E}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1}))\big|\mathcal{F}_k\Big]$, $h(z) \triangleq \mathbb{E}_{\nu_{w_b}}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1}))\Big] - z$ and $c_k \triangleq L(\mathbf{x}_k^\top \phi(\mathbf{s}_{k+1})) - L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1}))+$ $\mathbb{E}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1}))\big|\mathcal{F}_k\Big] - \mathbb{E}_{\nu_{w_b}}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}_{k+1}))\Big]$.

We state here a few observations:

1. $\{\mathbb{M}_k, k \geq 1\}$ is a martingale difference noise sequence w.r.t. $\{\mathcal{F}_k\}$, i.e., $\mathbb{M}_k$ is $\mathcal{F}_k$-measurable, integrable and $\mathbb{E}[\mathbb{M}_{k+1}|\mathcal{F}_k] = 0$ *a.s.*, $\forall k \geq 0$.
2. $h(\cdot)$ is a Lipschitz continuous function.
3. $\exists K > 0$ *s.t.* $\mathbb{E}[|\mathbb{M}_{k+1}|^2|\mathcal{F}_k] \leq K(1 + |\ell_k|^2)$ *a.s.*, $\forall k \geq 0$.
4. By Theorem 1, $c_k \to 0$ as $k \to \infty$ *w.p. 1*. This directly follows by considering the following facts: (a) by Eq. (1), the off-policy LSTD(λ) iterates $\{\mathbf{x}_k\}$ converges almost surely to the off-policy solution $x_{w|w_b}$ (b) by assumption (A2), $P_{w_b}(\mathbf{s}_k = s) \to \nu_{w_b}(s)$ as $k \to \infty$ and (c) $L(\cdot)$ and $\phi(\cdot)$ are bounded.
5. For a given $w \in \mathbb{W}$, the iterates $\{\ell^w_k\}_{k\in\mathbb{N}}$ are stable, i.e., $\sup_k |\ell^w_k| < \infty$ *a.s.* A brief proof is provided here: For $c > 0$, we define

   $$h_c(z) \triangleq \frac{h(cz)}{c} = \frac{\mathbb{E}_{\nu_{w_b}}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}))\Big]}{c} - z. \tag{35}$$

   Now consider the following ODE corresponding to the following ∞-system:

   $$\dot{z}(t) = h_\infty(z(t)) \triangleq \lim_{c\to\infty} h_c(z(t)). \tag{36}$$

   Note that $h_\infty(z) = -z$. It can be easily verified that the above ODE is globally asymptotically stable to the origin. This further implies the stability of the iterates $\{\ell^w_k\}$ using Theorem 2, Chapter 3 of Borkar (2008).

Now by appealing to the third extension of Theorem 2, Section 2.2, Chapter 2 of Borkar (2008) and from the above observations, we can henceforth conclude almost surely that the iterates $\{\ell^w_k\}$ asymptotically track the ODE given by:

$$\dot{z}(t) = h(z(t)). \tag{37}$$

This further implies that the limit points of the iterates $\{\ell^w_k\}$ are indeed contained in the limit set of the ODE (37) almost surely. However, it is easy to verify that $\mathbb{E}_{\nu_{w_b}}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}))\Big]$ is the unique globally asymptotically stable equilibrium of the ODE (37). Hence $\lim_{k\to\infty} \ell^w_k =$ $\mathbb{E}_{\nu_{w_b}}\Big[L(x_{w|w_b}^\top \phi(\mathbf{s}))\Big]$ *a.s.* This completes the proof of (34). □

Remark 1 By the above lemma, for a given $w \in \mathbb{W}$, the quantity ℓ^w_k tracks $\mathbb{E}_{\nu_{w_b}}\big[L(x_{w|w_b}^\top$ $\phi(\mathbf{s}))\big]$. This is however different from the true objective function value $J(w) =$ $\mathbb{E}_{\nu_w}\big[L(h_{w|w})\big]$, when $w \neq w_b$. This additional approximation error incurred is the extra cost one has to pay for the dearth in information (in the form of generative model) about the underlying MDP. Nevertheless, from Eqs. (16) and (19), we know that the relative errors in the solutions $x_{w|w}$ and $x_{w|w_b}$ as well as in the stationary distributions ν_w and ν_{w_b} are bounded. We also know that $\Phi x_{w|w} \approx h_{w|w}$. Further, if we can restrict the smoothness of the performance function L, then we can contain the deviation of $L(y)$ when the input variable y is perturbed slightly. All these factors further affirm the fact that the approximation proposed in (33) is well-conditioned. This is indeed significant, considering the restricted setting we operate in, i.e., non-availability of the generative model.

Algorithm 1 Predict Procedure

```
Input parameters: $w \in \mathbb{W}, N \in \mathbb{N}$   ▶ Input policy vector, Trajectory length
Data: A priori chosen sample trajectory $\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, \ldots\}$ generated using the behaviour policy $\pi_{w_b}$
function PREDICT($w$, $N$)
    $k := 0$;   ▶ Iteration count initialized to 0
    while $k < N$ do
        $\mathbf{e}_{k+1} := \gamma\lambda\rho_k\mathbf{e}_k + \phi(s_k)$;▶ The sampling ratio $\rho_k = \frac{\pi_w(a_k|s_k)}{\pi_{w_b}(a_k|s_k)}$
        $\mathbf{A}_{k+1} := \mathbf{A}_k + \frac{1}{k+1}\left(\mathbf{e}_k(\phi(s_k) - \gamma\rho_k\phi(s_{k+1}))^\top - \mathbf{A}_k\right)$;
        $\mathbf{b}_{k+1} := \mathbf{b}_k + \frac{1}{k+1}(\rho_k r_k \mathbf{e}_k - \mathbf{b}_k)$;
        $\mathbf{x}_{k+1} := \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}$;   ▶ Prediction vector
        $\ell_{k+1}^w := \ell_k^w + \alpha_{k+1}\left(L(\mathbf{x}_k^\top\phi(\mathbf{s}_{k+1})) - \ell_k^w\right)$;   ▶ Objective func estimation
        $k := k+1$;
    end while
    return $\ell_N^w$;   ▶ Outputs after N iterations
end function
```

3.3 Stochastic approximation version of Gaussian cross entropy method and its application to the control problem

Cross entropy method (Rubinstein and Kroese 2013; Kroese et al. 2006) solves optimization problems where the objective function does not possess good structural properties, such as possibly discontinuous, non-differentiable, i.e., those of the kind:

$$\text{Find } x^* \in \underset{x \in \mathbb{X} \subset \mathbb{R}^d}{\arg\max}\, J(x), \tag{38}$$

where $J : \mathbb{X} \to \mathbb{R}$ is a bounded Borel measurable function.

CE is a *model based search method* (Zlochin et al. 2004) used to solve the global optimization problem. CE is a zero-order method (*a.k.a.* gradient-free method) which implies the algorithm does not require gradient or higher-order derivatives of the objective function. This remarkable feature of the algorithm makes it a suitable choice for the “black-box” optimization setting, where neither a closed form expression nor structural properties of the objective function J are available. CE method has found successful application in diverse domains which include continuous multi-extremal optimization (Rubinstein 1999), buffer allocation (Alon et al. 2005), queueing models (de Boer 2000), DNA sequence alignment (Keith and Kroese 2002), control and navigation (Helvik and Wittner 2001), reinforcement learning (Mannor et al. 2003; Menache et al. 2005) and several NP-hard problems (Rubinstein 2002, 1999). We would also like to mention that there are other model based search methods in the literature, a few pertinent ones include the gradient-based adaptive stochastic search for simulation optimization (GASSO) (Zhou et al. 2014), estimation of distribution algorithm (EDA) (Mühlenbein and Paass 1996) and model reference adaptive search (MRAS) (Hu et al. 2007). However, in this paper, we do not explore the possibility of employing the above algorithms in a MDP setting.

The Gaussian based cross entropy method generates a sequence of Gaussian distributions $\{\theta_j = (\mu_j, \Sigma_j)^\top \in \Theta \subset \mathbb{R}^{d(d+1)}\}_{j \in \mathbb{N}}$ parametrized by its mean vector $\mu_j \in \mathbb{R}^d$ and the covariance matrix $\Sigma_j \in \mathbb{R}^{d \times d}$, with the property that the support of the multivariate Gaussian probability density function given by

$$f_{\theta_{j+1}}(x) = (2\pi|\Sigma_{j+1}|)^{-d/2} \exp(-\frac{1}{2}(x-\mu_{j+1})^\top \Sigma_{j+1}^{-1}(x-\mu_{j+1}))$$

satisfies (P1) below.

Property (P1) *supp*$(f_{\theta_{j+1}}) \subseteq \{x | J(x) \geq \gamma_\rho(J, \theta_j)\}$,

where $\rho \in (0, 1)$ is fixed *a priori*. Note that $\gamma_\rho(J, \theta_j)$ is the $(1-\rho)$-quantile of J w.r.t. the distribution f_{θ_j}. Hence it is easy to verify that the *threshold sequence* $\{\gamma_\rho(J, \theta_j)\}_{j \in \mathbb{N}}$ is a monotonically non-decreasing sequence. The intuition behind this recursive generation of the model sequence is that by assigning greater weight to the higher values of J at each iteration, the expected behaviour of the model sequence should improve. We make the following assumption on the model parameter space Θ:

⊛ **Assumption (A5)** *The parameter space* Θ *is a compact subset of* $\mathbb{R}^{d(d+1)}$.

The invariant in each iteration of the CE method is property (P1). At each instant $j+1$, the CE method seeks the distribution which is proximally optimal to maintaining the invariant by solving the following optimization problem:

$$\theta_{j+1} = \underset{\theta \in \Theta}{\arg\max}\, \Gamma_j(\theta, \gamma_\rho(J, \theta_j)), \tag{39}$$

where $\Gamma_j(\theta, \gamma) \triangleq \mathbb{E}_{\theta_j}\left[\varphi(J(\mathbf{x}))I_{\{J(\mathbf{x}) \geq \gamma\}} \log f_\theta(\mathbf{x})\right]$ and $\varphi : \mathbb{R} \to \mathbb{R}_+$ is a positive, strictly monotonically increasing function. This recursive equation forms the basis of the cross entropy method and is referred to as the *model update procedure*.

