Noname manuscript No.
(will be inserted by the editor)

Stochastic approximation for speeding up LSTD (and LSPI)

Prashanth L.A.[†] · **Nathaniel Korda**[♯] · **Rémi Munos**[†]

Abstract We propose a stochastic approximation (SA) based method with randomisation of samples for policy evaluation using the least squares temporal difference (LSTD) algorithm. Our method results in an $O(d)$ improvement in complexity in comparison to regular LSTD, where d is the dimension of the data. We provide convergence rate results for our proposed method, both in high probability and in expectation. Moreover, we also establish that using our scheme in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function and hence a low-complexity LSPI variant that uses our SA based scheme has the same order of the performance bounds as that of regular LSPI. These rate results coupled with the low complexity of our method make it attractive for implementation in *big data* settings, where d is large. Furthermore, we analyse a similar low-complexity alternative for least squares regression and provide finite-time bounds there. We demonstrate the practicality of our method for LSTD empirically by combining it with the LSPI algorithm in a traffic signal control application. We also conduct another set of experiments that combines the SA based low-complexity variant for least squares regression with the LinUCB algorithm for contextual bandits, using the large scale news recommendation dataset from Yahoo.

1 Introduction

Several machine learning problems involve solving a linear system of equations from a given set of training data. In this paper we consider the problem of policy evaluation in reinforcement learning (RL). The objective here is to estimate the value function V^π of a given policy π. Temporal difference (TD) methods are well-known in this context, and they are known to converge to the fixed point $V^\pi = \mathcal{T}^\pi(V^\pi)$, where $\mathcal{T}^\pi$ is the Bellman operator (see Section 3.1 for a precise definition).

The TD algorithm stores an entry representing the value function estimate for each state, making it computationally difficult to implement for problems with large state spaces. A

[†]INRIA Lille - Nord Europe, Team SequeL, FRANCE.
E-mail: {prashanth.la, remi.munos}@inria.fr
[♯]Oxford University, UNITED KINGDOM.
E-mail: nathaniel.korda@eng.ox.ac.uk

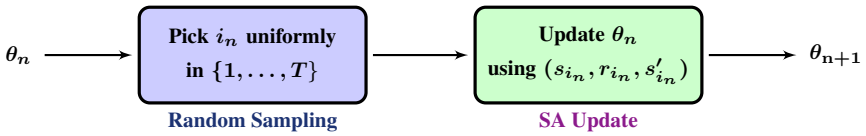


Fig. 1: Overall flow of the fLSTD-SA algorithm.

popular approach to alleviate this curse of dimensionality is to parameterize the value function using a linear function approximation architecture. For every s in the state space $\mathcal{S}$, we approximate $V^\pi(s) \approx \theta^\intercal \phi(s)$, where $\phi(\cdot)$ is a d-dimensional feature vector with $d << |\mathcal{S}|$, and θ is a tunable parameter. The function approximation variant of TD [39] is known to converge to the fixed point of $\Phi\theta = \Pi\mathcal{T}^\pi(\Phi\theta)$, where Π is the orthogonal projection onto the space within which we approximate the value function, and Φ is the feature matrix that characterises this space. For a detailed treatment of this subject matter, the reader is referred to the classic textbooks [5, 34].

Batch reinforcement learning is a popular paradigm for policy learning. Here, we are provided with a (usually) large set of state transitions $\mathcal{D} := \{(s_i, r_i, s'_i), i = 1, \ldots, T)\}$ obtained by simulating the underlying Markov decision process (MDP). For every $i = 1, \ldots, T$, the 3-tuple (s_i, r_i, s'_i) corresponds to a transition from state s_i to s'_i and the resulting reward is denoted by r_i. The objective is to learn an *approximately optimal* policy from this set. LSPI [20] is a well-known batch RL algorithm in this context, and it is based on the idea of policy iteration. A fundamental component of LSPI is LSTD [8] for policy evaluation, which is introduced next.

LSTD estimates the fixed point of $\Pi\mathcal{T}^\pi$, for a given policy π, using empirical data $\mathcal{D}$. The LSTD estimate is given as the solution to

$$\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T, \tag{1}$$

$$\text{where } \bar{A}_T = \frac{1}{T}\sum_{i=1}^{T}\phi(s_i)(\phi(s_i) - \beta\phi(s'_i))^\intercal \quad \text{and} \quad \bar{b}_T = \frac{1}{T}\sum_{i=1}^{T} r_i\phi(s_i).$$

We consider a special variant of LSTD called pathwise LSTD, proposed in [21]. The idea behind pathwise LSTD is to **(i)** have the dataset $\mathcal{D}$ created using a sample path simulated from the underlying MDP for the policy π and **(ii)** set $s'_T = 0$ while computing $\bar{A}_T$ defined above. The latter setting ensures the existence of the LSTD solution $\hat{\theta}_T$ under the condition that the family of features on the data set $\mathcal{D}$ are linearly independent. It is possible to make other minor modifications of the dataset or regularize the problem in order to ensure existence of $\hat{\theta}_T$, but this is beyond the scope of this work.

Our primary focus in this work is to solve the LSTD system in a computationally efficient manner. Computing the inverse of the matrix $\bar{A}_T$ is computationally expensive, especially when d is large. Indeed, assuming that the features $\phi(s_i)$ evolve in a compact subset of $\mathbb{R}^d$, the complexity of the above approach is $O(d^2T)$, where $\bar{A}_T^{-1}$ is computed iteratively using the Sherman-Morrison lemma. On the other hand, if we employ the Strassen algorithm or the Coppersmith-Winograd algorithm for computing $\bar{A}_T^{-1}$, the complexity is of the order $O(d^{2.807})$ and $O(d^{2.375})$, respectively, in addition to $O(d^2T)$ complexity for computing $\bar{A}_T$.

Fast LSTD: From the above discussion, it is evident that LSTD scales poorly with the number of features, making it inapplicable for large datasets with many features. A common trick, used in practice to alleviate this problem in high dimensions, is to replace the inversion

of the $\bar{A}_T$ matrix by the following iterative procedure that performs a fixed point iteration (see Figure 1 for an illustration): Set θ_0 arbitrarily and update

$$\theta_n = \theta_{n-1} + \gamma_n \left(r_{i_n} + \beta \theta_{n-1}^{\mathsf{T}} \phi(s'_{i_n}) - \theta_{n-1}^{\mathsf{T}} \phi(s_{i_n}) \right) \phi(s_{i_n}), \tag{2}$$

where each i_n is chosen uniformly at random from the set $\{1, \ldots, T\}$ and γ_n are step-sizes that satisfy standard stochastic approximation conditions (see (A1) in Section 4). The random sampling is sufficient to ensure convergence to the LSTD solution. The advantage of the above scheme is that it incurs a lower computational cost in comparison to the traditional LSTD solvers.

From a theoretical standpoint, the scheme (2) comes under the purview of stochastic approximation (SA). Stochastic approximation is a well-known technique that was originally proposed for finding zeroes of a nonlinear function in the seminal work of Robbins and Monro [31]. Iterate averaging is a standard approach to accelerate the convergence of SA schemes and was proposed independently in [32] and [28]. Non asymptotic bounds for Robbins Monro schemes have been provided in [11] and extended to incorporate iterate averaging in [10]. The reader is referred to [19] for a textbook introduction to SA.

Improving the complexity of TD-like algorithms is a popular line of research in RL. The popular Computer Go setting, with dimension $d = 10^6$, [33] and several practical application domains (e.g. transportation, networks) involve high-feature dimensions. Moreover, considering that linear function approximation is effective with a large number of features, our $O(d)$ improvement in complexity of LSTD by employing SA is meaningful. For other algorithms treating this complexity problem, see GTD [35], GTD2 [36], iLSTD [12] and the references therein. In particular, iLSTD is suitable for settings where the features admit a sparse representation.

Our contributions: In the context of improving the complexity of LSTD, our contributions can be summarised as follows:

Finite time bounds

- We show that our algorithm (2) converges to the pathwise LSTD solution at the optimal rate of $O(n^{-1/2})$ in expectation (see Theorem 4.2 in Section 4).
- By projecting the iterate (2) onto a compact/convex subset of $\mathbb{R}^d$, we are able to establish high probability bounds on the error $\left\| \theta_n - \hat{\theta}_T \right\|_2$. In particular, we show that, with probability $1 - \delta$, the fast LSTD iterate θ_n constructs an ϵ-approximation of the corresponding pathwise LSTD solution with $O(d \ln(1/\delta)/\epsilon^2)$ complexity, irrespective of the number of batch samples T.

The above rate results are for a step-size choice that is inversely proportional to the number of iterations of (2) and also require the knowledge of the minimum eigenvalue of $\bar{A}_T$. We overcome the latter dependence on the knowledge of the minimum eigenvalue through iterate averaging.

Performance bound We establish that using the SA based scheme in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function (see Theorem 4.4 in Section 4). Moreover, using this SA based scheme (2) in place of LSTD in the LSPI algorithm does not impact its convergence rate either (see Theorem 6.1).

Iterate averaging We also investigate the rates when larger stepsizes ($\Theta(n^{-\alpha})$ where $\alpha \in (1/2, 1)$) are used in conjunction with averaging of the iterate θ_n, i.e., the well known Polyak-Ruppert averaging scheme. The rates obtained in high probability are of the order $O(n^{-\alpha/2})$, with the added advantage that the rate is independent of the choice

of c in the step-sizes (see Theorem 5.1 in Section 5). Further, with iterate averaging the complexity of the algorithm stays at $O(d)$ per iteration as before.

Simulation experiments We illustrate these results in the context of a traffic control application. We test a variant of LSPI which uses the SA based scheme (2) in place of LSTD. In particular, for the experiments we employ step-sizes that were used to derive the finite-time bounds (see Theorem 4.2). We demonstrate that running SA based LSTD for a short number of iterations (~ 500) on big-sized problems with feature dimension ~ 4000, one gets a performance that is almost as good as regular LSTD at a significantly lower computational cost (see Figure 3 in Section 8).

Least squares regression and SGD: Many practical machine learning algorithms require computing the least squares solution at each iteration in order to make a decision. As in the case of LSTD, classic least squares solution schemes such as Sherman-Morrison lemma are of complexity of the order $O(d^2)$. A practical alternative is to use a SA based iterative scheme that is of the order $O(d)$. Such SA based schemes when applied to the least squares parameter estimation context are well known in the literature as stochastic gradient descent (SGD) algorithms.

We also analyse the low-complexity SGD alternative for the classic least squares parameter estimation problem. Using the same template as for the results of the SA variant of LSTD, we derive finite-time bounds, both in high probability as well as in expectation for the tracking error $\|\theta_n - \hat{\theta}_T\|_2$. Here θ_n is the SGD iterate, while $\hat{\theta}_T$ is the least squares solution (see Section 9 for a detailed description). We describe a fast variant of the LinUCB [22] algorithm for contextual bandits, where the SGD iterate is used in place of the least squares solution. We demonstrate the empirical usefulness of the SGD based LinUCB algorithm using the large scale news recommendation dataset from Yahoo [40]. We observe that, using the step-size suggested by our bounds (see Theorem 9.2), the SGD based LinUCB algorithm exhibits low tracking error, while providing significant computational gains.

The rate results coupled with the low complexity of our schemes, in the context of LSTD as well as least squares regression, make them more amenable to practical implementation in the canonical *big data* settings, where the dimension d is large. This is amply demonstrated in our applications in transportation and recommendation systems domains, where we establish that SA based LSTD and SGD perform almost as well as regular LSTD and regression solvers, albeit with much less computation (and with less memory). Note that the empirical evaluations are for higher level machine learning algorithms - least squares policy iteration (LSPI) [20] and linear bandits [9, 22], which use LSTD and regression in their inner loops.

The rest of the paper is organized as follows: In Section 2, we discuss related work. In Section 3 we present the fast LSTD algorithm based on stochastic approximation and in Section 4 we provide the non-asymptotic bounds for this algorithm. Next, in Section 7, we provide outlines for the proof and derivation of rates. In Section 5, we analyse a variant of our algorithm that incorporates iterate averaging. In Section 6, we describe a variant of LSPI that uses the SA based scheme (2) in place of LSTD. We provide experiments on a traffic signal control application in Section 8. In Section 9, we provide extensions to solve the problem of least squares regression and also a set of experiments that tests a variant of the LinUCB algorithm using a SA based subroutine for least squares regression. Finally, in Section 11 we provide the concluding remarks.

2 Literature review

2.1 Previous work related to LSTD

In Chapter 6 of [14], the authors establish that LSTD has the optimal asymptotic convergence rate, while [1] and [21] provide a finite time analysis for LSTD and also LSPI. Recent work in [37] derives sample complexity bounds for LSTD(λ). LSPE(λ) - an algorithm that is closely related to LSTD(λ) - is analyzed in [41]. The authors there provide asymptotic rate results for LSPE and show that it matches that of LSTD(λ). Also related is [27], where the authors study linear systems in general and as a special case, provide error bounds for LSTD that improve the dependence on the feature dimension. In this paper, we provide a finite time analysis of the fast LSTD algorithm (2) proposed here, which in conjunction the finite time bounds for LSTD from [21] establish that our approximation to LSTD does not impact its overall convergence rate to the true value function.

A related contribution that is geared towards improving the computational complexity of LSTD is iLSTD [12]. However, the analysis for iLSTD requires that the feature matrix be sparse, while we provide finite-time bounds for our fast LSTD algorithm without imposing sparsity on the features.

A related line of previous work are GTD [35] and GTD2 [36], which are temporal difference learning algorithms with an update iteration that can be viewed as gradient descent. This class of algorithms operate in the online setting as regular TD with function approximation, but with the advantage that GTD/GTD2 are provably convergent to the TD fixed point even when the policy used for collecting samples differs from the policy being evaluated - the so-called *off-policy* setting. Recent work in [24] provides finite time analysis for the GTD algorithm. Unlike GTD-like algorithms, we operate in an offline setting with a batch of samples provided beforehand. LSTD is a popular algorithm here, but has a bad dependency in terms of computational complexity on the feature dimension and we bring this down from $O(d^2)$ to $O(d)$ by running an algorithm that closely resembles TD on the batch of samples and the latter algorithm is shown to retain the convergence rate of LSTD.

To the best of our knowledge, efficient SA algorithms that approximate LSTD without impacting its rate of convergence to true value function, have not been proposed before in the literature. The high probability bounds that we derive for the SA based scheme do not directly follow from earlier work on LSTD algorithms. Concentration bounds for stochastic approximation schemes have been derived in [11]. While we use their technique for proving the high-probability bound on fast LSTD algorithm iterate (see Theorem 4.2), our analysis is more elementary, and we make all the constants explicit for the problem at hand. Moreover, in order to eliminate a possible exponential dependence of the constants in the resulting bound on the reciprocal of the minimum eigenvalue of $\bar{A}_T$, we depart from the argument in [11].

2.2 Previous work related to SGD

Finite time analysis of SGD methods have been provided in [2]. While the bounds in [2] are given in expectation, many machine learning applications require high probability bounds, which we provide for our case. Regret bounds for online SGD techniques have been given in [42, 13]: the gradient descent algorithm in [42] is in the setting of optimising the average of convex loss functions whose gradients are available, while that in [13] is for strongly convex loss functions. In comparison to previous work w.r.t. least squares regression, we highlight

the following differences:
(i) Earlier works on strongly convex optimization (cf. [13]) require the knowledge of the strong convexity constant in deciding the step-size. While one can regularize the problem to get rid of the step-size dependence on μ, it is not straightforward to choose the regularization constant. Notice that for SGD type schemes, one requires that the matrix $\bar{A}_T$ has a minimum positive eigenvalue μ. Equivalently, this implies that the original problem is regularized with $T\mu$. This may turn out to be too high a regularization and hence, it is desirable to have SGD get rid of this dependence without changing the problem itself. This is precisely what iterate-averaged SGD achieves, i.e., optimal rates both in high probability and expectation even for the un-regularized problem. To the best of our knowledge, there is no previous work that provides finite time bounds, both in high probability and in expectation, for iterate-averaged SGD.
(ii) Our analysis is for the classic SGD scheme that is anytime, whereas the epoch-GD algorithm in [13] requires the knowledge of the time horizon.
(iii) While the algorithm in [3] is shown to exhibit the optimal rate of convergence without assuming strong convexity, the bounds there are in expectation only. In contrast, for the special case of strongly convex functions, we derive high-probability bounds in addition to bounds in expectation. Furthermore, the bound in expectation from [2] is not optimal for a strongly convex function in the sense that the initial error (which depends on where the algorithm started) is not forgotten as fast as the rate that we derive.
(iv) On a minor note, our analysis is simpler since we work directly with least squares problems and we make all the constants explicit for the problems considered.