Note that the solution to Eq. (39) is obtained by equating $\nabla\Gamma_j$ to 0:

$$\nabla_{\vartheta_1^\theta}\Gamma_j(\theta, \gamma) = 0 \Rightarrow \mu = \frac{\mathbb{E}_{\theta_j}\left[\mathbf{g_1}\big(J(\mathbf{x}), \mathbf{x}, \gamma\big)\right]}{\mathbb{E}_{\theta_j}\left[\mathbf{g}_0(J(\mathbf{x}), \gamma)\right]} \triangleq \Upsilon_1(\theta_j, \gamma), \tag{40}$$

$$\nabla_{\vartheta_2^\theta}\Gamma_j(\theta, \gamma) = 0 \Rightarrow \Sigma = \frac{\mathbb{E}_{\theta_j}\left[\mathbf{g_2}\big(J(\mathbf{x}), \mathbf{x}, \gamma, \mu\big)\right]}{\mathbb{E}_{\theta_j}\left[\mathbf{g_0}\big(J(\mathbf{x}), \gamma\big)\right]} \triangleq \Upsilon_2(\theta_j, \gamma), \tag{41}$$

$$\text{where} \quad \mathbf{g_0}(y, \gamma) \triangleq \varphi(y)I_{\{y \geq \gamma\}}, \tag{42a}$$

$$\mathbf{g_1}(y, x, \gamma) \triangleq x\varphi(y)I_{\{y \geq \gamma\}}, \tag{42b}$$

$$\mathbf{g_2}(y, x, \gamma, \mu) \triangleq \varphi(y)(x-\mu)(x-\mu)^\top I_{\{y \geq \gamma\}} \tag{42c}$$

$$(\vartheta_1^\theta, \vartheta_2^\theta)^\top = (\Sigma^{-1}\mu, -\tfrac{1}{2}\Sigma^{-1})^\top. \tag{42d}$$

The mapping of $(\mu, \Sigma)^\top \mapsto (\Sigma^{-1}\mu, \frac{-1}{2}\Sigma^{-1})^\top$ is a bijective transformation and it makes the algebra a lot simpler. Also it is not hard to verify that Υ_1 and Υ_2 are well defined.

Now from (40) and (41), we can rewrite the recursion (39) as

$$\theta_{j+1} = \left(\Upsilon_1\left(\theta_j, \gamma_\rho(J, \theta_j)\right), \Upsilon_2\left(\theta_j, \gamma_\rho\left(J, \theta_j\right)\right)\right)^\top. \tag{43}$$

The above update rule for recursively generating model sequence $\{\theta_j\}$ is commonly referred to as the *ideal version of the standard CE method*. However, in this paper, we employ an extended version of the CE method proposed in Joseph and Bhatnagar (2016a, b, c) whose update rule is slightly different. In the extended version, a mixture PDF $\widehat{f}_{\theta_j} = (1-\zeta) f_{\theta_j} + \zeta f_{\theta_0}$ (with $\zeta \in (0, 1)$ and θ_0 is the initial distribution parameter) is employed to compute γ_ρ, Υ_1 and Υ_2 instead of the original PDF f_{θ_j}. In this case, the update rule is defined as follows:

$$\theta_{j+1} = \left(\Upsilon_1\left(\widehat{\theta}_j, \gamma_\rho(J, \widehat{\theta}_j)\right), \Upsilon_2\left(\widehat{\theta}_j, \gamma_\rho\left(J, \widehat{\theta}_j\right)\right)\right)^\top. \tag{44}$$

Here $\gamma_\rho(J, \widehat{\theta})$ is defined as the $(1-\rho)$-quantile of J w.r.t. the mixture distribution $\widehat{f}_\theta$. Similarly we define $\Upsilon_1(\widehat{\theta}, \cdot)$ and $\Upsilon_2(\widehat{\theta}, \cdot)$ respectively. This extended version is shown to exhibit global optimum convergence (Joseph and Bhatnagar 2016a, b, c).

However, there are certain tractability concerns. The quantities $\gamma_\rho(J, \widehat{\theta}_j)$, $\Upsilon_1(\widehat{\theta}_j, \cdot)$ and $\Upsilon_2(\widehat{\theta}_j, \cdot)$ involved in the update rule are intractable, i.e. computationally hard to compute (and hence the tag name '*ideal*'). To overcome this, a naive approach usually found in the literature is to employ sample averaging, with sample size increasing to infinity. However, this approach suffers from hefty storage and computational complexity which is primarily attributed to the accumulation and processing of huge number of samples. In Joseph and Bhatnagar (2016a, b, c), a stochastic approximation variant of the extended cross entropy method has been proposed. The proposed approach is efficient both computationally and storage wise, when compared to the rest of the state-of-the-art CE tracking methods (Hu et al. 2012; Wang and Enright 2013; Kroese et al. 2006). It also integrates the mixture approach (44) and henceforth exhibits global optimum convergence.

The goal of the stochastic approximation (SA) version of Gaussian CE method is to find a sequence of Gaussian model parameters $\{\theta_j = (\mu_j, \Sigma_j)^\top\}$ (where μ_j is the mean vector and Σ_j is the covariance matrix) which tracks the ideal CE method. The algorithm efficiently accomplishes the goal by employing multiple stochastic approximation recursions. The algorithm is shown to exhibit global optimum convergence, i.e., the model sequence $\{\theta_j\}$ converges to the degenerate distribution concentrated on any of the global optima of the objective function (Fig. 5), in both deterministic (when the objective function is deterministic) and stochastic settings, i.e., when noisy versions of the objective function are available. Successful application of the stochastic approximation version of CE in stochastic settings is appealing to the control problem we consider in this paper, since the off-policy LSTD(λ) method only provides estimates of the value function. The SA version of CE is a discrete evolutionary procedure where the model sequence $\{\theta_j\}$ is adapted to the degenerate distribution concentrated at global optima, where at each discrete step of the evolution a single sample from the solution space is used. This unique nature of the SA version is appealing to settings where the objective function values are hard to obtain, especially to the MDP control problem we consider in this paper. The single sample requirement attribute which is unique to the SA version implies that one does not need to scale the computing machine for unnecessary value function evaluations.

Our algorithm which attempts to solve the control problem defined in Eq. (15) is formally illustrated in Algorithm 2.

A few remarks about the algorithm are in order:

1. The learning rates $\{\bar{\beta}_j\}$, $\{\beta_j\}$ and the mixing weight ζ are deterministic, non-increasing and satisfy the following:

$$\begin{aligned} &\zeta \in (0,1), \beta_j > 0, \bar{\beta}_j > 0, \\ &\sum_{j=1}^{\infty} \beta_j = \infty, \sum_{j=1}^{\infty} \bar{\beta}_j = \infty, \sum_{j=1}^{\infty}\left(\beta_j^2 + \bar{\beta}_j^2\right) < \infty. \end{aligned} \tag{45}$$

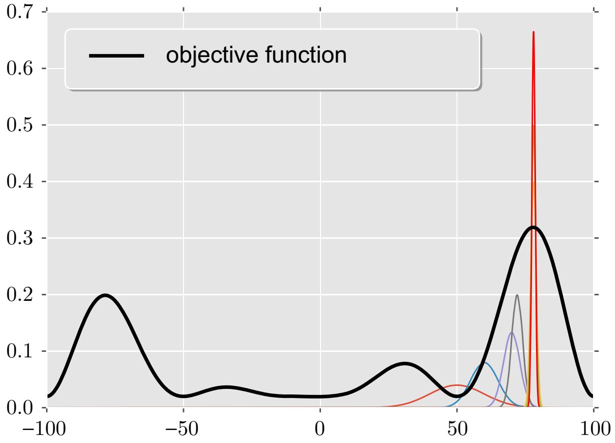


Fig. 5 Illustration of the sequence $\{\theta_j\}$ generated by the CE method

2. In our algorithm, the objective function is estimated in (50) using the *Predict* procedure which is defined in Algorithm 1. Even though an infinitely long sample trajectory is assumed to be available, the *Predict* procedure has to practically terminate after processing a finite number of transitions from the trajectory. Hence a user configured trajectory length rule $\{N_j \in \mathbb{N} \setminus \{0\}\}_{j \in \mathbb{N}}$ with $N_j \uparrow \infty$ is used. At each iteration j of the cross entropy method, when *Predict* procedure is invoked to estimate the objective function $L(h_{w_j|w_j})$, the procedure terminates after processing the first N_j transitions in the trajectory. It is also important to note that the same sample trajectory is reused for all invocations of *Predict*. This eliminates the need for any further observations of the MDP.
3. Recall that we employ the stochastic approximation (SA) version of the extended CE method to solve our control problem (15). The SA version (hence Algorithm 2) maintains three variables: $\gamma_j, \xi^{(0)}_j$ and $\xi^{(1)}_j$, with γ_j tracking $\gamma_\rho(\cdot, \widehat{\theta}_j)$, while $\xi^{(0)}_j$ and $\xi^{(1)}_j$ track $\Upsilon_1(\widehat{\theta}_j, \cdot)$ and $\Upsilon_2(\widehat{\theta}_j, \cdot)$ respectively. Their stochastic recursions are defined in Eqs. (51), (52) and (53) of Algorithm 2. The increment terms for their respective stochastic recursions are defined recursively as follows:

$$\Delta\gamma_j(y) \triangleq -(1-\rho)I_{\{y \geq \gamma_j\}} + \rho I_{\{y \leq \gamma_j\}}. \tag{46}$$

$$\Delta\xi^{(0)}_j(x, y) \triangleq \mathbf{g_1}(y, x, \gamma_j) - \xi^{(0)}_j \mathbf{g_0}(y, \gamma_j). \tag{47}$$

$$\Delta\xi^{(1)}_j(x, y) \triangleq \mathbf{g_2}(y, x, \gamma_j, \xi^{(0)}_j) - \xi^{(1)}_j \mathbf{g_0}(y, \gamma_j). \tag{48}$$

4. The initial distribution parameter θ_0 is chosen by hand such that probability density function f_{θ_0} has strictly positive values for every point in the solution space $\mathbb{W}$, i.e., $f_{\theta_0}(w) > 0, \forall w \in \mathbb{W}$.
5. The stopping rule we adopt here for the control problem is to terminate the algorithm when the model sequence $\{\theta_j\}$ is sufficiently close consequently for a finitely long time, i.e., $\exists \bar{j} \geq 0$ *s.t.* $\|\theta_j - \theta_{j+1}\| < \delta_1, \bar{j} \leq \forall j \leq \bar{j} + N(\delta_1)$, where $\delta_1 \in \mathbb{R}_+, N(\delta_1) \in \mathbb{N}$ are decided *a priori*.
6. The quantile factor ρ is also a relevant parameter of the CE method. An empirical analysis in Joseph and Bhatnagar (2016b) has revealed that the convergence rate of the algorithm is sensitive to the choice of ρ. The paper also recommends that [0.01, 0.3] is the most suitable choice of ρ.

7. We also extended the algorithm to include Polyak averaging of the model sequence $\{\theta_j\}$. The sequence $\{\bar{\theta}_j\}$ maintains the Polyak averages of the sequence $\{\theta_j\}$ and its update step is given in (57). Note that the Polyak averaging Polyak and Juditsky (1992) is a double averaging technique which does not cripple the convergence of the original sequence $\{\theta_j\}$, however it reduces the variance of the iterates and accelerates the convergence of the sequence.

Algorithm 2

```
Input parameters: $\epsilon, \rho \in (0, 1), \bar{\beta}_j, \beta_j, \zeta, c_j \in (0, 1), c_j \to 0, \theta_0 := (\mu_0, \Sigma_0)^\top$,
        $\{N_j, j \in \mathbb{N}\}$   ▶ Trajectory length rule chosen a priori
Initialization: $j := 0, \gamma_0 := 0, \xi_0^{(0)} := 0_{k_2 \times 1}, \xi_0^{(1)} := 0_{k_2 \times k_2}, T_0 := 0, \theta^p = NULL, \gamma_0^p := -\infty$.
while stopping criteria not satisfied do
    Mixture distribution generation: $\widehat{f}_{\theta_j} := (1 - \zeta) f_{\theta_j} + \zeta f_{\theta_0}$;   (49)
    Sample generation: $\mathbf{w}_{j+1} \sim \widehat{f}_{\theta_j}(\cdot)$;
    Objective function estimation: $\hat{J}(\mathbf{w}_{j+1}) := Predict(\mathbf{w}_{j+1}, N_{j+1})$;   (50)
    Tracking $\gamma_\rho(\hat{J}, \widehat{\theta}_j)$:   $\gamma_{j+1} := \gamma_j - \beta_j \Delta\gamma_j(\hat{J}(\mathbf{w}_{j+1}))$;   (51)
    Tracking $\Upsilon_1(\widehat{\theta}_j, \gamma_\rho(\hat{J}, \widehat{\theta}_j))$:   $\xi_{j+1}^{(0)} := \xi_j^{(0)} + \beta_j \Delta\xi_j^{(0)}(\mathbf{w}_{j+1}, \hat{J}(\mathbf{w}_{j+1}))$;   (52)
    Tracking $\Upsilon_2(\widehat{\theta}_j, \gamma_\rho(\hat{J}, \widehat{\theta}_j))$:   $\xi_{j+1}^{(1)} := \xi_j^{(1)} + \beta_j \Delta\xi_j^{(1)}(\mathbf{w}_{j+1}, \hat{J}(\mathbf{w}_{j+1}))$;   (53)
    if $\theta^p \neq NULL$ then
        $\mathbf{w}_{j+1}^p \sim \widehat{f}_{\theta^p} := (1 - \zeta) f_{\theta^p} + \zeta f_{\theta_0}$;
        $\gamma_{j+1}^p := \gamma_j^p - \beta_j \Delta\gamma_j^p(\hat{J}(\mathbf{w}_{j+1}^p))$;
    end if
    Threshold comparison: $T_{j+1} := T_j + c\left(I_{\{\gamma_j > \gamma_j^p\}} - I_{\{\gamma_j \leq \gamma_j^p\}} - T_j\right)$;   (54)
    if $T_{j+1} > \epsilon$ then
        Save old model and old threshold:   $\theta_{j+1}^p := \theta_j$;   $\gamma_{j+1}^p := \gamma_j$;
        Model parameter update:   $\theta_{j+1} := \theta_j + \beta_j \left((\xi_j^{(0)}, \xi_j^{(1)})^\top - \theta_j\right)$;   (55)
        Reset parameters:   $T_j := 0$;   $c := c_j$;   (56)
        Weighted Polyak averaging:   $\bar{\theta}_{j+1} := \bar{\theta}_j + \bar{\beta}_j \left(\theta_{j+1} - \bar{\theta}_j\right)$;   (57)
    else
        $\gamma_{j+1}^p = \gamma_j^p$;   $\theta_{j+1} = \theta_j$;
    end if
    $j := j + 1$;
end while
```

3.4 Convergence analysis of Algorithm 2

The convergence analysis of the generalized variant of Algorithm 2 is already addressed in Joseph and Bhatnagar (2016c) and its application to the prediction problem is given in Joseph and Bhatnagar (2016b). However, for completeness, we will restate the results here. We do not give proof of those results, however, provide references for the same. The additional Polyak averaging (step 19 of Algorithm 2) requires analysis, which is covered below.

Note that Algorithm 2 employs the off-policy prediction method for estimating the objective function. In particular, in step 6 of Algorithm 2, we have $\hat{J}(\mathbf{w}_{j+1}) := Predict(\mathbf{w}_{j+1}, N_{j+1})$, which converges to $\mathbb{E}_{\nu_{w_b}}\left[L(x_{w|w_b}^{\top}\phi(\mathbf{s}))\right]$ almost surely as $N_j \to \infty$ (by Lemma 1). Hence the objective function optimized by Algorithm 2 is $J_b(w) \triangleq \mathbb{E}_{\nu_{w_b}}\left[L(x_{w|w_b}^{\top}\phi(\mathbf{s}))\right]$, where $w_b \in \mathbb{W}$ is the chosen behaviour policy vector.

Also note that the model parameter θ_j in Algorithm 2 is not updated at each iteration j. Rather it is updated whenever T_j hits the ϵ threshold (step 15 of Algorithm 2), where $\epsilon \in (0, 1)$ is a constant. So the update of θ_j only happens along a sub-sequence $\{j_{(n)}\}_{n\in\mathbb{N}}$ of $\{j\}_{j\in\mathbb{N}}$. Between $j = j_{(n)}$ and $j = j_{(n+1)}$, the model parameter θ_j remains constant and the variable γ_j estimates $(1-\rho)$-quantile of J_b w.r.t. $\widehat{f}_{\theta_{j_{(n)}}}$.

Notation We denote by $\gamma_\rho(J_b, \widehat{\theta})$, the $(1-\rho)$-quantile of J_b w.r.t. the mixture distribution $\widehat{f}_\theta$ and let $E_{\widehat{\theta}}[\cdot]$ be the expectation w.r.t. $\widehat{f}_\theta$.

Since the model parameter θ_j remains constant between $j = j_{(n)}$ and $j = j_{(n+1)}$, the convergence behaviour of γ_j, $\xi_j^{(0)}$ and $\xi_j^{(1)}$ can be studied by keeping θ_j constant.

Lemma 2 *Let $\theta_j \equiv \theta, \forall j$. Also, assume $sup_j|\gamma_j| < \infty$ a.s. Then the stochastic sequence $\{\gamma_j\}$ defined in Eq. (51) satisfies* $\lim_{j\to\infty}\gamma_j = \gamma_\rho(J_b, \widehat{\theta})$ *a.s.*

Proof Refer Lemma 3 of Joseph and Bhatnagar (2016b). □

Lemma 3 *Assume $\theta_j \equiv \theta, \forall j$. Then almost surely,*

(i)
$$\lim_{j\to\infty}\xi_j^{(0)} = \xi_*^{(0)} = \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g_1}\big(J_b(\mathbf{x}), \mathbf{x}, \gamma_\rho(J_b, \widehat{\theta})\big)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g_0}\big(J_b(\mathbf{x}), \gamma_\rho(J_b, \widehat{\theta})\big)\right]}.$$

(ii)
$$\lim_{j\to\infty}\xi_j^{(1)} = \xi_*^{(1)} = \frac{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g_2}\big(J_b(\mathbf{x}), \mathbf{x}, \gamma_\rho(J_b, \widehat{\theta}), \xi_*^{(0)}\big)\right]}{\mathbb{E}_{\widehat{\theta}}\left[\mathbf{g_0}\big(J_b(\mathbf{x}), \gamma_\rho(J_b, \widehat{\theta})\big)\right]}.$$

(iii) *T_j defined in Eq. (54) satisfies $-1 < T_j < 1, \forall j$.*

(iv) *If $\gamma_\rho(J_b, \widehat{\theta}) > \gamma_\rho(J_b, \widehat{\theta^p})$, then $T_j, j \geq 1$ in (54) satisfy* $\lim_{j\to\infty} T_j = 1$ *a.s.*

Proof For (i), (ii) and (iv), refer Lemma 4 of Joseph and Bhatnagar (2016b). For (iii) refer Proposition 1 of Joseph and Bhatnagar (2016b). □

Notation For the subsequence $\{j_{(n)}\}_{n>0}$ of $\{j\}_{j\in\mathbb{N}}$, we denote $j_{(n)}^{-} \triangleq j_{(n)} - 1$ for $n > 0$.

Along the subsequence $\{j_{(n)}\}_{n\geq 0}$ with $j_0 = 0$ the updating of θ_j can be expressed as follows:

$$\theta_{j_{(n+1)}} := \theta_{j_{(n)}} + \beta_{j_{(n)}}\Delta\theta_{j_{(n)}}, \tag{58}$$

where $\Delta\theta_{j_{(n)}} = (\xi^{(0)}_{j^{-}_{(n+1)}}, \xi^{(1)}_{j^{-}_{(n+1)}})^{\top} - \theta_{j_{(n)}}$.

We now present our main result. The following theorem shows that the model sequence $\{\theta_j\}$ and the averaged sequence $\{\bar{\theta}_j\}$ generated by Algorithm 2 converge to the degenerate distribution concentrated on the global maximum of the objective function J_b.

Theorem 3 *Let $\varphi(x) = exp(rx)$, $r \in \mathbb{R}$. Let $\rho, \zeta \in (0, 1)$. Let the learning rates $\{\bar{\beta}_j\}$ and $\{\beta_j\}$ satisfy Eq. (45). Assume $J_b \in \mathcal{C}^2$. Let $\{\theta_j = (\mu_j, \Sigma_j)\}_{j \in \mathbb{N}}$ and $\{\bar{\theta}_j = (\bar{\mu}_j, \bar{\Sigma}_j)\}_{j \in \mathbb{N}}$ be the sequences generated by Algorithm 2 and also assume $\theta_j \in \Theta$, $\forall j \in \mathbb{N}$. Let $\bar{\beta}_j = o(\beta_j)$. Let $w_b \in \mathbb{W}$ be the chosen behaviour policy vector. Also, let the assumptions (A1–A5) hold. Then*

$$\theta_j \to (w^{b*}, 0_{k_2 \times k_2})^\top \text{ as } j \to \infty \quad w.p.1, \tag{59}$$

$$\bar{\theta}_j \to (w^{b*}, 0_{k_2 \times k_2})^\top \text{ as } j \to \infty \quad w.p.1, \tag{60}$$

where $w^{b*} \in \arg\max_{w \in \mathbb{W}} J_b(w)$ *with* $J_b(w) \triangleq \mathbb{E}_{\nu_{w_b}}\left[L(x_{w|w_b}^\top \phi(\mathbf{s}))\right]$.

Proof Since $\bar{\beta}_j = o(\beta_j)$, $\bar{\beta}_j \to 0$ faster than $\beta_j \to 0$. This implies that the updates of θ_j in (55) are larger than those of $\bar{\theta}_j$ in (57). Hence the sequence $\{\theta_j\}$ appears quasi-convergent when viewed from the timescale of $\{\bar{\theta}_j\}$ sequence.