3 Fast LSTD using Stochastic Approximation *(fLSTD-SA)*

We propose here a stochastic approximation variant of the least squares temporal difference (LSTD) algorithm, whose iterates converge to the same fixed point as the regular LSTD algorithm, while incurring much smaller overall computational cost.

The algorithm, which we call fast LSTD through Stochastic Approximation (fLSTD-SA), is a simple stochastic approximation scheme with randomised samples. The results that we present establish that fLSTD-SA computes an ϵ-approximation to the LSTD solution $\hat{\theta}_T$ with probability $1-\delta$, while incurring a complexity of the order $O(d\ln(1/\delta)/\epsilon^2)$, irrespective of the number of samples T. In turn, this enables us to give a performance bound for the approximate value function computed by fLSTD-SA. A schema of fLSTD-SA is given in Figure 1, while Algorithm 1 gives the pseudocode.

Using our analysis to set the step sequence for fLSTD-SA requires using the knowledge of the minimum eigenvalue of $\bar{A}_T$ - a matrix made from the features used in the linear approximation (see assumption (A4) below). We present a variant of fLSTD-SA employing iterate averaging for which knowledge of this eigenvalue is not needed to obtain the optimal rate of convergence (see Section 5).

3.1 Background

Consider an MDP with (finite) state space $\mathcal{S}$, (finite) action space $\mathcal{A}$ and transition probabilities $p(s,a,s')$, $s,s' \in \mathcal{S}, a \in \mathcal{A}$. Let π be a stationary randomized policy, i.e., $\pi(s,\cdot)$ is a

distribution over $\mathcal{A}$, for any $s \in \mathcal{S}$. The value function V^π is defined by

$$V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \sum_{a\in\mathcal{A}} r(s_t,a)\pi(s_t,a) \mid s_0 = s\right], \tag{3}$$

where s_t denotes the state of the MDP at time t, $\beta \in (0,1)$ is the discount factor, and $r(s,a)$ denotes the instantaneous rewards obtained in state s with action a. The value function V^π can be expressed as the fixed point of the Bellman operator $\mathcal{T}^\pi$ defined by

$$\mathcal{T}^\pi(V)(s) := \sum_{a\in\mathcal{A}} \pi(s,a)\left(r(s,a) + \beta \sum_{s'} p(s,a,s')V(s')\right), \tag{4}$$

Function approximation: When the cardinality of $\mathcal{S}$ is huge, a popular approach is to parameterize the value function using a linear function approximation architecture, i.e., for every $s \in \mathcal{S}$, we approximate $V^\pi(s) \approx \phi(s)^\intercal\theta$, where $\phi(s)$ is a d-dimensional feature vector for state s with $d << |\mathcal{S}|$, and θ is a tunable parameter. Let Φ denote the feature matrix with rows $\phi(s)^\intercal, \forall s \in \mathcal{S}$. In the following, we describe the TD fixed point that involves the feature matrix Φ. Subsequently, we outline the pathwise LSTD approach where the matrix Φ is composed of the features corresponding to the states in an empirical dataset. By an abuse of notation, we shall use Φ to denote the feature matrix for TD as well as LSTD and the composition of Φ should be clear from the context.

TD learning: The well-known TD learning algorithm [5] attempts to find the fixed point of the operator $\Pi\mathcal{T}^\pi$ given by

$$\Phi\theta^* = \Pi\mathcal{T}^\pi(\Phi\theta^*), \tag{5}$$

where Π is the orthogonal projection onto $\mathcal{B} = \{\Phi\theta \mid \theta \in \mathbb{R}^d\}$, the vector subspace of $\mathbb{R}^{|S|}$ within which we want to approximate the value function V^π. It is easy to derive that $\Pi = \Phi(\Phi^\intercal\Psi\Phi)^{-1}\Phi^\intercal\Psi$, where Ψ is the diagonal matrix whose diagonal elements form the stationary distribution of the Markov chain associated with the policy π. The solution θ^* of (5) can be re-written as follows (cf. [4, Section 6.3]):

$$A\theta^* = b, \text{ where } A = \Phi^\intercal\Psi(I - \beta P)\Phi \text{ and } b = \Phi^\intercal\Psi\mathcal{R}, \tag{6}$$

where $P = [P(s,s')]_{s,s'\in\mathcal{S}}$ is the transition probability matrix with components $p(s,s') = p(s,\pi(s),s')$, $\mathcal{R}$ is the vector with components $\sum_{a\in\mathcal{A}} \pi(s,a)r(s,a)$, for each $s \in \mathcal{S}$, and Ψ the stationary distribution (assuming it exists) of the Markov chain for the underlying policy π.

LSTD and Pathwise LSTD: In the absence of knowledge of the transition dynamics P and stationary distribution Ψ, LSTD is an approach which can approximate the solution θ^* using a batch of samples obtained from the underlying MDP. In particular it requires a dataset, $\mathcal{D} = \{(s_i, r_i, s_i'), i = 1,\ldots,T)\}$, where each tuple in the dataset (s_i, r_i, s_i') represents a state-reward-next-state triple chosen by the policy. The LSTD solution approximates A, b, and θ^* with $\bar{A}_T$, $\bar{b}_T$ using the samples in $\mathcal{D}$ as follows:

$$\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T \tag{7}$$

$$\text{where } \bar{A}_T = \frac{1}{T}\sum_{i=1}^{T} \phi(s_i)(\phi(s_i) - \beta\phi(s_i'))^\intercal \quad \text{and} \quad \bar{b}_T = \frac{1}{T}\sum_{i=1}^{T} r_i\phi(s_i).$$

Denoting the current state feature $T \times d$-matrix by $\Phi := (\phi(s_1)^\intercal, \ldots, \phi(s_T))$, next state feature $T \times d$-matrix by $\Phi' := (\phi(s_1')^\intercal, \ldots, \phi(s_T'))$, and reward $T \times 1$-vector by $\mathcal{R} = (r_1, \ldots, r_T)^\intercal$, we can rewrite $\bar{A}_T$ and $\bar{b}_T$ as follows:

$$\bar{A}_T = \frac{1}{T}(\Phi^\intercal \Phi - \beta \Phi^\intercal \Phi'), \text{ and } \bar{b}_T = \frac{1}{T}\Phi^\intercal \mathcal{R}.$$

It is not clear whether $\bar{A}_T$ is invertible for arbitrary datasets, $\mathcal{D}$. One way to alleviate this is studied in [21].

The pathwise LSTD approach, proposed in [21], is an on-policy version of LSTD. It obtains samples, $\mathcal{D}$ by simulating a sample path of the underlying MDP using policy π, so that $s_i' = s_{i+1}$ for $i = 1, \ldots, T-1$. The dataset thus obtained is perturbed slightly by setting the feature of the next state of the last transition, $\phi(s_T')$, to zero. This perturbation, as suggested in [21], is crucial to ensure that the system of the equations that we solve as an approximation to (6) is well-posed. For the sake of completeness, we make this precise in the following discussion, which is based on Sections 2 and 3 of [21].

Define the empirical Bellman operator $\hat{T} : \mathbb{R}^T \to \mathbb{R}^T$ as follows: For any $y \in \mathbb{R}^T$,

$$(\hat{T}y)_i = \begin{cases} r_i + \beta y_{i+1}, & 1 \leq i < T \\ r_T, & i = T. \end{cases} \tag{8}$$

Let $\hat{\mathcal{R}}$ be a $T \times 1$ vector with entries r_i, $i = 1, \ldots, T$ and $(\hat{\mathcal{V}}y)_i = y_{i+1}$ if $i < n$ and 0 otherwise. Then, it is clear that $\hat{T}y = \hat{\mathcal{R}} + \beta \hat{\mathcal{V}} y$.

Let $\mathcal{G}_T := \{(\phi(s_1)^\intercal \theta, \ldots, \phi(s_T)^\intercal \theta)^\intercal \mid \theta \in \mathbb{R}^d\} \subset \mathbb{R}^T$. $\mathcal{G}_T$ is the vector sub-space of $\mathbb{R}^T$ within which pathwise LSTD approximates the true values of the value functions of the states $s_1, \ldots, s_T$, and it is the empirical analogue of $\mathcal{B}$. Define Φ to be a $T \times d$ matrix with rows $\phi(s_i)^\intercal, i = 1, \ldots, T$, where $\phi(s_i)$ is a d-dimensional feature vector corresponding to state s_i, for all $i = 1, \ldots, T$. It is easy to see that $\mathcal{G}_T = \{\Phi\theta \mid \theta \in \mathbb{R}^d\}$.

Let $\hat{\Pi}$ be the orthogonal projection onto $\mathcal{G}_T$ using the empirical norm, which is defined as follows: $\|f\|_T^2 := T^{-1} \sum_{i=1}^T f(s_i)^2$, for any function f. We now claim that $\hat{\Pi}\hat{T}$ is a contraction mapping because

$$\begin{aligned} \left\| \hat{\Pi}\hat{T}y - \hat{\Pi}\hat{T}z \right\|_T \leq & \left\| \hat{T}y - \hat{T}z \right\|_T \\ = & \beta \left\| \hat{\mathcal{V}}y - \hat{\mathcal{V}}z \right\|_T \\ \leq & \beta \left\| y - z \right\|_T . \end{aligned}$$

So, by the Banach fixed point theorem, there exists some $v^* \in \mathcal{G}_T$ such that $\hat{\Pi}\hat{T}v^* = v^*$.

Let us assume that the feature matrix Φ is full rank - an assumption that is standard in the analysis of TD-like algorithms and also beneficial in the sense that it ensures that the system of equations we attempt to solve is well-posed[1]. Then, it is easy to see that there exists a unique $\hat{\theta}_T$ such that $v^* = \Phi\hat{\theta}_T$. Moreover, replacing $\bar{A}_T$ in (7) with

$$\bar{A}_T = \frac{1}{T}\Phi^\intercal (I - \beta \hat{P})\Phi, \tag{9}$$

where $\hat{P}$ is a $T \times T$ matrix with $\hat{P}(i, i+1) = 1$ for $i = 1, \ldots, T-1$ and 0 otherwise, it is clear that $\hat{\theta}_T$ is the unique solution to (7).

[1] In [21], the authors do not make this assumption and hence, have to resort to a pseudo-inverse based solution for pathwise LSTD.

From the foregoing, it is evident that, using the definition (9), Φ being full rank implies that the minimum eigenvalue of $\bar{A}_T$ - denoted by $\mu := \lambda_{\min}(\bar{A}_T)$ - is positive. This can be seen as follows: Since Φ is full rank, the matrix $\frac{1}{T}\Phi^{\mathsf{T}}\Phi$ is positive definite. Moreover, $(I - \beta\hat{P})$ is invertible since $\beta < 1$ and $\hat{P}$ is a right stochastic matrix. Thus, $\lambda_{\min}(\bar{A}_T) > 0$ if Φ is full rank. The converse also holds, though it is not necessary for our analysis.

Remark 1 ***(Regular vs. Pathwise LSTD)*** For a large data set, $\mathcal{D}$, generated from a sample path of the underlying MDP for policy π, the difference in the matrix used as $\bar{A}_T$ in LSTD and pathwise LSTD is negligible. In particular, the difference in ℓ_2-norm of $\bar{A}_T$ composed with and without zeroing out the next state in the last transition of $\mathcal{D}$ can be upper bounded by a constant multiple of $\dfrac{1}{T}$. As mentioned earlier, zeroing out the next state in the last transition of $\mathcal{D}$ ensures that the matrix $\bar{A}_T$ is positive definite, making the system of equations in (7) well-posed.

As an aside, the SA based scheme that we propose (see (10) below) would work as a good approximation to LSTD, as long as one ensures that $\bar{A}_T$ is positive definite. Pathwise LSTD presents one approach to achieve the latter requirement and it is an interesting future research direction to derive other conditions that ensure $\bar{A}_T$ is positive definite.

3.2 Update rule and pseudocode for fLSTD-SA

The idea is to perform an incremental update that is similar to TD, except that the samples are drawn uniformly randomly from the dataset $\mathcal{D}$. Recall that the data set corresponds to those along a sample path simulated from the underlying MDP for a given policy π, i.e., $s'_i = s_{i+1}$, $i = 1, \ldots, T-1$ and $s'_T = 0$.

Starting with an arbitrary θ_0, we update the parameter θ_n as follows:

$$\theta_n = \Upsilon\left(\theta_{n-1} + \gamma_n\left(r_{i_n} + \beta\theta_{n-1}^{\mathsf{T}}\phi(s'_{i_n}) - \theta_{n-1}^{\mathsf{T}}\phi(s_{i_n})\right)\phi(s_{i_n})\right), \tag{10}$$

where each i_n is chosen uniformly randomly from the set $\{1, \ldots, T\}$. In other words, we pick a sample with uniform probability $1/T$ from the set $\mathcal{D} = \{(s_i, r_i, s'_i), i = 1, \ldots, T)\}$ and use it to perform a fixed point iteration in (10). The quantities γ_n above are *step sizes* that are chosen in advance and satisfy standard stochastic approximation conditions (see (A1) below). The operator Υ projects the iterate θ_n onto the nearest point in a convex and compact set $\mathcal{C} \subset \mathbb{R}^d$ such that, for any $\theta \in \mathcal{C}$, $\|\theta\|_2 \leq H$. The full pseudocode for fLSTD-SA is given in Algorithm 1. We assume in the following sections that the set $\mathcal{C}$ is large enough to include the LSTD solution $\hat{\theta}_T$.

Projection-free update: We also consider a variant of fLSTD-SA that does not include the projection operator and updates as follows:

$$\theta_n = \theta_{n-1} + \gamma_n\left(r_{i_n} + \beta\theta_{n-1}^{\mathsf{T}}\phi(s'_{i_n}) - \theta_{n-1}^{\mathsf{T}}\phi(s_{i_n})\right)\phi(s_{i_n}), \tag{11}$$

Technically, the projection operator Υ is not necessary to ensure asymptotic convergence nor is it required to bound the error $\|\theta_n - \hat{\theta}_T\|_2$ in expectation. However, we are unable to derive bounds in high probability without having the iterates explicitly bounded using Υ and it would be a interesting future research direction to get rid of this operator for the bounds in high probability.

Algorithm 1 fLSTD-SA

Input: Sample path based dataset $\mathcal{D} := \{(s_i, r_i, s_i'), i = 1, \ldots, T)\}$ such that $s_i' = s_{i+1}$, $i = 1, \ldots, T-1$ and $s_T' = 0$; a choice of step-size sizes, γ_k; a time horizon n.
Initialisation: Set θ_0.
Run:
for $k = 1 \ldots n$ **do**
Get random sample index: $i_k \sim U(\{1, \ldots, T\})$
Perform update iteration: $\theta_k = \Upsilon\left(\theta_{k-1} + \gamma_k \left(r_{i_k} + \beta\theta_{k-1}^\intercal \phi(s_{i_k}') - \theta_{k-1}^\intercal \phi(s_{i_k})\right)\phi(s_{i_k})\right)$
end for
Output: θ_n

4 Main results for fLSTD-SA

Map of the results:

Asymptotic convergence: Theorem 4.1 proves almost sure convergence of fLSTD-SA iterate θ_n to LSTD solution $\hat{\theta}_T$, with and without projection.

Error bounds: Theorem 4.2 provides finite time bounds both in high probability and in expectation for the error $\|\theta_n - \hat{\theta}_T\|_2$, where θ_n is given by (10). We require high probability bounds to qualify the rate of convergence of the approximate value function $\Phi\theta_n$ to the true value function, i.e., a variant of Theorem 1 in [21] for the case of fLSTD-SA.

Error bound (projected fLSTD): Theorem 4.3 provides finite time bounds in expectation for the error $\|\theta_n - \hat{\theta}_T\|_2$, where θ_n is given by (11), i.e., without projection.

Performance bound: Theorem 4.4 presents a performance bound for the special case when the dataset $\mathcal{D}$ comes from a sample path of the underlying MDP for the given policy π.

Note that the first three results above hold irrespective of whether the dataset $\mathcal{D}$ is based on a sample path or not. However, the performance bound is for a sample path dataset only and is used to illustrate that using fLSTD-SA in place of regular LSTD does not harm the overall convergence rate of the approximate value function to the true value function.

We state all the results in Sections 4.2–4.4 and provide detailed proofs of all the claims in Section 7. Also, all the results are by default for the projected version of fLSTD-SA, i.e., θ_n given by (10), and we explicitly qualify the results for the projection free fLSTD-SA variant.

4.1 Assumptions

We make the following assumptions for the analysis of fLSTD-SA:

(A1) The step sizes γ_n satisfy $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2 < \infty$.
(A2) Bounded features: $\|\phi(s_i)\|_2 \leq \Phi_{\max} < \infty$, for $i = 1, \ldots, T$.
(A3) Bounded rewards: $|r_i| \leq R_{\max} < \infty$ for $i = 1, \ldots, T$.
(A4) The matrix $\bar{A}_T$ is positive definite, so its smallest eigenvalue $\mu = \lambda_{\min}(\bar{A}_T) > 0$.

By working with bounded rewards and features, and with step sizes that satisfy standard stochastic approximation conditions, we ensure that the parameter θ remains stable, and hence that (10) converges. Theorem 4.1 makes this claim precise.

In the following sections, we present results for the generalized setting, i.e., the dataset $\mathcal{D}$ does not necessarily come from a sample path of the underlying MDP, but we assume (see (A4)) that the matrix $\bar{A}_T$ is positive definite. For pathwise LSTD, as discussed earlier, (A4) can be replaced by the following assumption:

(A4') The matrix Φ is full rank and hence, the minimum eigenvalue, μ, of the matrix $\bar{A}_T = T^{-1}\Phi^{\intercal}(I - \beta\hat{P})^{\intercal}\Phi$ is strictly greater than zero.

Note that the dataset is assumed to be fixed for all the results presented below and all the probabilities and expectations are taken over the random choices of points from the dataset, except when otherwise stated.

4.2 Asymptotic convergence

Theorem 4.1 *Under (A1)-(A4), the iterate $\theta_n \to \hat{\theta}_T$ a.s. as $n \to \infty$, where θ_n is given by either* (10) *or* (11) *and $\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T$.*

Proof See Section 7.2. ■

4.3 Finite time bounds

The main result that bounds the computational error $\|\theta_n - \hat{\theta}_T\|_2$ with explicit constants is given below.

Theorem 4.2 (*Error bound for iterates of fLSTD-SA*)
Under (A2)-(A4), choosing $\gamma_n = \frac{c_0 c}{(c+n)}$ such that $c_0 \in (0, \mu((1+\beta)^2\Phi_{\max}^4)^{-1}]$ and $c_0\mu c \in (1, \infty)$, we have, for any $\delta > 0$,

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1(n)}{\sqrt{n+c}} \quad \textit{and} \quad \mathbb{P}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2(n)}{\sqrt{n+c}}\right) \geq 1 - \delta, \tag{12}$$

where $K_1(n)$ and $K_2(n)$ are functions of order $O(1)$, defined by:

$$K_1(n) = \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2}{(n+c)^{(c_0 c\mu - 1)/2}} + \frac{2c_0 c(R_{\max} + (1+\beta)H\Phi_{\max})}{2c_0 c\mu - 1},$$
$$K_2(n) = 2c_0 c\sqrt{\frac{\log\delta^{-1}(R_{\max} + (1+\beta)\Phi_{\max})}{c_0 c\mu - 1}} + K_1(n).$$

Remark 2 ***(Eigenvalue dependence)*** We note that setting c such that $c_0 c\mu = \eta \in (1, \infty)$ we can rewrite the constants in Theorem 4.2 as:

$$K_1(n) = \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2}{\sqrt{(n+c)^{(\eta-1)}}} + \frac{2\eta}{(2\eta-1)\mu}(R_{\max} + (1+\beta)H\Phi_{\max}),$$
$$K_2(n) = 2\frac{\eta}{\mu}\sqrt{\frac{\log\delta^{-1}(R_{\max} + (1+\beta)H\Phi_{\max})}{(\eta-1)}} + K_1(n).$$

So both the bounds in expectation and high probability have a linear dependence on the reciprocal of μ.

Remark 3 ***(Influence of boundedness assumptions)*** Note also that the constant $(R_{\max} + (1+\beta)H\Phi_{\max})$ is nothing more than a bound on the size of the random innovations made by the algorithm at each time step.

Remark 4 ***(Regularization)*** To obtain the best performance from fLSTD-SA we need to know the value of μ. However, we can get rid of this dependency easily by explicitly regularising the problem. In other words, instead of the LSTD solution (7), we solve the following regularised problem:

$$\hat{\theta}_T^{reg} = (\bar{A}_T + \mu I)^{-1}\bar{b}_T \tag{13}$$

where μ is now a constant set in advance. The update rule for this variant is

$$\theta_n^{reg} = (1-\gamma_n\mu)\theta_{n-1} + \gamma_n\left(r_{i_n} + \beta\theta_{n-1}^{\mathsf{T}}\phi(s'_{i_n}) - \theta_{n-1}^{\mathsf{T}}\phi(s_{i_n})\right)\phi(s_{i_n}). \tag{14}$$

This algorithm retains all the properties of the non-regularized fLSTD-SA algorithm, except that it converges to the solution of (13) rather than to that of (7). In particular the conclusions of Theorem 4.2 hold without requiring assumption (A4), but measuring $\theta_n - \hat{\theta}_T^{reg}$, the error to the regularized fixed point $\hat{\theta}_T^{reg}$.

Using a slightly different proof technique, we are able to give a bound in expectation for the error of the non-projected fLSTD-SA. However, as mentioned earlier, this analysis does not seem to provide also a bound in high probability.

Theorem 4.3 **(*Expectation error bound for iterates of fLSTD-SA without projection*)** *Under (A2)-(A4), choosing* $\gamma_n = \frac{c_0 c}{(c+n)}$ *such that* $c_0 \in (0, \mu((1+\beta)^2\Phi_{\max}^2)^{-1}]$ *and* $c_0\mu c \in (1,\infty)$*, we have, for any* $\delta > 0$*,*

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1(n)}{\sqrt{n+c}} \tag{15}$$

where $K_1(n)$ *is a function of order* $O(1)$*, defined by:*

$$K_1(n) = \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2}{(n+c)^{(c_0 c\mu-1)/2}} + \frac{c_0 c\left(R_{\max} + (1+\beta)\left\|\hat{\theta}_T\right\|_2\right)}{2c_0\mu c - 1}.$$

Proof See Section 7.2.

4.4 Performance bound for dataset $\mathcal{D}$ from a sample path

We can combine our error bounds above with the performance bound results derived in [21] for LSTD and LSPI. The results in [21] are data-dependendent, in that they do not require any assumptions on the matrix $\bar{A}_T$, and instead give their bounds in terms of the minimum positive eigevalue of the matrix $T^{-1}\Phi^{\mathsf{T}}\Phi$. By contrast, our bounds are given under the assumption that this matrix does not have any zero eigenvalues. For this reason we introduce the following extra assumption:

(A4") The stopping time $N := \min\{t : \text{rank}(\bar{A}_t) = d\}$ is finite with probability 1.

The theorem below shows that, when datasets generated by the underlying MDP can be created to satisfy the assumption (A4), using fLSTD-SA in place of regular LSTD does not hurt the overall convergence rate of the LSTD based value function estimate to the true value function.

Theorem 4.4 (***Performance bound for sample path based dataset*** $\mathcal{D}$) *Let* $\tilde{v}_n := \Phi\theta_n$ *denote the approximate value function obtained after* n *steps of fLSTD-SA, and let* v *denote the true value function, evaluated at the states* $s_1, \ldots, s_T$ *along the sample path. Then, under the assumptions (A1)-(A3) and (A4"), and assuming the sample size* $T \geq N$*, with probability* $1 - 2\delta$ *(taken w.r.t. the random path sampled from the MDP and the randomisation in fLSTD-SA) we have*

$$\|v - \tilde{v}_n\|_T \leq \underbrace{\frac{\|v - \Pi v\|_T}{\sqrt{1-\beta^2}}}_{\textit{approximation error}} + \underbrace{\frac{\beta R_{\max}\Phi_{\max}}{(1-\beta)}\sqrt{\frac{d}{\mu}}\left(\sqrt{\frac{8\ln\frac{2d}{\delta}}{T}} + \frac{1}{T}\right)}_{\textit{estimation error}} + \underbrace{\frac{\Phi_{\max}K_2(n)}{\sqrt{n+c}}}_{\textit{computational error}} . \tag{16}$$

where $\|f\|_T^2 := T^{-1}\sum_{i=1}^T f(s_i)^2$*, for any function* f*.*

$$\mu = \lambda_{\min}(\Phi^\intercal(I - \beta\hat{P})\Phi) \geq (1-\beta)\lambda_{\min}(\Phi^\intercal\Phi)$$
$$\mu = \lambda_{\min}(\Phi^\intercal(I - \beta\hat{P})\Phi) \leq \lambda_{\min}(\Phi^\intercal\Phi)$$

Proof Note that (A4) satisfied whenever rank$(\bar{A}_T) = d$. The result therefore follows by combining Theorem 4.2 above with Theorem 1 of [21] using a triangle inequality. ■

Remark 5 ***(Collecting the sample set*** $\mathcal{D}$***)*** Suppose that the Markov chain induced by the underlying MDP and the policy π is irreducible, and that the feature set $\{\phi(s)\}_{s\in S}$ contains a linearly independent subset of size d. Then assumption (A4") is satisfied, and it is possible to define a stopping time $T \geq N$ which terminates with probability 1, that can be used to collect the sample set $\mathcal{D}$. Furthermore, under extra conditions on the underlying MDP, Lemma 4 from [21] can be used to remove the need for assumption (A4"). This lemma proves conditions under which strict positive definiteness of $\bar{A}_T$ can be guaranteed in high probability.

Remark 6 ***(Description of error terms)*** The approximation and estimation errors (first and second terms in the RHS of (16)) are artifacts of function approximation and least squares methods, respectively. The third term is a consequence of using fLSTD-SA in place of the LSTD. Setting $n = T$ in the above theorem, we observe that using our scheme in place of LSTD does not impact the rate of convergence of the approximate value function $\tilde{v}_n$ to the true value function v.

Remark 7 ***(Generalization bounds)*** While Theorem 4.4 holds for only states along the sample path $s_1, \ldots, s_T$, it is possible to generalize the result to hold for states outside the sample path. This approach has been adopted in [21] for regular LSTD and the authors there provide performance bounds over the entire state space assuming a stationary distribution exists for the given policy π and the underlying Markov chain is mixing fast (Lemma 4 from [21] mentioned in Remark 5 is used here also). In the light of the result in Theorem 4.4 above, it is straightforward to provide generalization bounds similar to Theorems 5 and 6 of [21] for fLSTD-SA as well and the resulting rates from these generalization bound variants for fLSTD-SA are the same as that for regular LSTD. We omit these obvious generalizations and refer the reader to Section 5 of [21] for further details.

5 Iterate Averaging

Iterate averaging is a popular approach for which it is not necessary to know the value of the constant μ (see (A4) in Section 4) to obtain the (optimal) approximation error of order $O(n^{-1/2})$. Introduced independently by Ruppert [32] and Polyak [28], the idea here is to use a larger step-size $\gamma_n := c_0 (c/(c+n))^{\alpha}$, and then use the averaged iterate $\bar{\theta}_{n+1} := (\theta_1 + \ldots + \theta_n)/n$ to approximate the LSTD solution. Here the quantities θ_n are just the iterates of the fLSTD-SA presented earlier.

Define $\bar{\theta}_{n+1} := (\theta_1 + \ldots + \theta_n)/n$. The following result bounds the the distance of the averaged iterate to the LSTD solution.

Theorem 5.1 (*Error Bound for iterate averaged fLSTD-SA*)
Under (A2)-(A3), choosing $\gamma_n = c_0 \left(\frac{c}{c+n}\right)^{\alpha}$, *with* $\alpha \in (1/2, 1)$ *and* $c, c_0 > 0$, *we have, for any* $\delta > 0$, *and any* $n > n_0 := \max\{\lfloor (2c_0(1+\beta^2)\Phi_{\max}^2)^{1/\alpha}/\mu - 1)c \rfloor, 0\}$

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1^{IA}(n)}{(n+c)^{\alpha/2}} \text{ and } \mathbb{P}\left(\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2^{IA}(n)}{(n+c)^{\alpha/2}}\right) \geq 1-\delta, \qquad (17)$$

where, writing $C = \sum_{n=1}^{\infty} \exp\left(-c_0 \mu c^{\alpha} (n+c)^{1-\alpha}\right) (< \infty)$,

$$K_1^{IA}(n) := \left(\|\theta_{n_0} - \theta_T\|_2 + e + \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}} \exp\left(\frac{2\alpha}{1-\alpha}\right)\right) \frac{C}{(n+c)^{1-\frac{\alpha}{2}}}$$
$$+2\left(R_{\max} + (1+\beta) H \Phi_{\max}\right) c^{\alpha} c_0 \left(c_0 \mu c^{\alpha}\right)^{-\alpha \frac{1+2\alpha}{2(1-\alpha)}},$$
$$\text{and } K_2^{IA}(n) := \frac{4\sqrt{\log \delta^{-1}}}{\mu^2 c_0^2} \frac{\frac{1}{\mu}\left\{2^{\alpha} + \left\lceil \frac{2\alpha}{c^{\alpha}} + \frac{2\alpha}{1-\alpha} \right\rceil\right\}}{(n+c)^{(1-\alpha)/2}} + K_1^{IA}(n).$$

Proof See Section 7.4.

From the above, it is evident that the dependency on the knowledge of μ for the choice of c can be removed through averaging of the iterates, at the cost of $(1-\alpha)/2$ in the rate. However, choosing α close to 1 causes a sampling error blowup, and one still cannot specify the constants in the rates without knowledge of μ.

As suggested by earlier works on stochastic approximation, it is preferred to average after a few iterations since the initial error is not forgotten faster than the sampling error with averaging.

6 Fast LSPI using Stochastic Approximation *(fLSPI-SA)*

LSPI [20] is a well-known algorithm for control based on the policy iteration procedure for MDPs. We propose here a fast variant of LSPI, which we shall henceforth refer to as fLSPI-SA. The latter algorithm works by substituting the regular LSTDQ with its stochastic approximation variant fLSTDQ-SA. We first briefly describe the LSPI algorithm and later provide a detailed description of fLSPI-SA.

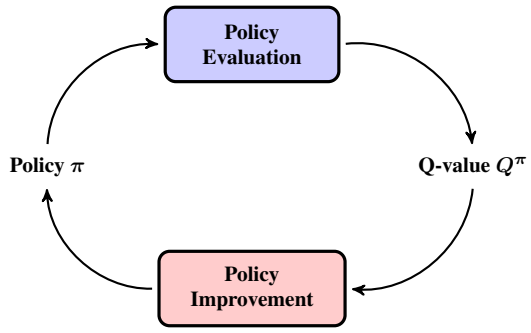


Fig. 2: Policy iteration principle central to LSPI.