Theorem 2 of Joseph and Bhatnagar (2016b) analyses the limiting behaviour of the stochastic recursion (55) of Algorithm 2 in great detail. The analysis discloses the global optimum convergence of the algorithm under limited regularity conditions. It is shown that the model sequence $\{\theta_j\}$ converges almost surely to the degenerate distribution concentrated on the global optimum. The proposed regularity conditions for the global optimum convergence are that the objective function belongs to $\mathcal{C}^2$ and the existence of a Lyapunov function on the neighbourhood of the degenerate distribution concentrated on the global optimum. This justifies the hypothesis $J_b \in \mathcal{C}^2$ in the statement of the theorem and we further assume the existence of a Lyapunov function on the neighbourhood of the degenerate distribution $(w^{b*}, 0_{k_2 \times k_2})^\top$. Then by Theorem 2 of Joseph and Bhatnagar (2016b), we deduce that $\{\theta_j\}$ converges to $(w^{b*}, 0_{k_2 \times k_2})^\top$. This completes the proof of (59).

For brevity, lets define $\theta^* \triangleq (w^{b*}, 0_{k_2 \times k_2})^\top$. We also define the filtration $\{\bar{\mathcal{F}}_j\}_{j \in \mathbb{N}}$, where the σ-field $\bar{\mathcal{F}}_j \triangleq \sigma(\theta_i, \bar{\theta}_i, 0 \le i \le j\})$. Now recalling recursion (57),

$$\begin{aligned}
\bar{\theta}_{j+1} &:= \bar{\theta}_j + \bar{\beta}_{j+1}\left(\theta_{j+1} - \bar{\theta}_j\right), \\
&:= \bar{\theta}_j + \bar{\beta}_{j+1}\left(\theta_j - \mathbb{E}\left[\theta_{j+1}|\bar{\mathcal{F}}_j\right] + \mathbb{E}\left[\theta_{j+1}|\bar{\mathcal{F}}_j\right] - \theta^* + \theta^* - \bar{\theta}_j\right), \\
&:= \bar{\theta}_j + \bar{\beta}_{j+1}\left(\overline{\mathbb{M}}_{j+1} + \bar{b}_j + \bar{h}(\bar{\theta}_j)\right),
\end{aligned}$$

where $\overline{\mathbb{M}}_{j+1} \triangleq \theta_{j+1} - \mathbb{E}\left[\theta_{j+1}|\bar{\mathcal{F}}_j\right]$, $\bar{b}_j \triangleq \mathbb{E}\left[\theta_{j+1}|\bar{\mathcal{F}}_j\right] - \theta^*$ and $\bar{h}(x) \triangleq \theta^* - x$.

Here we make the following observations:

1. $\bar{b}_j \to 0$ almost surely as $j \to \infty$. This follows from the hypothesis $\bar{\beta}_j = o(\beta_j)$ and by considering the fact that $\theta_j \to \theta^*$ almost surely.
2. $\bar{h}$ is Lipschitz continuous.
3. $\{\overline{\mathbb{M}}_j\}$ is a martingale difference sequence.
4. $\{\bar{\theta}_j\}$ is stable, i.e., $\sup_j \|\bar{\theta}_j\| < \infty$.
5. The ODE defined by $\dot{\bar{\theta}}(t) = \bar{h}(\bar{\theta}(t))$ is globally asymptotically stable at θ^*.

All the above facts are easy to verify. Now by appealing to the third extension of Theorem 2, Section 2.2, Chapter 2 of Borkar (2008) and from the above observations, we can henceforth conclude that $\bar{\theta}_j \to \theta^*$ almost surely as $j \to \infty$. This completes the proof of (60). □

4 Experimental illustrations

The performance of our algorithm is evaluated on four different MDP settings:

1. Chain walk MDP.
2. Linearized cart-pole balancing.
3. 5-link actuated pendulum balancing.
4. Random MDP.

Our algorithm is compared against the state-of-the-art algorithms such as least squares policy iteration (LSPI), fast policy search method, model reference adaptive search (MRAS) and simultaneous perturbation stochastic approximation (SPSA). In each setting, the results shown are averages over 10 independent sample sequences generated by the algorithms with different initial conditions. The function $\varphi(\cdot)$ used here is $\varphi(x) = \exp(rx)$, where $r \in \mathbb{R}_+$.

4.1 Experiment 1: chain walk

This particular setting (Fig. 6) which has been proposed in Koller and Parr (2000) demonstrates the unique scenario where policy iteration is non-convergent when approximate value functions are employed instead of true ones. This particular example is also utilized to empirically evaluate the performance of LSPI in Lagoudakis and Parr (2003). Here, we compare the performance of our algorithm against LSPI and also against the stable Q-learning algorithm with linear function approximation (called Greedy-GQ) proposed in Maei et al. (2010). This particular demonstration is pertinent in two ways: (1) when LSPI was evaluated on this setting, the maximum state space cardinality considered was 50. We consider here a larger MDP with 450 states and (2) the stable Greedy-GQ algorithm is only evaluated over a small experimental setting in Maei et al. (2010). Here, by applying it on a relatively harder setting, we attempt to assess its applicability and robustness.

Setup We consider a Markov decision process with $|\mathbb{S}| = 450$, $\mathbb{A} = \{L, R\}$, $k_1 = 5$, $k_2 = 10$ and the discount factor $\gamma = 0.99$.

Reward function $R(\cdot, \cdot, 150) = R(\cdot, \cdot, 300) = 1.0$ and zero for all other transitions. This implies that only the transitions to states 150 and 300 will acquire a positive payoff, while the rest are nugatory transitions.

Transition dynamics The transition probability kernel is defined as follows:

$$\text{For } 1 < s < |\mathbb{S}| \begin{cases} P(s, L, s+1) = 0.1, & P(s, L, s-1) = 0.9, \\ P(s, R, s+1) = 0.9, & P(s, R, s-1) = 0.1. \end{cases}$$

$$\begin{aligned} &P(1, L, 2) = 0.1,\ \ P(1, L, 1) = 0.9, \\ &P(1, R, 2) = 0.9,\ \ P(1, R, 1) = 0.1, \\ &P(|\mathbb{S}|, L, |\mathbb{S}|) = 0.1,\ \ P(|\mathbb{S}|, L, |\mathbb{S}| - 1) = 0.9, \\ &P(|\mathbb{S}|, R, |\mathbb{S}|) = 0.9,\ \ P(|\mathbb{S}|, R, |\mathbb{S}| - 1) = 0.1, \end{aligned}$$

Feature set We employ radial basis functions (RBF) as both policy and prediction features. We utilize 5 RBFs for prediction and 10 for policy features, i.e., $k_1 = 5$ and $k_2 = 10$. Note that RBFs are Gaussian kernels which are parametrized by the centroid $m \in \mathbb{R}$ and spread $v \in \mathbb{R}_+$ and are expressed as:

$$b(s) = e^{-\frac{(s-m)^2}{2.0v^2}}. \tag{61}$$

In our experiments, we initially tried to employ polynomials for features and found that the approximations they produced were quite poor. However, with RBFs one can indeed obtain decent performance by uniformly distributing the centroids in the state or state-action space

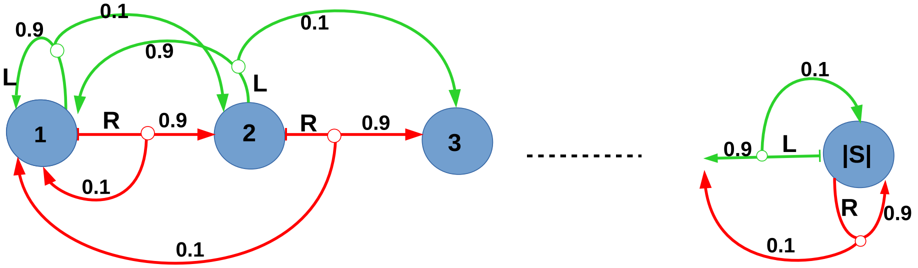


Fig. 6 Chain walk MDP

and by considering the spread to be the half of the distance between subsequent centroids. In this way, one can indeed cover the respective spaces reasonably well. The policy features and the prediction features are defined as follows:

Policy features	Prediction features
$\psi(s,a) = \begin{pmatrix} I_{\{a=L\}} e^{-\frac{(s-m_1)^2}{2.0v_1^2}} \\ \vdots \\ I_{\{a=L\}} e^{-\frac{(s-m_5)^2}{2.0v_5^2}} \\ I_{\{a=R\}} e^{-\frac{(s-m_1)^2}{2.0v_1^2}} \\ \vdots \\ I_{\{a=R\}} e^{-\frac{(s-m_5)^2}{2.0v_5^2}} \end{pmatrix}.$	$\phi_i(s) = e^{-\frac{(s-m_i)^2}{2.0v_i^2}},$

where $m_i = 5 + 10(i-1)$, $v_i = 5$, $1 \leq i \leq 5$.

Behaviour policy This is the most important choice and one has to be discreet while choosing the behaviour policy. For this setting, we prefer a policy which is unbiased and which uniformly covers the action space to provide sufficient exploration. Henceforth, by choosing $w_b = (0, 0, \ldots, 0)^\top$ we obtain a uniform distribution over action space for every state in $\mathbb{S}$.

Performance function Note that both LSPI and Q-learning seek in the policy parameter space to find the optimal or sub-optimal policy by recalibrating the parameter vector at each iteration in the direction of the improved value function. But the objective function that we consider in this paper is a more generalized version involving the performance function L and scalarization using $\mathbb{E}_{\nu_w}[\cdot]$. So the predicament, the above algorithms attempt to resolve becomes a special instance of our generalized version and hence to compare our algorithm against them, we consider the objective function to be the weighted Euclidean norm of the approximate value function (with weight being the stationary distribution ν_w). Therefore, the performance function L is defined as $L(h_{w|w}) = h^2_{w|w}$ (where squaring of the vector is defined as squaring of each of its components). Note that, in our algorithm, we approximate $h_{w|w}$ using the behaviour policy and the true approximation and the stationary distribution involved are $\Phi x_{w|w_b}$ and ν_{w_b} respectively. However, since the behaviour policy chosen is the

Table 1 Algorithm parameter values used in the chain walk experiment

β_j	0.2
$\bar{\beta}_j$	0
ζ	0
c_j	0.08
ρ	0.05
ϵ	0.9
τ	1.0
r	0.01

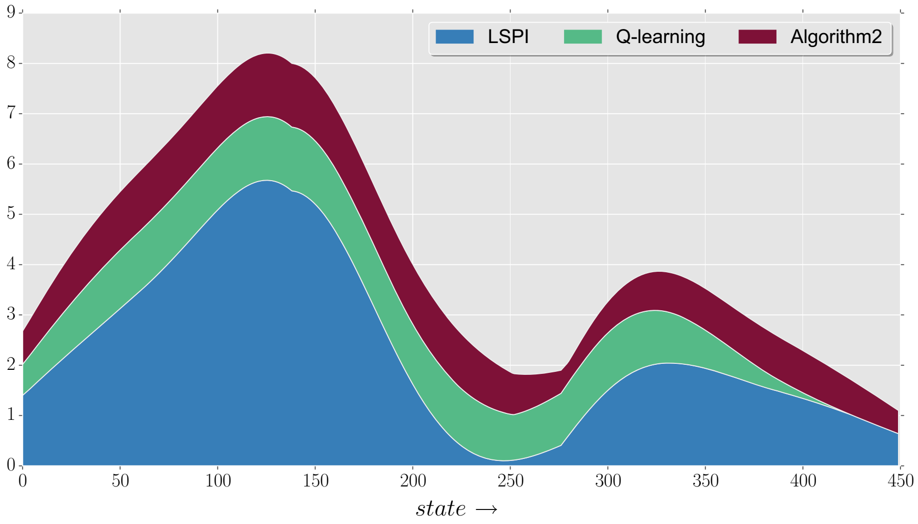


Fig. 7 The plot of the respective optimal value functions contrived by LSPI, Q-learning and Algorithm 2 for the chain walk MDP setting. The optimal solutions of various algorithms are being developed by averaging over 10 independent trials. For Algorithm 2, we averaged the various optimal solutions obtained for different sample trajectories generated using the same behaviour policy, but with different initial states which are chosen randomly. Our approach (Algorithm 2) literally surpassed other algorithms in terms of its quality. The random choice of the initial state effectively favoured sufficient exploration of the state space which directly assisted in generating high quality solutions

uniform distribution over the action space for each state in $\mathbb{S}$, one can easily deduce that the underlying Markov chain of the behaviour policy is a uniform random walk and its stationary distribution is the uniform distribution over the state space $\mathbb{S}$.

The various parameter values employed and the results obtained in the experiment are provided in Table 1 and Fig. 7 respectively.

4.2 Experiment 2: 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 in the interval $[-\pi, \pi]$ with negative angle representing the rotation in the counter clockwise direction. The cart is free to move in either direction within the bounds of a linear track and the distance lies in the region $[-4.0, 4.0]$ with negative distance representing the movement to the left of the origin.In our experiment, we have $m = 0.5$, $M = 0.5$, $l = 20.5$ and the discount factor $\gamma = 0.1$.

Goal To bring the cart to the equilibrium position, i.e., to balance the pole upright and the cart at the centre of the track.

State space The state is the 4-tuple $(x, \dot{x}, \psi, \dot{\psi})^\top$ where ψ is the angle of the pendulum w.r.t. the vertical axis, $\dot{\psi}$ is the angular velocity, x the relative cart position from the centre of the track and $\dot{x}$ is its velocity. For better tractability, we restrict $\dot{x} \in [-5.0, 5.0]$ and $\dot{\psi} \in [-5.0, 5.0]$, respectively.

Control (Policy) space The controller applies a horizontal force a on the cart parallel to the track. The stochastic policy used in this setting corresponds to $\pi(a|s) = \mathcal{N}(a|\vartheta^\top s, \sigma^2)$ (normal distribution with mean $\vartheta^\top s$ and standard deviation σ). Here the policy is parametrized by $\vartheta \in \mathbb{R}^4$ and $\sigma \in \mathbb{R}$.

System dynamics The dynamical equations of the system are given by

$$\ddot{\psi} = \frac{-3ml\dot{\psi}^2 \sin\psi \cos\psi + (6M + m)g \sin\psi - 6(a - b\dot{\psi}) \cos\psi}{4l(M + m) - 3ml \cos\psi}, \tag{62}$$

$$\ddot{x} = \frac{-2ml\dot{\psi}^2 \sin\psi + 3mg \sin\psi \cos\psi + 4a - 4b\dot{\psi}}{4(M + m) - 3m \cos\psi}. \tag{63}$$

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

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \\ \psi_{t+1} \\ \dot{\psi}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ \dot{x}_t \\ \psi_t \\ \dot{\psi}_t \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\psi}_t \\ \frac{3(M+m)\psi_t - 3a + 3b\dot{\psi}_t}{4Ml - ml} \\ \dot{x}_t \\ \frac{3mg\psi_t + 4a - 4b\dot{\psi}_t}{4M - m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{z} \end{bmatrix}, \tag{64}$$

where b is the friction coefficient of the cart on the floor, $g = 9.81\frac{m}{sec^2}$ is the gravitational constant, Δt is the integration time step, i.e., the time difference between two transitions and $\mathbf{z}$ is a standard Gaussian noise on the velocity of the cart. In our experiment, we set $b = 0.1 Newton(msec)^{-1}$ and $\Delta t = 0.1 sec$, respectively.

Reward function $R(s, a) = R(\psi, \dot{\psi}, x, \dot{x}, a) = -4\psi^2 - x^2 - 0.1a^2$. The reward function can be viewed as assigning penalty which is directly proportional to the deviation from the equilibrium state.

Prediction features $\phi(s \in \mathbb{R}^4) = (1, s_1^2, s_2^2 \ldots, s_1 s_2, s_1 s_3, \ldots, s_3 s_4)^\top \in \mathbb{R}^{11}$.

Behaviour policy $\pi_b(a|s) = \mathcal{N}(a|\vartheta_b^\top s, \sigma_b^2)$, where $\vartheta_b = (3.684, 3.193, 4.252, 3.401)^\top$ and $\sigma_b = 5.01$. The behaviour policy is determined by vaguely solving the problem using true value functions and then choosing the behaviour policy vector ϑ_b by perturbing each component of the vague solution so obtained. The margin of perturbation we considered is chosen randomly from the interval $[-5.0, 5.0]$.

Performance function The performance function L is defined as under: We randomly select (from the given intervals described in the definition of the state space), $s_0 = (0.235, 3.581, 2.276, 1.069)^\top$. Now, define

$$L(h_{w|w})(s) = \begin{cases} 0.1 h_{w|w}(s_0), & \text{for } s = s_0 \\ 0, \forall s \in \mathbb{S} \setminus \{s_0\}. \end{cases} \tag{65}$$

Table 2 Algorithm parameter values used in the experiments

	Cart-pole experiment	Actuated pendulum balancing
β_j	0.7	0.7
$\bar{\beta}_j$	$j_{(n)}^{-1}$	$j_{(n)}^{-1}$
ζ	$j_{(n)}^{-1}$	$j_{(n)}^{-1}$
λ	0.1	0.1
c_j	0.1	0.1
ρ	0.01	0.01
ϵ	0.9	0.9
r	0.01	0.01
N_j	4000, $\forall j$	4000, $\forall j$

Note that $\{j_{(n)}\}$ is the subsequence of $\{j\}$ when recursion (57) is executed

Here s_0 is the initial state of the cart-pole system which implies that the cart is initially stationed at a distance of 0.235 from the centre and the pendulum is at an angle of 2.276 ($= \frac{\pi}{1.38}$) from the vertical position. The initial velocity of the cart and the angular velocity of the pendulum are 3.581 and 1.069 respectively. The goal is to find the optimal policy (which corresponds to the parameters of the horizontal force) to bring the cart to the equilibrium position, i.e., cart at the centre of the track and the pendulum in the vertical position. The nature of the performance function L in Eq. (65) is to explicitly capture this aspect of the problem, i.e., to find the optimal policy that takes the cart from s_0 to the equilibrium position and hence, only the cumulative cost incurred starting from s_0 is considered. Note that s_0 is chosen arbitrarily for the experiment and thus does not render any particular advantage to any of the algorithms.

The various parameter values employed and the results obtained in the experiment are provided in Table 2 and Fig. 8 respectively.

4.3 Experiment 3: 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. In our experiment, we take $m = 1.5$, $l = 10.0$ and the discount factor $\gamma = 0.1$.

Goal To keep all the poles in the horizontal 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 w.r.t. the horizontal axis and $\dot{\psi}_i$ is the angular velocity. In our experiment, we consider the following bounds on the state space: $\psi_i \in [-\pi, \pi], \forall 1 \leq i \leq 5$ and $\dot{\psi}_i \in [-5.0, 5.0], \forall 1 \leq i \leq 5$.

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

$$\pi(a|s) = \mathcal{N}_5(a|As, B) \quad \text{where } A \in \mathbb{R}^{5\times 10}, B \in \mathbb{R}^{5\times 5}. \tag{66}$$

We assume that the torques a_i applied at each joint are independent and hence B is a diagonal matrix. The policy parameter space $\mathbb{W}$ is defined as $\mathbb{W} = \{w \in \mathbb{R}^{55} | w = (A_{00}, A_{01}, A_{02}, \ldots, A_{48}, A_{49}, B_{00}, B_{11}, \ldots, B_{44})^\top\}$.

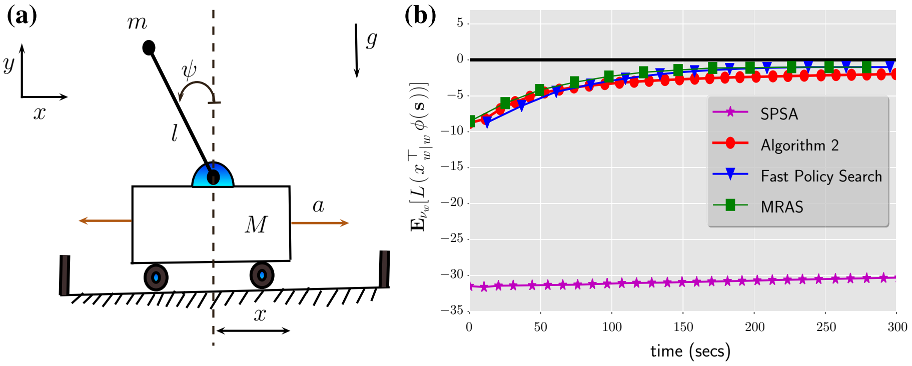


Fig. 8 **a** The cart-pole system: the goal is to keep the pole in the upright position and the cart at the centre of the track by applying a force a either to the right or to the left. The system is parametrized by the position x of the cart, the angle of the pole ψ, the velocity $\dot{x}$ and the angular velocity $\dot{\psi}$. **b** Cart-pole results. Here, for Algorithm 2, we plot $\mathbb{E}_{\nu_{w_b}}\left[L\left(x^\top_{\bar{\mu}_j|w_b}\phi(\mathbf{s})\right)\right]$, where $\bar{\mu}_j$ is the mean vector of the Polyak averaged model sequence $\{\bar{\theta}_j\}$, i.e., $\bar{\theta}_j = (\bar{\mu}_j, \bar{\Sigma}_j)^\top$. For the other algorithms, i.e., SPSA, MRAS and fast policy search, we plot $\mathbb{E}_{\nu_{w_j}}\left[L\left(x^\top_{w_j|w_j}\phi(\mathbf{s})\right)\right]$, where $\{w_j \in \mathbb{W}\}$ is the iterative sequence generated by the respective algorithms. This implies that Algorithm 2 operates in the off-policy setting, while the rest of the algorithms utilize on-policy value function approximations to generate the optimal policy vector. With this advantage, the algorithms SPSA, MRAS and fast policy search are expected to perform better as they have complete access to the generative model unlike Algorithm 2 which has access only to the sample trajectory generated by the behaviour policy. Also, note that x-axis is time in seconds relative to the start of the algorithm since MRAS and fast policy search are batch based approaches, while Algorithm 2 and SPSA are incremental schemes. Regarding the accuracy of the solution obtained by our algorithm, note that the global optimum is indeed zero, since the reward function is defined as the negative penalty with respect to the deviation from the equilibrium position and the goal is to bring the cart to the equilibrium position

System dynamics The state equations representing the approximate linear system dynamics are given by

$$\begin{bmatrix} q_{t+1} \\ \dot{q}_{t+1} \end{bmatrix} = \begin{bmatrix} I & \Delta t I \\ -\Delta t M^{-1} U & I \end{bmatrix} \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} + \Delta t \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} a + \mathbf{z} \tag{67}$$

where Δt is the integration time step, i.e., the time difference between two transitions and M is the mass matrix in the horizontal position with $M_{ij} = l^2(6 - max(i, j))m$. U is a diagonal matrix with $U_{ii} = -gl(6-i)m$, where g is the gravitational constant. Each component of $\mathbf{z}$ is a standard Gaussian noise. In our experiment, we take $\Delta t = 0.1$ and $g = 9.8$.