6.1 Background for LSPI

We are given a set of samples $\mathcal{D} := \{(s_i, a_i, r_i, s_i'), i = 1, \ldots, T)\}$, where each sample i denotes a one-step transition of the MDP from state s_i to s_i' under action a_i, while resulting in a reward r_i. The objective is to find an *approximately optimal* policy using this set. This is in contrast with the goal of LSTD, which aims to approximate the state-value function of a particular policy (see Section 3.1).

For a given stationary policy π, the Q-value function $Q^\pi(s, a)$ for any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}(\mathcal{S})$ is defined as follows:

$$Q^\pi(s, a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t r(s_t, \pi(s_t)) \mid s_0 = s, a_0 = a\right]. \tag{18}$$

In the above, the initial state s and the action a in s are fixed, and thereafter the actions taken are governed by the policy π. This function can be thought of as the value function for a policy π in state s, given that the first action taken is the action a. As before, we parameterize the Q-value function using a linear function approximation architecture,

$$Q^\pi(s, a) \approx \theta^\intercal \phi(s, a), \tag{19}$$

where $\phi(s, a)$ is a d-dimensional feature vector corresponding to the tuple (s, a) and θ is a tunable policy parameter.

LSPI is built in the spirit of policy iteration algorithms. These perform policy evaluation and policy improvement in tandem, as illustrated in Fig. 2. For the purpose of policy evaluation, LSPI uses a LSTD-like algorithm called LSTDQ, which learns an approximation to the Q- (state-action value) function. It does this for any policy π, by solving the linear system

$$\hat{\theta}_T = \bar{A}_T^{-1} \bar{b}_T, \text{ where} \tag{20}$$

$$\bar{A}_T = \frac{1}{T} \sum_{i=1}^{T} \phi(s_i, a_i)(\phi(s_i, a_i) - \beta \phi(s_i', \pi(s_i')))^\intercal, \text{ and } \bar{b}_T = T^{-1} \sum_{i=1}^{T} r_i \phi(s_i, a_i).$$

As in the case of LSTD, the above can be seen as approximately solving a system of equations similar to (6), but in this case for the Q-value function. The pathwise LSTDQ variant is obtained by forming the dataset $\mathcal{D}$ from a sample path of the underlying MDP for a given policy π and also zeroing out the feature vector of the next state-action tuple in the last sample of the dataset.

The policy improvement step uses the approximate Q-value function to derive a greedily updated policy as follows:

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} \theta^\intercal \phi(s,a).$$

Since this policy is provably better than π, iterating this procedure (as illustrated in fig. 2) allows LSPI to find an approximately optimal policy.

6.2 fLSPI-SA Algorithm

The fast LSPI variant, henceforth referred to as fLSPI-SA, works by substituting the regular LSTDQ with its stochastic approximation variant fLSTDQ-SA. The overall structure of fLSPI-SA is given in Algorithm 2.

For a given policy π, fLSTDQ-SA approximates LSTDQ solution (20) by an iterative update scheme as follows (starting with an arbitrary θ_0):

$$\theta_k = \theta_{k-1} + \gamma_k \left(r_{i_k} + \beta \theta_{k-1}^\intercal \phi(s'_{i_k}, \pi(s'_{i_k})) - \theta_{k-1}^\intercal \phi(s_{i_k}, a_{i_k}) \right) \phi(s_{i_k}, a_{i_k}) \qquad (21)$$

From Section 3, it is evident that the claims in Proposition 7.1 and Theorem 4.2 hold for the above scheme as well.

Algorithm 2 fLSPI-SA

Input: Sample set $D := \{s_i, a_i, r_i, s'_i\}_{i=1}^T$, obtained from an initial (arbitrary) policy
Initialisation: ϵ, τ, step-sizes $\{\gamma_k\}_{k=1}^\tau$, initial policy π_0 (given as θ_0)
$\pi \leftarrow \pi_0$, $\theta \leftarrow \theta_0$
repeat
  Policy Evaluation
    Approximate LSTDQ(D, π) using fLSTDQ-SA(D, π) as follows:
    for $k = 1 \ldots \tau$ **do**
      Get random sample index: $i_k \sim U(\{1, \ldots, T\})$
      Update fLSTDQ-SA iterate θ_k using (21)
    end for
  $\theta' \leftarrow \theta_\tau$, $\Delta = \|\theta - \theta'\|_2$
  Policy Improvement
    Obtain a greedy policy π' as follows: $\pi'(s) = \arg\max_{a \in \mathcal{A}} \theta'^\intercal \phi(s,a)$
  $\theta \leftarrow \theta'$, $\pi \leftarrow \pi'$
until $\Delta < \epsilon$

6.3 Error bounds for fLSPI-SA

Here we present *prediction error* bounds that establish that using a SA based procedure in place of LSTD does not impact the overall convergence behaviour of the LSPI[2]. The prediction error is the difference in σ-weighted norm between the optimal value function V^* and the value function estimate obtained after running K iterations of fLSPI-SA. Here, σ denotes the so-called target distribution, forms part of an assumption made in [21] stating roughly that the mixing in the underlying Markov chain is sufficiently fast, and already

[2] As noted in [21], one can derive bounds for LSTDQ and the optimal Q-value function as well. However, for simplicity, here we use the value function and derive bounds on the prediction error of LSPI.

briefly mentioned in Remarks 5 and 7. In particular, given the current state, the future state of the underlying Markov chain is not allowed to deviate too far from σ at any time[3].

The following bound is for an on-policy version of fLSPI-SA: each iteration k involves generating a path of size T of the underlying MDP using the policy π_k. Therefore, the difference with the algorithm presented in Algorithm 2 is that the sample set changes in each iteration[4].

Theorem 6.1 (*Error Bound for iterates of fLSPI-SA*)
Let V^* denote the optimal value function, i.e., $V^*(s) := \max_\pi V^\pi(s)$ for any $s \in \mathcal{S}$. Let $\tilde{V}^{\pi_K}$ be the value estimate corresponding to the policy π_K that is obtained after running K iterations of fLSPI-SA. Then, under assumptions $1-4$ of [21] and with $\tau = T$ steps for fLSTD-SA in Algorithm 2, with probability $1-\delta$, we have

$$\left\|V^* - \tilde{V}^{\pi_K}\right\|_\sigma \leq \frac{4\beta}{(1-\beta)^2}\left[(1+\beta)\sqrt{CC_{\sigma,\nu}}\left(\frac{4\sqrt{2}E_0(\mathcal{F}) + E_2}{\sqrt{1-\beta^2}}\right.\right. \tag{22}$$
$$\left.\left. + \frac{2}{1-\beta}\left(\beta\sqrt{\frac{d}{\mu}}\sqrt{\frac{8\ln 8dK/\delta}{T}} + \frac{1}{T} + \frac{K_2(T)}{\sqrt{T+c}}\right) + E_1\right) + \beta^{\frac{K-1}{2}}\right],$$

where C, $C_{\sigma,\nu}$, μ, $\mathcal{F}$, $E_0(\mathcal{F})$, E_1 and E_2 are as in [21][5]. In particular,

- *ν is a distribution that lower-bounds the stationary distribution ρ^π of the Markov chain induced under the policy π such that $\mu \leq C\rho^\pi$ for some $C < \infty$ (see Assumption 1 in [21]).*
- *$\mathcal{F} := \{f_\theta \mid \theta \in \mathbb{R}^d$ and $f_\theta(\cdot) = \phi(\cdot)^\intercal\theta\}$ denotes the linear function space in which the value-functions are approximated, and $\tilde{\mathcal{F}} := \{g(\cdot) = \min\{f_\theta(\cdot), V_{\max}\} : f_\theta \in \mathcal{F}\}$ is the truncated version of this space.*
- *$E_0(\mathcal{F})$ is the approximation error for the worst value function in the space of functions considered and is defined by*

$$E_0(\mathcal{F}) := \sup_{\pi \in \mathcal{G}(\tilde{\mathcal{F}})} \inf_{f \in \mathcal{F}} \|f - V^\pi\|_{\rho^\pi},$$

  where $\mathcal{G}(\tilde{\mathcal{F}}) = \{\pi_f : \forall s, \pi_f(s) = \arg\max_{a \in \mathcal{A}(s)} \left(r(s,a) + \beta\mathbb{E}_{s,a}V(s')\right), f \in \tilde{\mathcal{F}}\}$, and the expectation $\mathbb{E}_{s,a}[V(s')]$ is taken w.r.t. state transition dynamics.
- *E_1 and E_2 are error terms, both of the order $O\left(\frac{1}{\sqrt{T}}\right)$ (see Theorems 4 and 5 of [21]).*
- *μ is the smallest eigenvalue of the covariance matrix $\frac{1}{T}\Phi^\intercal\Phi$ and is assured to be positive with high probability if the number of samples T in each iteration of fLSPI-SA is large enough (see Lemma 4 in [21]).*
- *$K_2(\cdot)$ and c are as defined in Theorem 4.2.*

Proof In lieu of Theorem 4.4, the proof of (22) follows in a similar manner as Theorem 8 of [21]. ■

[3] See Remark 2 in Section 6.1 of [21] for a detailed discussion on the target distribution σ.

[4] While off-policy LSPI is shown to work well in practice, no finite time analysis of this algorithm is available to the best of our knowledge. Moreover, Theorem 4.4 ensures that the fLSTD-SA iterate is a good approximation to LSTD, irrespective of the manner in which samples are collected.

[5] For consistency within our notation, we have exchanged roles of ν and μ from [21]

Remark 8 We highlight that all the terms on the RHS of (22) are the same as that obtained for the regular LSPI algorithm, except the term $K_2(T)/\sqrt{T+c}$. The latter term is present in the bound owing to the fact that we use an SA based scheme for policy evaluation instead of regular LSTD. It is evident that the resulting bound in (22) matches the order of the bound presented for LSPI in Theorem 8 of [21].

7 Convergence proofs

Throughout this section, we let $f_n(\theta) := \left(r_{i_n} + \beta\theta^\intercal\phi(s'_{i_n}) - \theta^\intercal\phi(s_{i_n})\right)\phi(s_{i_n})$ and $\mathcal{F}_n$ denotes the sigma algebra generated by $i_1, \ldots, i_n$.

Recall that the current state feature $T \times d$-matrix by $\Phi := (\phi(s_1)^\intercal, \ldots, \phi(s_T))$, next state feature $T \times d$-matrix by $\Phi' := (\phi(s'_1)^\intercal, \ldots, \phi(s'_T))$, and reward $T \times 1$-vector by $\mathcal{R} = (r_1, \ldots, r_T)^\intercal$. Recall also that the LSTD approximation to θ^* can as

$$\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T, \text{ where } \bar{A}_T = \frac{1}{T}(\Phi^\intercal\Phi - \beta\Phi^\intercal\Phi') \text{ and } \bar{b}_T = \frac{1}{T}\Phi^\intercal\mathcal{R}.$$

Finally we note also the pathwise LSTD approximation to θ^* has the same form, except that $\Phi' := \hat{P}\Phi = (\phi(s'_1)^\intercal, \ldots, \phi(s'_{T-1})^\intercal, \mathbf{0}^\intercal)$, where $\mathbf{0}$ is the $d \times 1$ zero-vector.

7.1 Proofs of almost sure convergence

Proof of Theorem 4.1 for fLSTD-SA without projection:

We first rewrite (10) as follows:

$$\theta_n = \theta_{n-1} + \gamma_n\left(-\bar{A}_T\theta_{n-1} + \bar{b}_T + \Delta M_n\right), \tag{23}$$

where $\Delta M_n = f_n(\theta_{n-1}) - \mathbb{E}(f_n(\theta_{n-1}) \mid \mathcal{F}_n)$ is a martingale difference sequence.

The ODE associated with (23) is

$$\dot{\theta}(t) = q(\theta(t)), \tag{24}$$

where $q(\theta(t)) := -\bar{A}_T\theta(t) + \bar{b}_T$.

To show that θ_n converges a.s. to $\hat{\theta}_T$, one requires that the iterate θ_n remains bounded a.s. Both boundedness and convergence can be inferred from Theorems 2.1-2.2(i) of [7], provided we verify assumptions (A1)-(A2) there. These assumptions are as follows:
(a1) The function q is Lipschitz. For any η, define $q_\eta(\theta) = q(\eta\theta)/\eta$. Then, there exists a continuous function q_∞ such that $q_\eta \to q_\infty$ as $\eta \to \infty$ uniformly on compact sets. Furthermore, the origin is an asymptotically stable equilibrium for the ODE

$$\dot{\theta}(t) = -q_\infty(\theta(t)). \tag{25}$$

(a2) The martingale difference $\{\Delta M_n, n \geq 1\}$ is square-integrable with

$$\mathbb{E}[\|\Delta M_{n+1}\|^2 \mid \mathcal{F}_n] \leq C_0(1 + \|\theta_n\|^2), n \geq 0,$$

for some $C_0 < \infty$.

We now verify (a1) and (a2) in our context. Notice that $q_\eta(\theta) := -\bar{A}_T\theta + \bar{b}_T/\eta$ converges to $q_\infty(\theta(t)) = -\bar{A}_T\theta(t)$ as $\eta \to \infty$. Since the matrix $\bar{A}_T$ is positive definite by

(A4), the aforementioned ODE has the origin as its globally asymptotically stable equilibrium. This verifies (a1).

For verifying (a2), notice that

$$\begin{aligned}\mathbb{E}[\|\Delta M_{n+1}\|^2 \mid \mathcal{F}_n] \leq & \mathbb{E}[\|f_{n+1}(\theta)\|^2 \mid \mathcal{F}_n] \\ \leq & (R_{\max} + (1+\beta)\Phi_{\max} \|\theta_n\|_2)^2\end{aligned}$$

The first inequality follows from the fact that for any random variable Y, $\mathbb{E}\|Y - E[Y \mid \mathcal{F}_n]\|^2 \leq \mathbb{E}Y^2$, while the second inequality follows from (A2) and (A3). The claim follows. ■

Proof of Theorem 4.1 for fLSTD-SA with projection:

We first rewrite (10) as follows:

$$\theta_n = \Upsilon\left(\theta_{n-1} + \gamma_n\left(-\bar{A}_T\theta_{n-1} + \bar{b}_T + \Delta M_n\right)\right), \tag{26}$$

where ΔM_n, $\mathcal{F}_n$ and $f_n(\theta)$ are as defined in (23).

From (A3) and the fact that the iterate θ_n is projected onto a compact and convex set $\mathcal{C}$, it is easy to see that the norm of the martingale difference ΔM_n is upper bounded by $R_{\max} + (1+\beta)H\Phi_{\max}$. Thus, (26) can be seen as a discretization of the ODE

$$\dot{\theta} = \breve{\Upsilon}(-\bar{A}_T\theta + \bar{b}_T), \tag{27}$$

where $\breve{\Upsilon}(\theta) = \lim_{\tau\to 0}\left[\left(\Upsilon\left(\theta + \tau f(\theta)\right) - \theta\right)/\tau\right]$, for any bounded continuous f. The operator $\breve{\Upsilon}$ ensures that θ governed by (27) evolves within the set $\mathcal{C}$. Since the matrix $\bar{A}_T$ is positive definite by (A4), it is evident that the ODE (25) in this case also has the origin as its globally asymptotically stable equilibrium. The claim now follows from Theorem 2 in Chapter 2 of [6] (or even Theorem 5.3.1 on pp. 191-196 of [18]). ■

7.2 Proofs finite-time error bounds for fLSTD-SA

To obtain high probability bounds on the computational error $\|\theta_n - \hat{\theta}_T\|_2$, we consider separately the deviation of this error from its mean (see (28) below), and the size of its mean itself (see (29) below). In this way the first quantity can be directly decomposed as a sum of martingale differences, and then a standard martingale concentration argument applied, while the second quantity can be analyzed by unrolling iteration (10)[6].

Proposition 7.1 below gives these results for general step sequences. The proof involes two martingale analyses which also form the templates for the proofs for the least squares regression extension (see Section 9), and the regularized and iterate averaged variants of fLSTD-SA (see Theorem 5.1).

After proving the results for general step sequences, we give the proof of Theorem 4.2, which gives explicit rates of convergence of the computational error in high probability for a specific choice of step sizes.