Reward function $R(q, \dot{q}, a) = -q^\top q$. The reward function can be viewed as assigning penalty (negative reward) with respect to the deviation from the optimal pole position (the unique position with zero deviation from the horizontal position and hence attracts no penalty, i.e., highest reward).

Feature vectors $\phi(s \in \mathbb{R}^{10}) = (1, s_1^2, s_2^2 \ldots, s_1 s_2, s_1 s_3, \ldots, s_9 s_{10})^\top \in \mathbb{R}^{46}$.

Behaviour policy The behaviour policy considered in the experiment is given by $\pi_b(a|s) = \mathcal{N}_5(a|A_b s, B_b)$, where

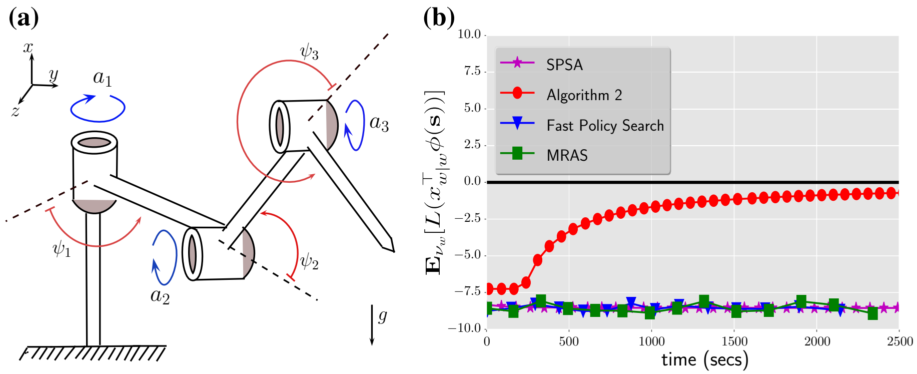


Fig. 9 **a** Each rotational joint i, $1 \leq i \leq 3$ is independently actuated by a torque a_i. The system is parametrized by the angle ψ_i against the horizontal direction and the angular velocity $\dot{\psi}_i$. The goal is to balance the pole in the horizontal direction, i.e., all ψ_i should be as close to 0 as possible by actuating Gaussian torques a_i [Eq. (66)]. **b** Here, for Algorithm 2, we plot $\mathbb{E}_{\nu_{w_b}}\left[L\left(x_{\bar{\mu}_j|w_b}^{\top}\phi(\mathbf{s})\right)\right]$, where $\bar{\mu}_j$ is the mean vector of the Polyak averaged model sequence $\{\bar{\theta}_j\}$, i.e., $\bar{\theta}_j = (\bar{\mu}_j, \bar{\Sigma}_j)^{\top}$. For the other algorithms, i.e., SPSA, MRAS and fast policy search, we plot $\mathbb{E}_{\nu_{w_j}}\left[L\left(x_{w_j|w_j}^{\top}\phi(\mathbf{s})\right)\right]$, where $\{w_j \in \mathbb{W}\}$ is the iterative sequence generated by the respective algorithms. This implies that Algorithm 2 operates in the off-policy setting, while the rest of the algorithms utilize on-policy value function approximations to generate the optimal policy vector. With this advantage, the algorithms MRAS, SPSA and fast policy search are expected to perform better as they have unrestricted access to the generative model unlike Algorithm 2 which has access only to a sample trajectory generated by the behaviour policy. Also, note that x-axis is time in seconds relative to the start of the algorithm since MRAS and fast policy search are batch based approaches, while Algorithm 2 and SPSA are incremental schemes. Again, regarding the accuracy of the solution obtained by our algorithm, note that the global optimum is indeed zero, since the reward function is defined as the negative penalty with respect to the deviation from the equilibrium position and the goal is to bring the system to the equilibrium position. **a** 3-link actuated pendulum setting. **b** 5-link actuated pendulum results

$$A_b^{\top} = \begin{pmatrix} 5.794 & 2.000 & 6.230 & 4.500 & 6.145 \\ 4.843 & 5.014 & 2.306 & 2.796 & 7.000 \\ 6.031 & 6.500 & 6.600 & 8.379 & 4.252 \\ 6.640 & 3.424 & 5.937 & 5.045 & 3.617 \\ 8.661 & 3.463 & 4.430 & 3.000 & 4.233 \\ 5.660 & 3.437 & 7.275 & 7.417 & 5.755 \\ 3.781 & 2.989 & 4.756 & 6.417 & 6.760 \\ 3.391 & 3.696 & 4.153 & 5.761 & 3.196 \\ 5.725 & 2.929 & 3.205 & 3.631 & 8.651 \\ 1.337 & 4.677 & 8.009 & 3.609 & 5.602 \end{pmatrix} \text{ and } B_b = \begin{pmatrix} 5.0 & & & \mathbf{O} & \\ & 5.0 & & & \\ & & 5.0 & & \\ & & & 5.0 & \\ & \mathbf{O} & & & 5.0 \end{pmatrix}.$$

The methodology employed to induce the behaviour policy in this case is similar to that of the cart-pole setting.

Performance function The performance function L is defined as under: We randomly select (from the given intervals described in the definition of the state space), $s_0 = (-1.515, -2.437, -1.386, -3.041, 0.001, 4.510, 0.691, 1.450, 3.241, \quad 3.535)^{\top}$. Now define

$$L(h_{w|w})(s) = \begin{cases} 0.1h_{w|w}(s_0), & \text{for } s = s_0 \\ 0, \forall s \in \mathbb{S} \setminus \{s_0\}. \end{cases} \tag{68}$$

The rationale behind the choice of the above particular performance function is similar to that of Experiment 2. Also, note that s_0 is chosen arbitrarily for the experiment and thus does not accord any unfounded predisposition to any of the algorithms.

The various parameter values employed and the results obtained in the experiment are provided in Table 2 and Fig. 9 respectively.

4.4 Experiment 4: random MDP

Setup We consider a randomly generated Markov decision process with $|\mathbb{S}| = 500$, $|\mathbb{A}| = 30$, $k_1 = 5$, $k_2 = 5$ and $\gamma = 0.8$.

Reward function The reward function R is defined as follows:

$$R(s, a, s') = \omega_1(s)\omega_1(s')\left(\frac{\sin(a) + 2.0}{(1.0 + s')^{0.25}}\right), s, s' \in \mathbb{S}, a \in \mathbb{A}. \tag{69}$$

Here $\omega_1 \in [3, 5]^{|\mathbb{S}|}$ is initialized for the algorithm with $\omega_1(s) \sim U(1, 4)$.

Transition dynamics The transition probability kernel P is defined as follows:

$$P(s, a, s') = \binom{n}{s'}\omega_2(s, a)^{s'}(1.0 - \omega_2(s, a))^{n-s'}, s, s' \in \mathbb{S}, a \in \mathbb{A}. \tag{70}$$

Here the matrix $\omega_2 \in [0, 1]^{\mathbb{S}\times\mathbb{A}}$ is initialized for the algorithm with $\omega_2(s, a) \sim U(0, 1)$.

Feature set The policy features and the prediction features are as follows:

Policy features	Prediction features
$\psi(s, a) = B[s\lvert\mathbb{A}\rvert + a]$	$\phi_i(s) = e^{-\frac{(s-m_i)^2}{2.0v_i^2}}$
where $B = \begin{pmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ \vdots&&\ddots&&\vdots\\ \vdots&&&&\vdots \end{pmatrix}_{15000\times5}$.	where $m_i = 5 + 10(i - 1)$, $v_i = 5$.

In this experimental setting, we employ the Gibbs "softmax" policies defined in Eq. (7).

Behaviour policy The behaviour policy vector w_b considered for the experiment is $w_b = (12.774, 15.615, 20.626, 25.877, 11.945)^\top$.

Performance function The performance function L is defined as follows:

$L(h_{w|w}) = 0.1h^2_{w|w}$ (Note that squaring the vector here corresponds to co-ordinate wise squaring).

As with the previous two experiments, Algorithm 2 was run for the off-policy case while SPSA, MRAS and fast policy search were run for the on-policy setting.

The various parameter values employed and the results obtained in the experiment are provided in Table 3 and Fig. 10 respectively.