Proposition 7.1 *Let $z_n = \theta_n - \hat{\theta}_T$, where θ_n is given by* (10)*. Under (A1)-(A4), we have $\forall \epsilon > 0$,*

[6] In this proof we employ a technique similar to that used in [11]. However, our analysis is more elementary, and we make all the constants explicit for the problem at hand. Moreover, in order to eliminate a possible exponential dependence of the constants in the resulting bound on the reciprocal of $(1-\beta)\mu$, we depart from the argument in [11].

*(1) a bound in **high probability** for the **centered error**:*

$$\mathbb{P}\left(\|z_n\|_2 - \mathbb{E}\,\|z_n\|_2 \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{4(R_{\max} + (1+\beta)H\Phi_{\max})\sum_{k=1}^{n} L_k^2}\right), \quad (28)$$

where $L_k := \gamma_k \prod_{j=k+1}^{n}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\Phi_{\max}^4))^{1/2}$,

*(2) and a bound in **expectation** for the **non-centered error**:*

$$\mathbb{E}\left(\|z_n\|_2\right)^2 \leq \underbrace{\left[\prod_{k=1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^2\right)\|z_0\|_2\right]^2}_{\textbf{\textit{initial error}}} \quad (29)$$

$$\underbrace{+\,4\sum_{j=1}^{n}\gamma_k^2\left[\prod_{k=j}^{n-1}(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^2)\right]^2 (R_{\max} + (1+\beta)H\Phi_{\max})^2}_{\textbf{\textit{sampling error}}}.$$

The initial error depends on the initial point θ_0 of the algorithm. The sampling error arises out of a martingale difference sequence that depends on the random deviation of the stochastic update from the standard fixed point iteration. Note that the initial error is forgotten faster than the sampling error in this case.

Proof of Proposition 7.1 part (1):

The proof gives a martingale analysis of the centered computational error. It proceeds in three steps:

Step 1: (Decomposition of error into a sum of martingale differences)

Recall that $z_n := \theta_n - \hat{\theta}_T$. We rewrite $\|z_n\|_2 - \mathbb{E}\,\|z_n\|_2$ as a telescoping sum of martingale differences:

$$\|z_n\|_2 - \mathbb{E}\,\|z_n\|_2 = \sum_{k=1}^{n} g_k - \mathbb{E}[g_k\,|\mathcal{F}_{k-1}] = \sum_{k=1}^{n} D_k, \quad (30)$$

where $g_k := \mathbb{E}[\|z_n\|_2\,|\mathcal{F}_k]$, $D_k := g_k - \mathbb{E}[g_k\,|\mathcal{F}_{k-1}]$, and $\mathcal{F}_k$ denotes the sigma algebra generated by the random variables $\{i_1, \ldots, i_k\}$. D_k is the change in the exepcted error at time n after between iterations $k-1$ and k.

Step 2: (Showing the martingale differences are Lipschitz functions of the random innovations)

The next step is to show that the functions g_k are Lipschitz continuous in the random innovation at time k, with Lipschitz constants L_k. It then follows immediately that the martingale difference D_k is a Lipschitz function of the k^{th} random innovation with the same Lipschitz constant, which is the property leveraged in Step 3 below. In order to obtain Lipschitz constants with no exponential dependence on the inverse of $(1-\beta)\mu$ we depart from the general scheme of [11], and use our knowledge of the form of the random innovation f_k to eliminate the noise due to the rewards between time k and time n:

Recall that $f_j(\theta) := (\theta^\intercal \phi(s_{i_j}) - (r_{i_j} + \beta\theta^\intercal\phi(s'_{i_j})))\phi(s_{i_j})$ denotes the random innovation at time j given that $\theta_{j-1} = \theta$. Let $\Theta_j^k(\theta)$ denote the value of the random iterate at instant j evolving according to (10) and beginning from the value θ at time k.

First we note that as the projection, Υ, is a contraction mapping,

$$\begin{aligned}
&\mathbb{E}\left(\left\|\Theta_j^k(\theta) - \Theta_j^k(\theta')\right\|_2 \mid \mathcal{F}_{j-1}\right) \\
&\quad \leq \mathbb{E}\left(\left\|\Theta_{j-1}^k(\theta) - \Theta_{j-1}^k(\theta') - \gamma_j[f_j(\Theta_{j-1}^k(\theta)) - f_j(\Theta_{j-1}^k(\theta'))]\right\|_2 \mid \mathcal{F}_{j-1}\right).
\end{aligned}$$

Expanding the random innovation terms, we have

$$\begin{aligned}
&\Theta_{j-1}^k(\theta) - \Theta_{j-1}^k(\theta') - \gamma_j[f_j(\Theta_{j-1}^k(\theta)) - f_j(\Theta_{j-1}^k(\theta'))] \\
&\qquad = \Theta_{j-1}^k(\theta) - \Theta_{j-1}^k(\theta') \\
&\qquad\quad - \gamma_j[\phi(s_{i_j})\phi(s_{i_j})^\intercal - \beta\phi(s_{i_j})\phi(s'_{i_j})^\intercal](\Theta_{j-1}^k(\theta) - \Theta_{j-1}^k(\theta')) \\
&\qquad = [I - \gamma_j a_j](\Theta_{j-1}^k(\theta) - \Theta_{j-1}^k(\theta')),
\end{aligned} \tag{31}$$

where $a_j := [\phi(s_{i_j})\phi(s_{i_j})^\intercal - \beta\phi(s_{i_j})\phi(s'_{i_j})^\intercal]$. Note that

$$\begin{aligned}
a_j^\intercal a_j =& \phi(s_{i_j})\phi(s_{i_j})^\intercal\phi(s_{i_j})\phi(s_{i_j})^\intercal \\
&- \beta\left(\phi(s_{i_j})\phi(s_{i_j})^\intercal\phi(s_{i_j})\phi(s'_{i_j})^\intercal + \phi(s'_{i_j})\phi(s_{i_j})^\intercal\phi(s_{i_j})\phi(s_{i_j})^\intercal\right) \\
&+ \beta^2\phi(s'_{i_j})\phi(s_{i_j})^\intercal\phi(s_{i_j})\phi(s'_{i_j})^\intercal \\
=& \left\|\phi(s_{i_j})\right\|_2^2\left[\phi(s_{i_j})\phi(s_{i_j})^\intercal\right. \\
&\qquad \left. -\beta(\phi(s_{i_j})\phi(s'_{i_j})^\intercal + \phi(s'_{i_j})\phi(s_{i_j})^\intercal) + \beta^2\phi(s'_{i_j})\phi(s'_{i_j})^\intercal\right].
\end{aligned}$$

Recall that $\Phi^\intercal := (\phi(1), \ldots, \phi(T))$ and $\Phi'^\intercal := (\phi(1)', \ldots, \phi(T)')$, and assumption (A4) which implies that

$$\begin{aligned}
&\lambda_{min}\left(2\Phi^\intercal\Phi - \beta\left(\Phi'^\intercal\Phi + \Phi^\intercal\Phi'\right)\right) \\
&\qquad = \lambda_{min}\left((\Phi^\intercal\Phi - \beta\Phi'^\intercal\Phi) + (\Phi^\intercal\Phi - \beta\Phi'^\intercal\Phi)^\intercal\right) > 2T\mu.
\end{aligned}$$

So, setting $\Delta := \mathrm{diag}(\|\phi(\mathrm{s}_1)\|_2^2, \ldots, \|\phi(s_{j-1})\|_2^2)$, we find that for any vector θ:

$$\begin{aligned}
&\mathbb{E}\left(\theta^\intercal\left(I - \gamma_j a_{i_j}\right)^\intercal\left(I - \gamma_j a_{i_j}\right)\theta \mid \mathcal{F}_{j-1}\right) \\
&\quad = \theta^\intercal\mathbb{E}(I - \gamma_j[a_j^\intercal + a_j - \gamma_j a_j^\intercal a_j])\theta \mid \mathcal{F}_{j-1}) \\
&\quad = \|\theta\|_2^2 - \gamma_j\theta^\intercal\frac{1}{T}\sum_{k=1}^{T}\left[a_k^\intercal + a_k - \gamma_j a_k^\intercal a_k\right]\theta \\
&\quad = \|\theta\|_2^2 - \gamma_j\theta^\intercal\frac{1}{T}\left[2\Phi^\intercal\Phi - \beta\left(\Phi^\intercal\Phi' + \Phi'^\intercal\Phi\right)\right. \\
&\qquad\qquad \left. -\gamma_j\left(\Phi^\intercal\Delta\Phi - \beta\left(\Phi'^\intercal\Delta\Phi + \Phi^\intercal\Delta\Phi'\right) + \beta^2\Phi'^\intercal\Delta\Phi'\right)\right]\theta \\
&\quad \leq (1 - \gamma_j(2\mu - \gamma_j\Phi_{\max}^4(1+\beta)^2))\,\|\theta\|_2^2
\end{aligned}$$

For the third equality, we have used that $\sum_{k=1}^{T}\phi(s_k)^\intercal\phi(s_k) = \Phi^\intercal\Phi$ and similar identities. For the inequality, we have used the boundedness assumption on the features, (A2), together

with the assumption (A4). Hence, from the tower property of conditional expectations, it follows that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\Theta_n^k(\theta)-\Theta_n^k(\theta')\right\|_2^2\right] = \mathbb{E}\left[\mathbb{E}\left(\left\|\Theta_n^k(\theta)-\Theta_n^k(\theta')\right\|_2^2 \mid \mathcal{F}_{n-1}\right)\right] \\
&\leq \left(1-\gamma_n\left(2\mu-\gamma_n\Phi_{\max}^4(1+\beta)^2\right)\right)\mathbb{E}\left[\left\|\Theta_{n-1}^k(\theta)-\Theta_{n-1}^k(\theta')\right\|_2^2\right] \\
&\leq \left[\prod_{j=k+1}^{n}\left(1-\gamma_j\left(2\mu-\gamma_j\Phi_{\max}^4(1+\beta)^2\right)\right)\right]\left\|\theta-\theta'\right\|_2^2 \qquad (32)
\end{aligned}$$

Finally, writing f and f' for two possible values of the random innovation at time k, and writing $\theta = \theta_{k-1} + \gamma_i f$ and $\theta' = \theta_{k-1} + \gamma_k f'$, we have that

$$\begin{aligned}
&\left|\mathbb{E}\left[\left\|\theta_n-\hat{\theta}_T\right\|_2 |\theta_k=\theta\right] - \mathbb{E}\left[\left\|\theta_n-\hat{\theta}_T\right\|_2 |\theta_k=\theta'\right]\right| \\
&\leq \mathbb{E}\left[\left\|\Theta_n^k(\theta)-\Theta_n^k(\theta')\right\|_2\right] \leq L_k\gamma_k\left\|f-f'\right\|_2 = L_k\left\|f-f'\right\|_2.
\end{aligned}$$

where

$$L_k := \gamma_k\left[\prod_{j=k+1}^{n}\left(1-\gamma_j\left(2\mu-\gamma_j\Phi_{\max}^4(1+\beta)^2\right)\right)\right]^{1/2}$$

which proves that the functions g_k are L_k-Lipschitz in the random innovations at time k.

Step 3: (Applying a subgaussian concentration inequality)

Now we derive a standard martingale concentration bound in the lemma below. Note that, for any $\lambda > 0$,

$$\begin{aligned}
\mathbb{P}(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon) &= \mathbb{P}\left(\sum_{k=1}^{n} D_k \geq \epsilon\right) \leq \exp(-\lambda\epsilon)\mathbb{E}\left(\exp\left(\lambda\sum_{k=1}^{n} D_k\right)\right) \\
&= \exp(-\lambda\epsilon)\mathbb{E}\left(\exp\left(\lambda\sum_{k=1}^{n-1} D_k\right)\mathbb{E}\Big(\exp(\lambda D_n)\,|\mathcal{F}_{n-1}\Big)\right).
\end{aligned}$$

The first equality above follows from (30), while the inequality follows from Markov's inequality. Now for any bounded random variable f, and L-Lipschitz function D we have

$$\mathbb{E}\left(\exp(\lambda g(f))\right) \leq \exp\left(\lambda^2 B L^2/2\right),$$

where $|f| < B$. Note that by (A3), and the projection step of the algorithm, we have that $|f_k(\theta_{k-1})| < (R_{\max} + (1+\beta)H\Phi_{\max})$ is a bounded random variable, and, conditioned on $\mathcal{F}_{k-1}$, D_k is Lipschitz in $f_k(\theta_{k-1})$ with constant L_k. So we obtain

$$\mathbb{E}\left(\exp(\lambda D_n)\,|\mathcal{F}_{n-1}\right) \leq \exp\left(\frac{\lambda^2(R_{\max}+(1+\beta)H\Phi_{\max})L_n^2}{2}\right),$$

and so

$$\mathbb{P}(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon) \leq \exp(-\lambda\epsilon)\exp\left(\frac{\lambda^2(R_{\max}+(1+\beta)H\Phi_{\max})}{2}\sum_{k=1}^{n} L_k^2\right). \qquad (33)$$

The proof of Proposition 7.1 part (1) follows by optimizing over λ in (33). ■

Proof of Proposition 7.1 part (2)

The proof of this result also follows a martingale analysis. In contrast to the high probability bound, here we work directly with the error, rather than the centered error, and split it into predictable and martingale parts. Bounding the predictable part then bounds the influence of the initial error, and bounding the martingale part bounds the error due to sampling.

Step 1: (Extract a martingale difference from the update)

First, by using that $\bar{A}_T = \mathbb{E}((\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^\intercal \mid \mathcal{F}_{n-1})$ and that $\mathbb{E}(f_n(\hat{\theta}_T) \mid \mathcal{F}_{n-1}) = 0$, we can rearrange the update rule (10) to get

$$\begin{aligned}\theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1}) &= \theta_{n-1} - \hat{\theta}_T - \gamma_n(\mathbb{E}(f_n(\theta_{n-1}) - \Delta M_n) \\ &= \left(I - \gamma_n \bar{A}_T\right) z_{n-1} - \gamma_n \Delta M_n\end{aligned}$$

where $\Delta M_n := f_n(\theta_{n-1}) - \mathbb{E}(f_n(\theta_{n-1}) \mid \mathcal{F}_{n-1})$ is a martingale difference.

Step 2: (Apply Jensen to the square of the norm)

From Jensen's inequality, and the fact that the projection in the update rule (10) is a contraction, we obtain

$$\begin{aligned}&\mathbb{E}\left(\|z_n\|_2 \mid \mathcal{F}_{n-1}\right)^2 \leq \mathbb{E}(\langle z_n, z_n\rangle \mid \mathcal{F}_{n-1}) \\ &\leq \mathbb{E}(\langle \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1}), \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1})\rangle \mid \mathcal{F}_{n-1}) \\ &= \mathbb{E}(\langle \left(I - \gamma_n \bar{A}_T\right) z_{n-1} - \gamma_n \Delta M_n, \left(I - \gamma_n \bar{A}_T\right) z_{n-1} - \gamma_n \Delta M_n\rangle \mid \mathcal{F}_{n-1}) \\ &= z_{n-1}^\intercal \left(I - \gamma_n \bar{A}_T\right)^\intercal \left(I - \gamma_n \bar{A}_T\right) z_{n-1} + \gamma_n^2 \mathbb{E}\left(\langle \Delta M_n, \Delta M_n\rangle \mid \mathcal{F}_{n-1}\right) \qquad (34) \\ &\leq \|z_{n-1}\|_2^2 \left\|\left(I - \gamma_n \bar{A}_T\right)^\intercal \left(I - \gamma_n \bar{A}_T\right)\right\|_2 + \gamma_n^2 \mathbb{E}\left(\|\Delta M_n\|_2^2 \mid \mathcal{F}_{n-1}\right)\end{aligned}$$

Note that the cross-terms have vanished in (34) since ΔM_n is martingale difference, independent of the other terms, given $\mathcal{F}_{n-1}$.

Step 3: (Unroll the iteration)

Using assumptions (A2) and (A4)

$$\begin{aligned}\left\|(I - \gamma_n \bar{A}_T)^\intercal (I - \gamma_n \bar{A}_T)\right\|_2 &= \left\|(I - \gamma_n((\bar{A}_T^\intercal + \bar{A}_T) - \gamma_n \bar{A}_T^\intercal \bar{A}_T)\right\|_2 \qquad (35) \\ &\leq 1 - \gamma_n(2\mu - \gamma_n(1+\beta)^2 \Phi_{\max}^2) \qquad (36)\end{aligned}$$

Furthermore, by assumption (A3), and the projection step, the martingale differences ΔM_n are bounded in norm by $2(R_{\max} + (1+\beta)H\Phi_{\max})$. By applying the tower property of conditional expectations repeatedly together with (36) we arrive at the bound:

$$\begin{aligned}\mathbb{E}\left(\|z_n\|_2\right)^2 &\leq \left[\prod_{k=1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^2\right)\|z_0\|_2\right]^2 \\ &+ 4\sum_{j=1}^{n}\gamma_k^2\left[\prod_{k=j}^{n-1}(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^2)\right]^2 (R_{\max} + (1+\beta)H\Phi_{\max})^2\end{aligned}$$

■

Derivation of rates or Proof of Theorem 4.2

Proof

High probability bound: Let $\gamma_n = \frac{c_0 c}{(c+n)}$, and choose $c_0 \in (0, \mu((1+\beta)^2\Phi_{\max}^4)^{-1}]$. Then,

$$\sum_{i=1}^{n} L_i^2 = \sum_{i=1}^{n} \frac{c_0^2 c^2}{(c+i)^2} \prod_{j=i}^{n} \left(1 - \frac{c_0 c}{(c+j)} \left(2\mu - (1+\beta)^2 \Phi_{\max}^4 \frac{c_0 c}{(c+j)}\right)\right)$$
$$\leq \sum_{i=1}^{n} \frac{c_0^2 c^2}{(c+i)^2} \exp\left(-c_0 c\mu \sum_{j=i}^{n} \frac{1}{(c+j)}\right)$$
$$\leq \frac{c_0^2 c^2}{(n+c)^{c_0 c\mu}} \sum_{i=1}^{n} (i+c)^{-(2-c_0 c\mu)}.$$

We now find three regimes for the rate of convergence, based on the choice of c:
(i) $\sum_{i=1}^{n} L_i^2 = O\left((n+c)^{c_0 c\mu}\right)$ when $c_0 c\mu \in (0,1)$,
(ii) $\sum_{i=1}^{n} L_i^2 = O\left(n^{-1}\ln n\right)$ when $c_0 c\mu = 1$, and
(iii) $\sum_{i=1}^{n} L_i^2 \leq \frac{c_0^2 c^2}{(c_0 c\mu - 1)}(n+c)^{-1}$ when $c_0 c\mu \in (1,\infty)$.
(We have used comparisons with integrals to bound the summations, such as $\sum_{j=1}^{n} n^{-1} \geq \int_1^n x^{-1} dx$.) Thus, setting $c \in (1/(c_0\mu), 2/(c_0\mu))$, the high probability bound from Proposition 7.1 gives

$$\mathbb{P}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2 - \mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2(n+c)}{4K_{\mu,c,c_0,\beta}}\right) \tag{37}$$

where $K_{\mu,c,c_0,\beta} := \dfrac{c_0^2 c^2 (R_{\max} + (1+\beta)H\Phi_{\max})}{(c_0 c\mu - 1)}$.

Expectation bound: Under the same choice for c_0, and supposing that $2c_0 c\mu \in (1,\infty)$, we have:

$$\sum_{k=1}^{n} \gamma_k^2 \left[\prod_{j=k+1}^{n-1} (1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\Phi_{\max}^2)\right]^2$$
$$\leq \sum_{k=1}^{n-1} \gamma_{k+1}^2 \exp\left(-2c_0 c\mu \left(\sum_{k+1}^{n} \frac{1}{c+j}\right)\right)$$
$$\leq \frac{c_0^2 c^2}{(n+c)^{c_0 c\mu}} \sum_{k=1}^{n} (c+k)^{-(2-2c_0 c\mu)} \leq \frac{c_0^2 c^2}{(2c_0 c\mu - 1)(n+c)}$$

where in the last inequality we have again compared the sum with an integral. Similarly, supposing that $c_0 c\mu \in (1,\infty)$, we have

$$\prod_{k=1}^{n} (1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^2)$$
$$\leq \exp\left(-c_0 c\mu \sum_{j=1}^{n} \frac{1}{c+j}\right) \leq \left(\frac{1}{n+c}\right)^{c_0 c\mu} \leq \left(\frac{1}{n+c}\right).$$

So we have, when $c_0 c\mu \in (1, \infty)$,

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \left(\frac{\sqrt{c}\,\|\theta_0 - \theta^*\|_2}{\sqrt{(n+c)^{c_0 c\mu - 1}}} + \frac{4c_0 c(R_{\max} + (1+\beta)H\Phi_{\max}}{2c_0 c\mu - 1}\right)(c+n)^{-\frac{1}{2}}, \tag{38}$$

and the result in Theorem 4.2 now follows. ■

7.3 Proof of expectation bound for fLSTD-SA without projection

The proof of the theorem follows just as the proof of Theorem 4.2 but using the following proposition in place of Proposition 7.1 part 2. The proof of the following proposition differs from that of Proposition 7.1 part 2 in that the decomposition of the computational error extracts a noise term dependent only on $\hat{\theta}_T$ rather than on θ_n, and so projection is not needed:

Proposition 7.2 *Let $z_n = \theta_n - \hat{\theta}_T$, where θ_n is given by* (11)*. Under (A1)-(A4), we have $\forall \epsilon > 0$,*

$$\mathbb{E}\left(\|z_n\|_2\right)^2 \leq \underbrace{\left[\prod_{k=1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^4\right)\|z_0\|_2\right]^2}_{\textit{initial error}} \tag{39}$$

$$+ \underbrace{\sum_{j=1}^{n}\gamma_k^2\left[\prod_{k=j}^{n-1}(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\Phi_{\max}^4)\right]^2\left(R_{\max} + (1+\beta)\left\|\hat{\theta}_T\right\|_2\right)^2}_{\textit{sampling error}}$$

Proof

Step 1: (Unrolling the error recursion)

First, by rearranging the update rule (10) we obtain an iteration for the computational error $z_n = \theta_n - \hat{\theta}_T$, and subsequently unroll this iteration:

$$\begin{aligned} z_n &= \theta_n - \hat{\theta}_T = \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1}) \\ &= \left(I - \gamma_n(\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^\intercal\right) z_{n-1} - \gamma_n f_n(\hat{\theta}_T) \\ &= \mathit{\Pi}_1^n z_0 + \sum_{k=1}^{n}\gamma_k \mathit{\Pi}_k^{n-1} f_k(\hat{\theta}_T) \end{aligned}$$

where $\mathit{\Pi}_k^n := \prod_{j=k}^{n}\left(I - \gamma_j(\phi(s_{i_j}) - \beta\phi(s'_{i_j}))\phi(s_{i_j})^\intercal\right)$, and we have used that the random increment at time n has the form $f_n(\theta) = (\theta^\intercal\phi(s_{i_n}) - (r_{i_n} + \beta\theta^\intercal\phi(s'_{i_n})))\phi(s_{i_n})$. Notice that by the definition of the LSTD solution, we have that $\mathbb{E}(f_n(\hat{\theta}_T) \mid \mathcal{F}_{n-1}) = 0$, and so $f_n(\hat{\theta}_T)$ is a zero mean random variable.

Step 2: (Taking the expectation of the norm)

From Jensen's inequality, we obtain

$$\mathbb{E}\left(\|z_n\|_2\right)^2 \leq \left(\mathbb{E}(\langle z_n, z_n\rangle)\right)^2$$
$$= z_0^{\intercal}\mathbb{E}\left(\Pi_1^{n\intercal}\Pi_1^n\right) z_0 + \sum_{k=1}^{n}\gamma_k^2\mathbb{E}\left(f_k(\hat{\theta}_T)^{\intercal}\Pi_k^{n-1\intercal}\Pi_k^{n-1}f_k(\hat{\theta}_T)\right) \quad (40)$$

Note that the cross-terms have vanished in (40) since $f_k(\hat{\theta}_T)$ is not only zero mean, but also independent of all the random variables i_j for which $j \neq k$.

Now using assumptions (A2) and (A4)

$$\left\|\mathbb{E}(\left(I-\gamma_n(\phi(s_{i_n})-\beta\phi(s'_{i_n}))\phi(s_{i_n})^{\intercal}\right)^{\intercal}\left(I-\gamma_n(\phi(s_{i_n})-\beta\phi(s'_{i_n}))\phi(s_{i_n})^{\intercal}\right)\right\|_2$$
$$= \left\|\mathbb{E}\left(I-\gamma_n((\phi(s_{i_n})-\beta\phi(s'_{i_n}))\phi(s_{i_n})^{\intercal}-\gamma_n\phi(s_{i_n})(\phi(s_{i_n})-\beta\phi(s'_{i_n}))^{\intercal}\right.\right.$$
$$\left.\left.+\gamma_n^2\left(\|\phi(s_{i_n})\|_2^2-2\beta\langle\phi(s'_{i_n}),\phi(s_{i_n})\rangle+\beta^2\left\|\phi(s'_{i_n})\right\|_2^2\right)\phi(s_{i_n}))\phi(s_{i_n}))^{\intercal}\right)\right\|_2$$
$$\leq 1-\gamma_n(2\mu-\gamma_n(1+\beta)^2\Phi_{\max}^2) \quad (41)$$

Furthermore, by assumption (A3), the random variables $f_n(\hat{\theta}_T)$ are bounded in norm by $R_{\max}+(1+\beta)\|\hat{\theta}_T\|$. So, by applying the tower property of conditional expectations repeatedly together with (41) we arrive at the bound:

$$\mathbb{E}\left(\|z_n\|_2\right) \leq \left(\left[\prod_{k=1}^{n}(1-\gamma_k(2\mu-\gamma_k(1-\beta)^2\Phi_{\max}^4)\right]\|z_0\|_2\right]^2$$
$$+\sum_{j=1}^{n}\gamma_k^2\left[\prod_{k=j}^{n-1}(1-\gamma_k(2\mu-\gamma_k(1+\beta)^2\Phi_{\max}^4)\right]^2\left(R_{\max}+(1+\beta)\left\|\hat{\theta}_T\right\|_2\right)^2\Bigg)^{\frac{1}{2}}$$

■

7.4 Proofs of finite time bounds for iterate averaged fLSTD-SA

Map of the proof of Theorem 5.1:

- We first give a bound on the error in high probability for the averaged iterates in Proposition 7.3 below. This result is for general step-size sequences.
- Next, we derive the bounds for the Lipschitz constants L_m when the iterates are averaged and the step-sizes are chosen to be $\gamma_n = c_0\,(c/c+n)^{-\alpha}$ for some $\alpha \in (1/2, 1)$. This is a crucial step that helps in establishing the order $O(n^{-\alpha/2})$ rate for the high-probability bound in Theorem 4.2, independent of the choice of c. Recall that in order to obtain this rate for the algorithm without averaging one had to choose $c_0\mu c \in (1,\infty)$.
- Finally, we bound the expected error by directly averaging the errors of the non-averaged iterates:

$$\mathbb{E}\left\|\bar{\theta}_{n+1}-\hat{\theta}_T\right\|_2 \leq \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left\|\theta_k-\hat{\theta}_T\right\|_2,$$

and directly applying the bounds in expectation given in Proposition 7.1. This involves specializing the bounds for the bound in expectation in Proposition 7.1 for the new choice of step-size sequence.

Proposition 7.3 *Under (A1)-(A3) we have, for all* $\epsilon \geq 0$ *and* $\forall n \geq 1$,

$$\mathbb{P}(\|z_n\|_2 - \mathbb{E}\,\|z_n\|_2 \geq \epsilon) \leq \exp\left(-\frac{\epsilon^2}{2(R_{\max}+(1+\beta)H)^2\sum_{m=1}^{n} L_m^2}\right),$$

where $L_m := \frac{\gamma_i}{n}\left(\sum_{l=m+1}^{n-1}\prod_{j=m}^{l}\left(1-\gamma_{j+1}\left(2\mu-\gamma_{j+1}(1+\beta)^2\Phi_{\max}^4\right)\right)^{1/2}\right)$.

Proof

Recall that z_n denotes the error of the algorithm at time n, which in this case is $z_n = \left\|\bar{\theta}_n - \theta\right\|_2$. The proof follows the scheme of the proof of Proposition 7.1, part (1), given in Section 7.2:

Step 1: As before, we decompose the centered error into a sum of martingale differences:

$$\|z_n\|_2 - \mathbb{E}\,\|z_n\|_2 = \sum_{k=1}^{n} D_k, \tag{42}$$

where $D_k := g_k - \mathbb{E}[g_k\,|\mathcal{F}_{k-1}\,]$ and $g_k := \mathbb{E}[\|z_n\|_2\,|\mathcal{F}_k\,]$.

Step 2: We need to prove that the functions g_k are Lipschitz continuous in the random innovation at time k with the new constants L_k. Recall from Step 2 of the proof of the high probability bound in Theorem 7.1 in Section 7.2 that the random variables $\Theta_n^k(\theta)$ is defined to be the value of the iterate at time n that evolves according to (10), and beginning from θ at time k. Now we define

$$\bar{\Theta}_n^k(\bar{\theta},\theta) = \frac{(k-1)\bar{\theta}}{n} + \frac{1}{n}\sum_{k=1}^{n}\Theta_n^k(\theta).$$

Then, letting f and f' deonte two possible values for the random innovation at time k, and setting $\theta = \theta_{k-1} + \gamma_k f$ and $\theta' = \theta_{k-1} + \gamma_k f'$, we have

$$\begin{aligned}\mathbb{E}\left\|\bar{\Theta}_n^k\left(\bar{\theta}_{k-1},\theta\right) - \bar{\Theta}_n^k\left(\bar{\theta}_{k-1},\theta'\right)\right\|_2 &\leq \mathbb{E}\left\|\frac{1}{n}\sum_{l=k}^{n}\left(\Theta_l^k\left(\theta\right) - \Theta_l^k\left(\theta'\right)\right)\right\|_2 \\ &\leq \frac{1}{n}\sum_{l=k}^{n}\prod_{j=k+1}^{l}\left(1-\gamma_j\left(2\mu-\gamma_j(1+\beta)^2\Phi_{\max}^4\right)\right)^{1/2}\left\|f-f'\right\|_2 \end{aligned} \tag{43}$$

where we have used (32) derived in Step 2 of the proof the high probability bound in Proposition 7.1. Hence, similarly to Step 2 of the proof of Proposition 7.1, part (1), we find that g_k is L_k-Lipschitz in the random inovation at time k, and so D_k is also.

Step 3 follows in a similar manner to the proof of Proposition 7.1, part (1). ■

Proof of the high probability bound in Theorem 5.1:

The following lemma derives the bounds for the Lipschitz constants L_i when the step-sizes are chosen to be $\gamma_n = c_0\left(c/(c+n)\right)^{-\alpha}$ for some $\alpha \in \left(\frac{1}{2},1\right)$, and the iterates are then averaged. The main ingredients of this derivation can be found in the argument of pp. 15 in [10]. However, here we manage to give all the constants explicitly:

Lemma 7.1 *Under conditions of Theorem 5.1, we have*

$$\sum_{i=1}^{n} L_i^2 \leq \frac{1}{\mu^2}\left\{2^\alpha + \left[\frac{2\alpha}{c^\alpha} + \frac{2\alpha}{1-\alpha}\right]\right\}^2 \frac{1}{n}. \tag{44}$$

Proof See Appendix A. ■

Proof of the bound in expectation in Theorem 5.1:

We bound the expected error by directly averaging the errors of the non-averaged iterates:

$$\mathbb{E}\left\|\bar{\theta}_{n+1} - \hat{\theta}_T\right\|_2 \leq \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left\|\theta_k - \hat{\theta}_T\right\|_2, \tag{45}$$

and directly applying the bounds in expectation given in Proposition 7.1. The following lemma specializes the bounds for the bound in expectation in Proposition 7.1 for the new choice of step-size sequence and then averages the resulting bound using (45) to obtain the final rate in expectation in Theorem 5.1.