Table 3 Algorithm parameter values used in the random MDP experiment

β_j	0.7
$\bar{\beta}_j$	$j_{(n)}^{-1}$
ζ	$j_{(n)}^{-1}$
c_j	0.1
ρ	0.01
ϵ	0.9
τ	10^3
r	0.001
N_j	1000, $\forall j$

Note that $\{j_{(n)}\}$ is the sub-sequence of $\{j\}$ when recursion (57) is executed

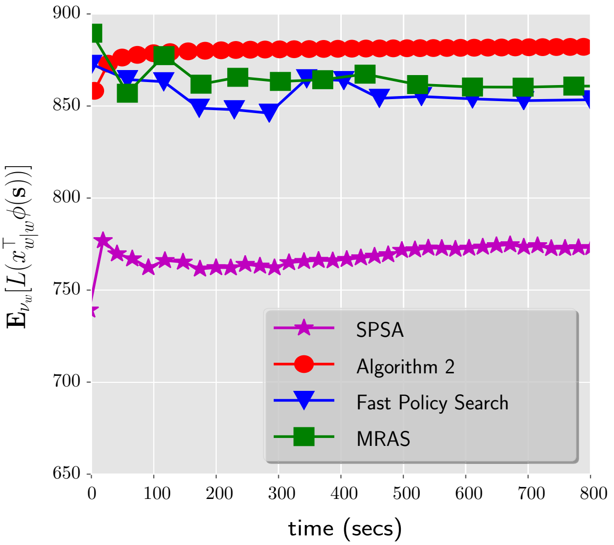


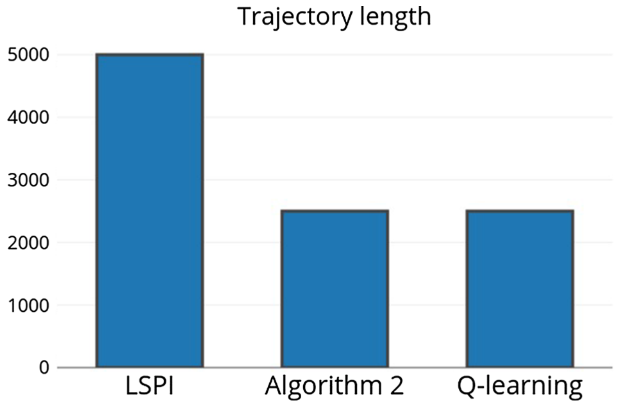
Fig. 10 Plot of the results obtained in the random MDP experiment. Here also, x-axis is time in secs relative to the start of the algorithm

4.5 Exegesis of the experiments

In this section, we summarize the inferences drawn from the above experiments:

(1) The proposed algorithm performed better than the state-of-the-art methods without compromising on the rate of convergence. The choice of the underlying behaviour policy indeed influenced this improved performance. Note that to labour high quality solutions, the choice of the behaviour policy is pivotal. In Experiment 1, we considered a uniform policy, where every action is equally likely to be chosen for each state in $\mathbb{S}$. The results obtained in that experiment are quite promising, since, by only utilizing a uniform behaviour policy, we were able to grind out superior quality solutions. One has to justify the results to add credibility, considering the fact that LSPI is shown to produce optimal policy given a generative model. Note that in the original LSPI paper, we find that the LSPI method utilizes a sample trajectory provided in the form of tuples $\{(s_i, a_i, r_i, s'_i)\}_{i \in \mathbb{N}}$, where s_i and a_i are drawn uniformly randomly from $\mathbb{S}$ and $\mathbb{A}$ respectively, while s'_i is the transitioned state given s_i and a_i by following the underlying transition dynamics of the MDP and r_i is the immediate

reward for that transition. One can immediately see that the information content required to generate such a trajectory is equivalent to that of maintaining a generative model.



Further, in Lagoudakis and Parr (2003), where LSPI is being empirically evaluated, we find that a trajectory length of 5000 is being used in the 20-state chain walk to obtain optimal performance. However, in our experiment (Experiment 1) with 450 states, we only consider a trajectory length of 5000 for LSPI and hence obtain the sub-optimal performance. But, one should also consider the fact that the behaviour policy utilized by our algorithm in the same experiment is uniform (no prior information about the MDP is being availed) and the trajectory length is only half of that of LSPI. Now, regarding the performance of Q-learning, we know [from Theorem 1 of Maei et al. (2010)] that the method can only provide sub-optimal solutions.

In Experiments 2, 3 and 4, we surmised the behaviour policy based on more than a passable knowledge of the MDP. To make the comparison unbiased (since our algorithm utilized prior information about the MDP to induce the behaviour policy), in the algorithms (MRAS, fast policy search and SPSA) to which our method is being compared, we employed the more accurate on-policy approximation which requires the generative model. This is contrary to our method, where off-policy approximation is tried. Our algorithm exhibited as good a performance as the state-of-the-art methods in the cart-pole experiment and noticeably the finest performance in the actuated pendulum experiment. This is regardless of the fact that our algorithm is primarily designed for the discrete, finite MDP setting, while the cart-pole experiment and the actuated pendulum experiment are MDPs with continuous state and action spaces. The suboptimal performance of the fast policy search and MRAS is primarily attributed to the insufficient sample size. But the underlying computing machine which we consider for the experiments is a 64-bit Intel i3 processor with 4GB of memory. Because of these limited resources, there is a finite limit to which the sample size can be scaled. This illustrates the effectiveness of our approach on a resource restricted setting. Now regarding the random MDP experiment, the performance of our algorithm is on par (in fact superior) to the state-of-the-art schemes.

(2) The significance of these results is further strengthened by the fact that all the baseline algorithms considered in the experiments have access to the generative-model and the outcome depicted above is obtained after processing a bevy of sample trajectories. This is contrary to our method where such a privilege is not conferred.

(3) The algorithm does not seem to be heavily dependent on the discount factor γ. To corroborate the claim, we show here the performance of the algorithm for two different, yet

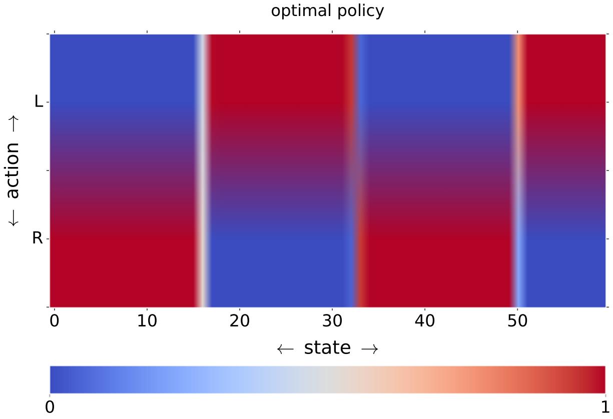


Fig. 11 The schematic diagram of the optimal policy generated by Algorithm 2 for the chain walk MDP with $|\mathbb{S}| = 60$, $\mathbb{A} = \{L, R\}$ and the discount factor $\gamma = 0.01$

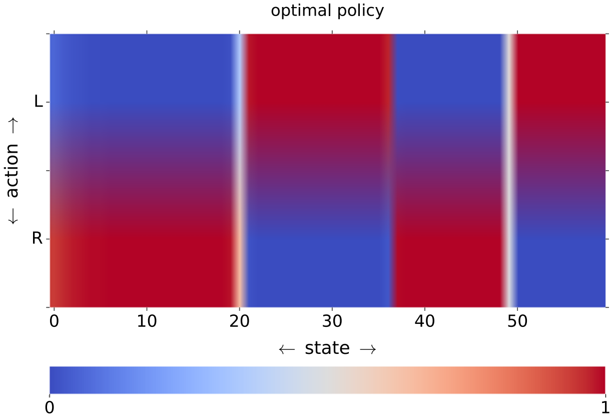


Fig. 12 The schematic diagram of the optimal policy generated by Algorithm 2 for the chain walk MDP with $|\mathbb{S}| = 60$, $\mathbb{A} = \{L, R\}$ and the discount factor $\gamma = 0.99$.

extreme values of γ, i.e., for $\gamma \in \{0.01, 0.99\}$ on the chain walk MDP with 60 states. Here, only the transitions to states 20 and 40 incur a positive cost, while the rest are null transitions. The optimal policies generated by our algorithm in the two cases are shown in Figs. 11 and 12 respectively. As one can observe, for $\gamma = 0.99$, the window around state 20 is wider than that for $\gamma = 0.01$. This is the expected behaviour since the discount factor controls the relative weights of future transitions while evaluating the discounted value function. However, note that this is not the case with regards to state 40. This lack of accuracy in the final third is primarily due to the fact that the behaviour policy we consider in this setting has its stationary distribution heavily concentrated on the first half of the state space. This particular scenario thus also illustrates the dependency of behaviour policy on the accuracy of the solution generated by our algorithm. This is indeed revealed in Theorem 3. To exemplify it further, we show here how the relative frequency of the states in the given trajectory generated using the behaviour policy determines the accuracy of the solution of our algorithm. Remember that the relative frequency of the states in the sample trajectory is indeed decided by the stationary distribution of the Markov chain induced by the behaviour policy. The results are shown in Figs. 13 and 14.

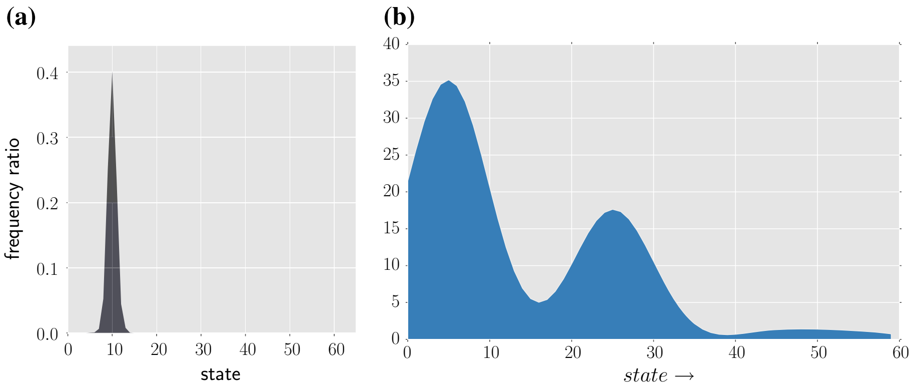


Fig. 13 **a** Frequency ratio of the states in the sample trajectory. **b** Optimal value function generated by Algorithm 2. The frequency ratio of a particular state in the sample trajectory is defined as the ratio of the number of occurrences of that state in the sample trajectory to the total number of state transitions in the sample trajectory. For an ergodic Markov chain, this ratio will eventually converge to is stationary distribution. In this particular example, observe that the accuracy of the value function is better for states whose relative frequency is good

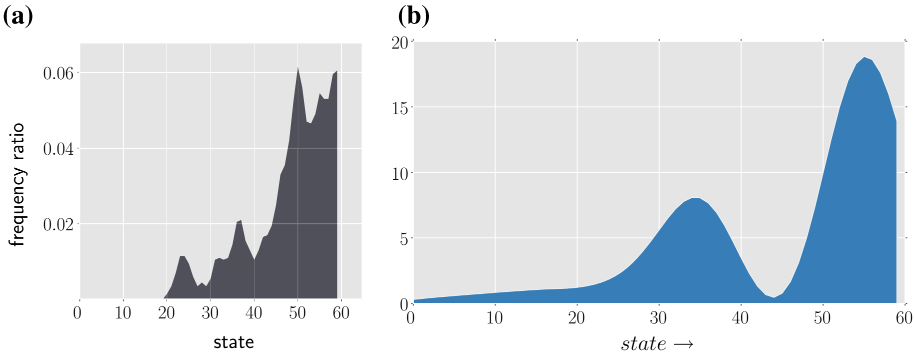


Fig. 14 **a** Frequency ratio of the states in the sample trajectory. **b** Optimal value function generated by Algorithm 2. In this setting, the relative frequency is better on the right half of the state space and the value function also seems to be more accurate in that region

(4) Finally, in the experiments, we found that the parameter which required the highest tuning is β_j which is also intuitive since β_j controls most of the stochastic recursions. The other parameters required minimum tuning with almost all of them taking common values.

4.6 Data efficiency

Here, we compare the efficiency of our algorithm with respect to the state-of-the-art algorithms. To measure the efficiency, we consider two benchmarks: *system configuration count* and *memory usage*. The system configuration count denotes the number of times the algorithm queries the generative model of the MDP with a policy to obtain sample trajectories. Memory usage denotes the average real time memory consumed by the algorithms. The results are shown in Fig. 15. The performance of our algorithm with regard to the above benchmarks is commendable.