Lemma 7.2 *Under conditions of Theorem 5.1, we have*

$$\begin{aligned}\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq & \left(\sum_{n=1}^{\infty}\exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right)\right) \\ & \cdot\left(\|\theta_0 - \theta_T\|_2 + e + \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}\exp\left(\frac{2\alpha}{1-\alpha}\right)\right)\frac{1}{n} \\ & + 2\left(R_{\max} + (1+\beta)H\Phi_{\max}\right)c^\alpha\left(c_0\mu c^\alpha\right)^{-\alpha\frac{1+2\alpha}{2(1-\alpha)}}(n+c)^{-\frac{\alpha}{2}}.\end{aligned}$$

Proof See Appendix B. ■

8 Traffic Control Application

8.1 Simulation Setup

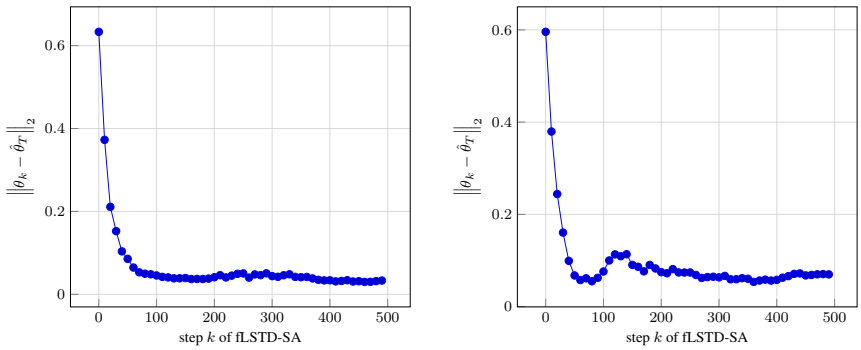
The idea behind the experimental setup is to study both LSPI and the variant of LSPI, fLSPI-SA, where we use fLSTDQ-SA as a subroutine to approximate the LSTDQ solution. Algorithm 2 provides the pseudo-code for the latter algorithm.

We consider a traffic signal control application for conducting the experiments. The problem here is to adaptively choose the sign configurations for the signalized intersections in the road network considered, in order to maximize the traffic flow in the long run. Let L be the total number of lanes in the road network considered. Further, let $q_i(t), i = 1, \ldots, L$ denote the queue lengths and $t_i(t), i = 1, \ldots, L$ the elapsed time (since signal turned to red) on the individual lanes of the road network. Following [29], the traffic signal control MDP is formulated as follows:

State $s_t = \big(q_1(t), \ldots, q_L(t), t_1(t), \ldots, t_L(t)\big)$,
Action a_t belongs to the set of feasible sign configurations,
Single-stage cost $h(s_t) = u_1\left[\sum_{i\in I_p} u_2 \cdot q_i(t) + \sum_{i\notin I_p} w_2 \cdot q_i(t)\right] + w_1\left[\sum_{i\in I_p} u_2 \cdot t_i(t) + \sum_{i\notin I_p} w_2 \cdot t_i(t)\right]$, where $u_i, w_i \geq 0$ such that $u_i + w_i = 1$ for $i = 1, 2$ and $u_2 > w_2$. Here, the set I_p is the set of prioritized lanes.

Table 1: Features for the traffic control application

State	**Action**	**Feature $\phi_i(s,a)$**
$q_i < \mathcal{L}_1$ and $t_i < \mathcal{T}_1$	RED GREEN	0.01 0.06
$q_i < \mathcal{L}_1$ and $t_i \geq \mathcal{T}_1$	RED GREEN	0.02 0.05
$\mathcal{L}_1 \leq q_i < \mathcal{L}_2$ and $t_i < \mathcal{T}_1$	RED GREEN	0.03 0.04
$\mathcal{L}_1 \leq q_i < \mathcal{L}_2$ and $t_i \geq \mathcal{T}_1$	RED GREEN	0.04 0.03
$q_i \geq \mathcal{L}_2$ and $t_i < \mathcal{T}_1$	RED GREEN	0.05 0.02
$q_i \geq \mathcal{L}_2$ and $t_i \geq \mathcal{T}_1$	RED GREEN	0.06 0.01



(a) Tracking error on 7x9-grid network

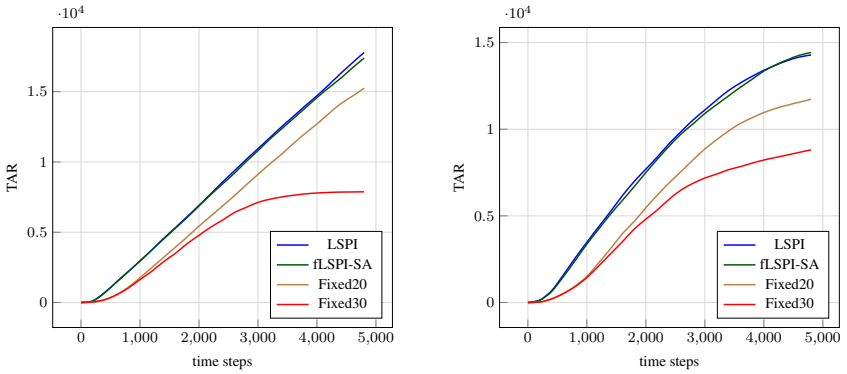
(b) Tracking error on 14x9-grid network

Fig. 3: Tracking error of fLSTDQ-SA in iteration 1 of fLSPI-SA on two grid networks.

Function approximation is a standard technique employed to handle high-dimensional state spaces (as is the case with the traffic signal control MDP on large road networks). We employ the feature selection scheme from [30], which is briefly described in the following: the features $\phi(s,a)$ corresponding to any state-action tuple (s,a) is an L-dimensional vector, with one bit for each line in the road network. The feature value $\phi_i(s,a), i = 1,\ldots,L$ corresponding to lane i is chosen as described in Table 1, with q_i and t_i denoting the queue length and elapsed times for lane i. Thus, as the size of the network increases, the feature dimension scales in a linear fashion.

Note that the feature selection scheme depends on certain thresholds $\mathcal{L}_1$ and $\mathcal{L}_2$ on the queue length and $\mathcal{T}_1$ on the elapsed times. The motivation for using such graded thresholds is owing to the fact that queue lengths are difficult to measure precisely in practice. We set $(\mathcal{L}_1, \mathcal{L}_2, \mathcal{T}_1) = (6, 14, 130)$ in all our experiments and this choice has been used, for instance, in [30].

We implement both LSPI as well as fLSPI-SA for the above problem. The experiments involve two stages - an initial training stage where LSPI/fLSPI-SA is run to find an approx-



(a) Throughput (TAR) on 7x9-grid network

(b) Throughput (TAR) on 14x9-grid network

Fig. 4: Performance comparison of LSPI and fLSPI-SA using throughput (TAR) on two grid networks.

imately optimal policy and a test stage where ten independent simulations are run using the policy that LSPI/fLSPI-SA converged to in the training stage. In the training stage, for both LSPI and fLSPI-SA, we collect $T = 10000$ samples from an exploratory policy that picks the actions in a uniformly random manner. For both LSPI and fLSPI-SA, we set $\beta = 0.9$ and $\epsilon = 0.1$. We set τ, the number of fLSTDQ-SA iterations in fLSPI-SA, to 500. This choice is motivated by an experiment where we observed that at 500 steps, fLSTD-SA is already very close to LSTDQ and taking more steps did not result in any significant improvements for fLSPI-SA. We implement the regularized variant of LSTDQ, with regularization constant μ set to 1. The step-size γ_k used in the update iteration of fLSTDQ-SA is set as recommended by Theorem 4.2.

8.2 Results

We use total arrived road users (TAR) and runtimes as performance metrics for comparing the algorithms implemented. TAR is a throughput metric that denotes the total number of road users who have reached their destination, while runtimes are measured for the policy evaluation step in LSPI/fLSPI-SA. For fLSTDQ-SA, which is the policy evaluation algorithm in fLSPI-SA, we also report the tracking error, which measures the distance in ℓ^2 norm between the fLSTD-SA iterate θ_k, $k = 1, \ldots, \tau$ and LSTDQ solution $\hat{\theta}_T$.

We report the tracking error and total arrived road users (TAR) in Fig. 3 and Fig. 4, respectively. The run-times obtained from our experimental runs for LSPI and fLSPI-SA is presented in Fig. 5. Iteration 1 for fLSPI-SA is used for reporting the tracking error and we observed similar behavior across iterations, i.e., we observed that fLSTD-SA iterate θ_τ is close to the corresponding LSTDQ solution in each iteration of fLSPI-SA. The experiments are performed for four different grid networks of increasing size and hence, increasing feature dimension.

From Fig. 3a–3b, we observe that fLSTD-SA algorithm converges rapidly to the corresponding LSTDQ solution. Further, from the runtime plots (see Fig. 5), we notice that fLSPI-SA is several orders of magnitude faster than regular LSPI. From a traffic application

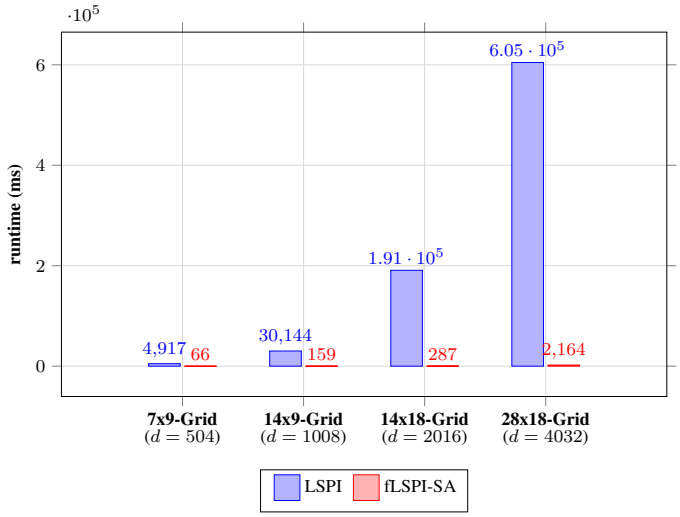


Fig. 5: Run-times of LSPI and fLSPI-SA on four road networks

standpoint, we observe in Figs. 4a–4b that fLSPI-SA results in a throughput (TAR) performance that is on par with LSPI. Moreover, the throughput observed for LSPI/fLSPI-SA is higher than that for a traffic light control (TLC) algorithm that cycles through the sign configurations in a round-robin fashion, with a fixed green time period for each sign configuration. We report the TAR results in Figs. 4a–4b for two such fixed timing TLCs with periods 10 and 20, respectively denoted Fixed10 and Fixed20. The rationale behind this comparison is that fixed timing TLCs are the de facto standard. Moreover, the results establish that LSPI outperforms fixed timing TLCs that we implemented and fLSPI-SA gives performance comparable to that of LSPI, but at a lower computational cost.

9 Extension to Least Squares Regression

In this section, we describe the classic parameter estimation problem using the method of least squares, the standard approach to solve this problem and the low-complexity SGD alternative. Subsequently, we outline the fast LinUCB algorithm that uses a SGD iterate in place of least squares solutions and present the numerical experiments for this algorithm on a news recommendation application.

9.1 Least squares regression and SGD

In this setting, we are given a set of samples $\mathcal{D} := \{(x_i, y_i), i = 1, \ldots, T\}$ with the underlying observation model $y_i = x_i^\intercal \theta^* + \xi_i$ (ξ_i is a bounded, zero-mean random variable, and θ^* is an unknown parameter). The least squares estimate $\hat{\theta}_T$ minimizes $\sum_{i=1}^T (y_i - \theta^\intercal x_i)^2$. It can be shown that $\hat{\theta}_T = \bar{A}_T^{-1} b_T$, where $\bar{A}_T = T^{-1} \sum_{i=1}^T x_i x_i^\intercal$ and $\bar{b}_T = T^{-1} \sum_{i=1}^T x_i y_i$.

Notice that, unlike the RL setting, $\hat{\theta}_T$ here is the minimizer of an empirical loss function. However, as in the case of LSTD, the computational cost of a Sherman-Morrison lemma based approach for solving the above would be of the order $O(d^2T)$. Similarly to the case of the fLSTD-SA algorithm, we update the SGD iterate θ_n using a SA scheme as follows (starting with an arbitrary θ_0),

$$\theta_n = \theta_{n-1} + \gamma_n(y_{i_n} - \theta_{n-1}^\intercal x_{i_n})x_{i_n}, \tag{46}$$

where, as before, each i_n is chosen uniformly randomly from $\{1, \ldots, T\}$, and γ_n are step-sizes chosen in advance.

Unlike fLSTD-SA which is a fixed point iteration, the above is a stochastic gradient descent procedure. Nevertheless, using the same proof template as for fLSTD-SA earlier, we can derive bounds on the computational error, i.e., the distance between θ_n and the least squares solution $\hat{\theta}_T$, both in high probability as well as expectation.

9.2 Main results

9.2.1 Assumptions

As in the case of fLSTD-SA, we make some assumptions on the step sizes, features, noise and the matrix $\bar{A}_T$:

(A1) The step sizes γ_n satisfy $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2 < \infty$.

(A2) Boundedness of x_i, i.e., $\|x_i\|_2 \leq \Phi_{\max}$, for $i = 1, \ldots, T$.

(A3) The noise $\{\xi_i\}$ is i.i.d., zero mean and $|\xi_i| \leq \sigma$, for $i = 1, \ldots, T$.

(A4) The matrix $\bar{A}_T$ is positive definite, and its smallest eigenvalue is at least $\mu > 0$.

Assumptions (A2) and (A3) are standard in the context of least squares minimization. As for fLSTD-SA, in cases when the fourth assumption is not satisfied we can employ either explicit regularization or iterate averaging to produce similar results.

9.2.2 Asymptotic convergence

An analogue of Theorem 4.1 holds as follows:

Theorem 9.1 *Under (A1)-(A4), the iterate $\theta_n \to \hat{\theta}_T$ a.s. as $n \to \infty$, where θ_n is given by* (46) *and $\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T$.*

Proof Follows in exactly the same manner as the proof of Theorem 4.1. ■

9.2.3 Finite time bounds

An analogue of Theorem 4.2 for this setting holds as follows:

Theorem 9.2 (*Error Bound for iterates of SGD*)
Assume (A1)-(A4). Choosing $\gamma_n = \frac{c_0 c}{(c+n)}$ and c such that $c_0 \in (0, \Phi_{\max}^{-1})$ and $\mu c_0 c \in (1, \infty)$, for any $\delta > 0$,

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1^{LS}}{\sqrt{n+c}} \text{ and } \mathbb{P}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2^{LS}}{\sqrt{n+c}}\right) \geq 1 - \delta,$$

where

$$K_1^{LS}(n) := \frac{\sqrt{c}\left\|\theta_0 - \hat{\theta}_T\right\|_2}{(n+c)^{\mu c_0 c - \frac{1}{2}}} + \frac{c_0 c h(n)}{2c_0 c\mu - 1},$$
$$K_2^{LS}(n) := 2c_0 c\sqrt{\frac{h(n)\log\delta^{-1}}{\mu c_0 c - 1}} + K_1(n).$$

where $h(n) := (\sigma + \|\hat{\theta}_T\|_2 + \|\theta_0\|_2 + \sigma\Phi_{\max}\ln(c+n))\Phi_{\max}^2$

Proof See Section 9.4. ■

Remark 9 (**Rates.**) With step-sizes specified in Theorem 9.2, we see that the initial error is forgotten faster than the sampling error, which vanishes at the rate $O\left(n^{-1/2}\right)$. Thus, the rate derived in Theorem 9.2 matches the asymptotically optimal convergence rate for SGD type schemes (c.f. [26]).