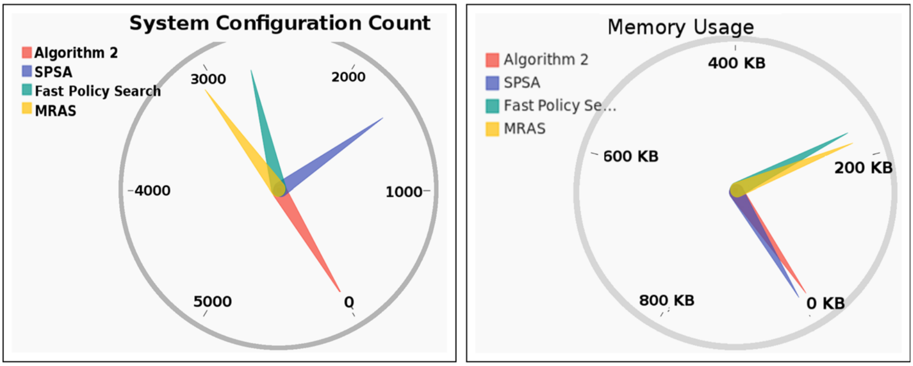


Fig. 15 Efficiency comparison of Algorithm 2 w.r.t. the state-of-the-art methods.

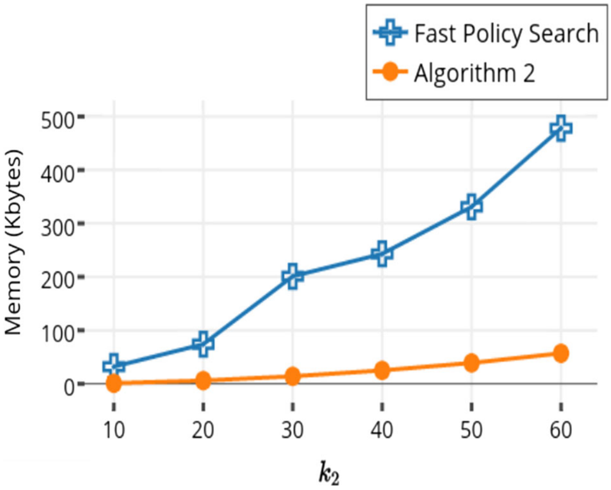


Fig. 16 Memory usage w.r.t. k_2

We also compare here the average memory usage of the fast policy search algorithm and our algorithm with respect to k_2 which is the dimension of the policy space. The results are shown in Fig. 16. The illustration shows that memory usage of our algorithms almost remains constant, however fast policy search is very sensitive to the parameter k_2.

This non-dependency of our algorithm on the dimension of the policy space has a real pragmatic advantage since, as a result of this, our algorithm can be applied to very large and complex MDPs with wider policy spaces where fast policy search and MRAS might become intractable.

Another advantage of our approach is the application on legacy systems. In such systems, the information on the dynamics of the system in the form of bits or bytes or paper might be hard to find. However, human experience through long time interaction with the system is available in most cases. Utilizing this human experience to develop a generative model of the system might be hard, however using it to find a behaviour policy which can give average performance is more plausible, and which in turn can be exploited using our algorithm to find an optimal policy.

5 Conclusion

We presented an algorithm which solves the modified control problem in a model free MDP setting. We showed its convergence to the global optimal policy relative to the choice of the behaviour policy. The algorithm is data efficient, robust, stable as well as computationally and storage efficient. Using an appropriately chosen behaviour policy, it is also seen to consistently outperform or is competitive against the current state-of-the-art (both) off-policy and on-policy methods.

References

Alon, G., Kroese, D. P., Raviv, T., & Rubinstein, R. Y. (2005). Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*, *134*(1), 137–151.

Antos, A., Szepesvári, C., & Munos, R. (2007). Value-iteration based fitted policy iteration: Learning with a single trajectory. In *2007 IEEE international symposium on approximate dynamic programming and reinforcement learning* (pp. 330–337).

Antos, A., Szepesvári, C., & Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, *71*(1), 89–129.

Bagnell, J. A., & Schneider, J. G. (2001). Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings 2001 ICRA. IEEE international conference on robotics and automation*, vol. *2* (pp. 1615–1620).

Balleine, B. W., & Dickinson, A. (1998). Goal-directed instrumental action: Contingency and incentive learning and their cortical substrates. *Neuropharmacology*, *37*(4), 407–419.

Barreto, A. D. M. S., Pineau, J., & Precup, D. (2014). Policy iteration based on stochastic factorization. *Journal of Artificial Intelligence Research*, *50*, 763–803.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, *72*(1), 81–138.

Baxter, J., & Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control* (Vol. 1). Belmont, MA: Athena Scientific.

Bertsekas, D. P., & Castanon, D. A. (1989). Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control*, *34*(6), 589–598.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., & Lee, M. (2009). Natural actor-critic algorithms. *Automatica*, *45*(11), 2471–2482.

Borkar, V. S. (2008). *Stochastic approximation*. Cambridge: Cambridge University Press.

Chang, H. S., Hu, J., Fu, M. C., & Marcus, S. I. (2013). *Simulation-based algorithms for Markov decision processes*. Berlin: Springer.

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

Deisenroth, M., & Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning (ICML)* (pp. 465–472).

de Boer, P. T. (2000). Analysis and efficient simulation of queueing models of telecommunication systems. Centre for Telematics and Information Technology University of Twente.

Ertin, E., Dean, A. N., Moore, M. L., & Priddy, K. L. (2001). Dynamic optimization for optimal control of water distribution systems. *Applications and Science of Computational Intelligence IV*, *4390*, 142–149.

Feinberg, E. A., & Shwartz, A. (2012). *Handbook of Markov decision processes: Methods and applications*. Berlin: Springer.

Fracasso, P., Barnes, F., & Costa, A. (2014). Optimized control for water utilities. *Procedia Engineering*, *70*, 678–687.

Glynn, P. W., & Iglehart, D. L. (1989). Importance sampling for stochastic simulations. *Management Science*, *35*(11), 1367–1392.

Helvik, B. E., & Wittner, O. (2001). Using the cross-entropy method to guide/govern mobile agents path finding in networks. In *International Workshop on Mobile Agents for Telecommunication Applications* (pp. 255–268). Springer.

Higham, N. J. (1994). A survey of componentwise perturbation theory in numerical linear algebra. In W. Gautschi (Ed.), *Mathematics of computation 1943–1993: A half century of computational mathematics*

(Proceedings of Symposia in Applied Mathematics) (Vol. 48, pp. 49–77). Providence, RI: American Mathematical Society.

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., Hu, P., & Chang, H. S. (2012). A stochastic approximation framework for a class of randomized optimization algorithms. *IEEE Transactions on Automatic Control*, *57*(1), 165–178.

Ikonen, E., & Bene, J. (2011). Scheduling and disturbance control of a water distribution network. *IFAC Proceedings Volumes*, *44*(1), 7138–7143.

Joseph, A. G., & Bhatnagar, S. (2016a). A randomized algorithm for continuous optimization. In *Winter simulation conference, WSC 2016, Washington, DC, USA, December 11–14* (pp. 907–918).

Joseph, A. G., & Bhatnagar, S. (2016b). A cross entropy based stochastic approximation algorithm for reinforcement learning with linear function approximation. CoRR **abs/1207.0016**.

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

Keith, J., & Kroese, D. P. (2002). Rare event simulation and combinatorial optimization using cross entropy: Sequence alignment by rare event simulation. In *Proceedings of the 34th conference on winter simulation: Exploring new frontiers, winter simulation conference* (pp. 320–327).

Koller, D., & Parr, R. (2000). Policy iteration for factored MDPs. In *Proceedings of the sixteenth conference on uncertainty in artificial intelligence* (pp. 326–334). Morgan Kaufmann Publishers Inc.

Konda, V. R., & Tsitsiklis, J. N. (2003). Actor-critic algorithms. *SIAM journal on Control and Optimization*, *42*(4), 1143–1166.

Kroese, D. P., Porotsky, S., & Rubinstein, R. Y. (2006). The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, *8*(3), 383–407.

Kumar, P., & Lin, W. (1982). Optimal adaptive controllers for unknown Markov chains. *IEEE Transactions on Automatic Control*, *27*(4), 765–774.

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

Lee, S. W., Shimojo, S., & O'Doherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, *81*(3), 687–699.

Maei, H. R., Szepesvári, C., Bhatnagar, S., & Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *Proceedings of the 27th international conference on machine learning (ICML)* (pp. 719–726).

Mannor, S., Rubinstein, R. Y., & Gat, Y.(2003). The cross entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning (ICML)* (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.

Moore, A. W., & Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, *13*(1), 103–130.

Mühlenbein, H., & Paass, G. (1996). From recombination of genes to the estimation of distributions i. Binary parameters. In *International conference on parallel problem solving from nature* (pp. 178–187). Springer.

O'Doherty, J. P., Lee, S. W., & McNamee, D. (2015). The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences*, *1*, 94–100.

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855.

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley.

Rubinstein, R. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, *1*(2), 127–190.

Rubinstein, R. Y. (2002). Cross-entropy and rare events for maximal cut and partition problems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *12*(1), 27–53.

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.

Sato, M., Abe, K., & Takeda, H. (1982). Learning control of finite Markov chains with unknown transition probabilities. *IEEE Transactions on Automatic Control*, *27*(2), 502–505.

Sato, M., Abe, K., & Takeda, H. (1988). Learning control of finite Markov chains with an explicit trade-off between estimation and control. *IEEE Transactions on Systems, Man, and Cybernetics*, *18*(5), 677–684.

Singh, S. P., & Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, *22*(1–3), 123–158.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*(3), 332–341.

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). *Reinforcement learning: An introduction*. Cambridge: MIT Press.

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.

Varga, R. S. (1976). On diagonal dominance arguments for bounding $\|A^{-1}\|_\infty$. *Linear Algebra and its Applications*, *14*(3), 211–217.

Wang, B., & Enright, W. (2013). Parameter estimation for ODEs using a cross-entropy approach. *SIAM Journal on Scientific Computing*, *35*(6), A2718–A2737.

Watkins, C. J. C. H. (1989). Learning from delayed rewards. Ph.D. Thesis, University of Cambridge England.

Xue, J. (1997). A note on entrywise perturbation theory for Markov chains. *Linear Algebra and its Applications*, *260*, 209–213.

Yu, H. (2012). Least squares temporal difference methods: An analysis under general conditions. *SIAM Journal on Control and Optimization*, *50*(6), 3310–3343.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Proceedings of the conference on computational learning theory*.

Zhou, E., Bhatnagar, S., Chen, X. (2014). Simulation optimization via gradient-based stochastic search. In *Proceedings of the 2014 winter simulation conference* (pp. 3869–3879). IEEE Press.

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.