9.3 Iterate Averaging

The expectation and high-probability bounds in Theorem 9.2 as well as earlier works on SGD (cf. [13]) require the knowledge of the strong convexity constant μ. Iterate averaged SGD gets rid of this dependence while exhibiting the optimal convergence rates both in high probability and expectation and this claim is made precise in the following theorem.

Theorem 9.3 (***Error Bound for iterate averaged SGD***)
Under (A2)-(A3), choosing $\gamma_n = c_0\left(\frac{c}{(c+n)}\right)^{\alpha}$, *with* $\alpha \in (1/2, 1)$ *and* $c_0 \in (0, \Phi_{\max}^{-1})$, *we have, for any* $\delta > 0$,

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1^{IA}(n)}{(n+c)^{\alpha/2}} \text{ and } \mathbb{P}\left(\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2^{IA}(n)}{(n+c)^{\alpha/2}}\right) \geq 1-\delta, \quad (47)$$

where, writing $C = \sum_{n=1}^{\infty}\exp(-\mu c_0 c^{\alpha} n^{1-\alpha})(< \infty)$,

$$K_1^{IALS}(n) := \left(\|\theta_{n_0} - \theta_T\|_2 + e + \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}\exp\left(\frac{2\alpha}{1-\alpha}\right)\right)\frac{C}{(n+c)^{1-\frac{\alpha}{2}}}$$
$$+ \left(\sigma + (\sigma + \left\|\theta^*\right\|_2 + \|\theta_0\|_2 + \sigma\log(n+c))\Phi_{\max}\right)c^{\alpha}c_0\left(c_0\mu c^{\alpha}\right)^{-\alpha\frac{1+2\alpha}{2(1-\alpha)}},$$
$$\text{and } K_2^{IALS}(n) := \frac{4\sqrt{\log\delta^{-1}}}{\mu^2 c_0^2}\frac{\max\left\{\frac{1}{\mu}, 1\right\}\left\{c_0 + 2^{\alpha} + \left[\frac{2\alpha}{c^{\alpha}} + \frac{2\alpha}{1-\alpha}\right]\right\}}{(n+c)^{(1-\alpha)/2}} + K_1^{IALS}(n).$$

Proof The proof is completely analogous to that of Theorem 5.1 and hence omitted. ■

9.4 Proofs for least squares regression extension

The overall schema of the proof here is the same as that used to prove Theorem 4.2. In the following, we present an analogue of Proposition 7.1 for the least squares setting. (Recall that $\hat{\theta}_T = \bar{A}_T^{-1} b_T$ is the least squares solution):

Proposition 9.1 *Let $z_n = \theta_n - \hat{\theta}_T$, where θ_n is given by* (46)*. Under (A1)-(A4), and assuming that $\gamma_n \Phi_{\max} \leq 1$ for all n, we have $\forall \epsilon > 0$,*

*(1) a bound in **high probability** for the **centered error**:*

$$\mathbb{P}\left(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{2h(n)\sum_{i=1}^n L_i^2}\right), \tag{48}$$

where $L_i := \gamma_i \prod_{j=i}^{n-1}(1 - \gamma_{j+1}\mu(2 - \Phi_{\max}\gamma_{j+1}))^{1/2}$, and $h(k) = (\sigma + \|\hat{\theta}_T\|_2 + \|\theta_0\|_2 + \sigma\Phi_{\max}\Gamma_n)\Phi_{\max}^2$,

*(2) and a bound in **expectation** for the **non-centered error**:*

$$\mathbb{E}\left(\|z_n\|_2\right)^2 \leq \underbrace{\exp\left(-\mu\sum_{j=1}^n \gamma_j\right)\left\|\theta_0 - \hat{\theta}_T\right\|_2}_{\textit{initial error}} + \underbrace{\left(\sum_{k=1}^{n-1} 2h(k)^2\gamma_{k+1}^2 \exp\left(-2\mu\sum_{j=k+1}^n \gamma_j\right)\right)^{\frac{1}{2}}}_{\textit{sampling error}}. \tag{49}$$

The proof of the Proposition 9.1 has the same scheme as the proof of Proposition 7.1. The major difference is that the update rule is no longer the update rule of a fixed point iteration, but of a gradient descent scheme. In the following proofs, we give only the major differences with the proof of Proposition 7.1:

High-probability bound. There are two alterations to the proof of the high probability bound in Proposition 7.1: slightly different Lipschitz constants are derived according to the different form of the random innovation (Step 2 of the proof of Proposition 7.1); the constant by which the the size of the random innovations is bounded is different, and projection is not necessary to achieve this bound (Step 3 of the proof of Proposition 7.1).

Bound in expectation. The overall scheme of this proof is similar to that used in proving the expectation bound in Proposition 7.1. However, we see differences in the proof wherever the update rule is unrolled and bounds on the various quantities in the resulting expansion need to be obtained.

Proof of Proposition 9.1 part (1):

First we derive the Lipschitz dependency of the i^{th} iterate on the random innovation at time $j < i$, as in Step 2 of Proposition 7.1.

Let $\Theta_j^i(\theta)$ denote the mapping that returns the value of the iterate updated according to (46) at instant j, given that $\theta_i = \theta$. Now we note that

$$\Theta_n^i(\theta) - \Theta_n^i(\theta') = \left(I - \gamma_n x_{i_n} x_{i_n}^T\right)\left[\Theta_{n-1}^i(\theta) - \Theta_{n-1}^i(\theta')\right]$$

and

$$\left(I - \gamma_n x_{i_n} x_{i_n}^T\right)^T\left(I - \gamma_n x_{i_n} x_{i_n}^T\right) = \left(I - \gamma_n(2 - \|x_{i_n}\|_2^2\gamma_n) x_{i_n} x_{i_n}^T\right).$$

So using Jensen's inequality, the Tower property of conditional expectations, and Cauchy-Schwarz, we can deduce that

$$\begin{aligned}&\mathbb{E}\left[\|\Theta_n^i(\theta)-\Theta_n^i(\theta')\|_2 \mid \Theta_{n-1}^i(\theta),\Theta_{n-1}^i(\theta')\right]\\&\quad\leq\left[\|I-\gamma_n(2-\Phi_{\max}\gamma_n)\bar{A}_{n-1}\|_2^2\|\Theta_{n-1}^i(\theta)-\Theta_{n-1}^i(\theta')\|_2^2\right]^{1/2}\end{aligned}$$

A repeated application of this inequality together with (A4) yields the following

$$\mathbb{E}\left[\left\|\Theta_n^i(\theta)-\Theta_n^i(\theta')\right\|_2^2\right]\leq\left\|\theta-\theta'\right\|_2^2\prod_{j=i}^{n-1}(1-\mu\gamma_{j+1}(2-\Phi_{\max}\gamma_{j+1})).$$

Finally putting all this together, if f and f' denote two possible values for the random innovation at time i, and letting $\theta=\theta_{i-1}+\gamma_i f$ and $\theta'=\theta_{i-1}+\gamma_i f'$, then we have

$$\begin{aligned}&\left\|\mathbb{E}\left[\left\|\theta_n-\hat{\theta}_T\right\|_2|\theta_i=\theta\right]-\mathbb{E}\left[\left\|\theta_n-\hat{\theta}_T\right\|_2|\theta_i=\theta'\right]\right\|_2\\&\leq\mathbb{E}\left[\left\|\Theta_n^m(\theta)-\Theta_n^m(\theta')\right\|_2\right]\leq\left(\prod_{j=i}^{n-1}(1-\mu\gamma_{j+1}(2-\Phi_{\max}\gamma_{j+1}))\right)^{\frac{1}{2}}\gamma_i\left\|f-f'\right\|_2\\&=L_i\left\|f-f'\right\|_2.\end{aligned}$$

Finally we need to bound the size of the random innovations. Recall that in Proposition 7.1, the bound on the size of the iterates followed from the projection step in the algorithm. In this case, we can derive a bound directly for the iterates directly:

$$\begin{aligned}\|\theta_n\|_2&=\left\|\left[\prod_{k=1}^{n}(I-\gamma_k x_{i_k}x_k^{\mathsf{T}})\right]\theta_0+\sum_{k=1}^{t}\gamma_k\left[\prod_{j=k}^{n}(I-\gamma_j x_{i_j}x_j^{\mathsf{T}})\right]\xi_k x_k\right\|_2\\&\leq\|\theta_0\|_2+\sigma\Phi_{\max}\sum_{j=1}^{n}\gamma_j\end{aligned}\tag{50}$$

where we have used that $\gamma_j x_{i_j}x_j^{\mathsf{T}}$ is a positive definite matrix, with eigenvalues smaller than 1. Now we can bound the random innovation by

$$\left\|(y_{i_n}-\theta_{n-1}^{\mathsf{T}}x_{i_n})x_{i_n}\right\|_2\leq h(n).$$

The proof now follows just as in Proposition 7.1. ■

Proof of Proposition 9.1 part (2):

First we extract a martingale difference from the update rule (46): Let $f_n(\theta):=(\xi_{i_n}-(\theta-\hat{\theta}_T)^{\mathsf{T}}x_{i_n})\xi_{i_n}$ and let $F(\theta):=\mathbb{E}(f_n(\theta)\mid\mathcal{F}_{n-1})$, where $\mathcal{F}_{n-1}$ is the sigma field generated by the random variables $\{i_1,\ldots,\imath_{n-1}\}$ as before. Then

$$z_n=\theta_n-\hat{\theta}_T=\theta_{n-1}-\hat{\theta}_T-\gamma_n\left(F(\theta_{n-1})-\Delta M_n\right),$$

the $\Delta M_n=F(\theta_{n-1})-f_n(\theta_{n-1})$ is a martingale difference.

Now since $\hat{\theta}_T$ is the least squares solution, $F(\hat{\theta}_T) = 0$. Moreover $F(\cdot)$ is linear, and so we obtain a recursion:

$$z_n = z_{n-1} - \gamma_n \left(z_{n-1}\bar{A}_n - \Delta M_n \right) = \Pi_n z_0 - \sum_{k=1}^{n} \gamma_k \Pi_n \Pi_k^{-1} \Delta M_k,$$

where $\Pi_n := \prod_{k=1}^{n} \left(I - \gamma_k \bar{A}_k \right)$. By Jensen's inequality

$$\mathbb{E}(\|z_n\|_2) \leq (\mathbb{E}(\langle z_n, z_n \rangle))^{\frac{1}{2}} = \left(\mathbb{E}\left\|\Pi_n z_0\right\|_2^2 + \sum_{k=1}^{n} \gamma_k^2 \mathbb{E}\left\| \Pi_n \Pi_k^{-1} \Delta M_k \right\|_2^2 \right)^{\frac{1}{2}} \tag{51}$$

Notice that $\bar{A}_n - \mu I$ is positive definite by (A4) and hence

$$\begin{aligned} \left\| \Pi_n \Pi_k^{-1} \right\|_2 &= \left\| \prod_{j=k+1}^{n} \left(I - \gamma_j \bar{A}_j \right) \right\|_2 \leq \prod_{j=k+1}^{n} \left\| (1 - \gamma_j \mu) I - \gamma_j (\bar{A}_j - \mu I) \right\|_2 \\ &\leq \prod_{j=k+1}^{n} \left\| (1 - \gamma_j \mu) I \right\|_2 \leq \prod_{j=k+1}^{n} (1 - \gamma_j \mu) \leq \exp\left(-\mu \sum_{j=k}^{n} \gamma_j \right), \end{aligned} \tag{52}$$

Finally we need to bound the variance of the martingale difference. Using (A2) and (A3), a calculation shows that

$$\mathbb{E}_{\xi, i_t} \langle f_{i_t}(\theta_{t-1}), f_{i_t}(\theta_{t-1}) \rangle, \mathbb{E}_{\xi} \langle F(\theta_{t-1}, F(\theta_{t-1} \rangle \leq h(n)$$

where we have used the bound (50). Hence $\mathbb{E}[\|\Delta M_n\|_2^2] \leq 2h(n)$.

The result now follows from (51) and (52). ■

10 Fast LinUCB using SA and application to news-recommendation

10.1 Background for LinUCB

As illustrated in Fig. 6, at each iteration n, the objective is to choose an article from a pool of K articles with respective features $x_1(n), \ldots, x_K(n)$. Let x_n denote the chosen article at time n. LinUCB computes a regularised least squares (RLS) solution $\hat{\theta}_n$ based on the chosen arms x_i and rewards y_i seen so far, $i = 1, \ldots, n-1$ as follows:

$$\hat{\theta}_n = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta^\intercal x_i)^2 + \lambda \|\theta\|_2^2 . \tag{53}$$

Note that $\{x_i, y_i\}$ do not come from a distribution. Instead, at every iteration n, the arm x_n chosen by LinUCB is based on the RLS solution $\hat{\theta}_n$. The latter is used to estimate the UCB values for each of the K articles as follows:

$$\text{UCB}(x_k(n)) := x_k(n)^\intercal \hat{\theta}_n + \kappa \sqrt{x_k(n)^\intercal A_n^{-1} x_k(n)}, k = 1, \ldots, K. \tag{54}$$

The algorithm then chooses the article with the largest UCB value and the cycle is repeated.

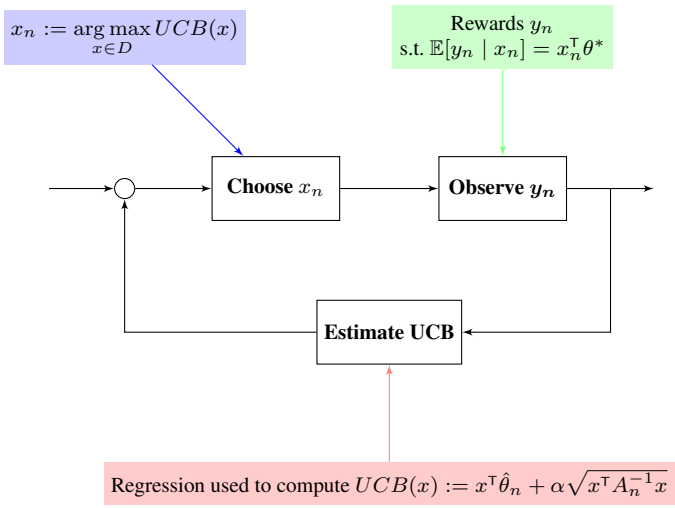


Fig. 6: Operational model of LinUCB

Algorithm 3 fLinUCB-SA

Initialisation: Set θ_0, $\lambda > 0$ - the regularization parameter, γ_k - the step-size sequence.
for $n = 1, 2, \ldots$ **do**
  Observe article features $x_1(n), \ldots, x_K(n)$
  Approximate Least Squares Regression using fLS-SA
  for $l = 1 \ldots \tau$ **do**
    Get random sample index: $i_l \sim U(\{1, \ldots, n-1\})$
    Update fLS-SA iterate $\theta_l(n)$ as follows:
    $\theta_l(n) = \theta_{l-1}(n) + \gamma_l(y_{i_l} - \theta_{l-1}(n)^\mathsf{T}x_{i_l})x_{i_l} - \gamma_l\frac{\lambda}{n}\theta_{l-1}(n)$
  end for
  UCB computation using SGD
  for $k = 1 \ldots K$ **do**
    for $l = 1 \ldots \tau'$ **do**
      Get random sample index: $i_l \sim U(\{1, \ldots, n-1\})$
      Update SGD iterate $\phi_k(n)$ as follows:
      $\phi_k(l) = \phi_k(l-1) + \gamma_l(n^{-1}x_k(n) - (\phi_k(l-1)^\mathsf{T}x_{i_l})x_{i_l})$,
    end for
  end for
  Choose article achieving $\arg\max_{k=1,\ldots,K} \theta_\tau(n)^\mathsf{T}x_k(n) + \kappa\sqrt{\phi_k(\tau')^\mathsf{T}x_k(n)}$
  Observe the reward y_n.
end for

10.2 Fast LinUCB using SA (fLinUCB-SA)

We implement a fast SGD variant of LinUCB, where SGD is used for two purposes (See Algorithm 3 for the pseudocode):

Least squares approximation. Here we use fLS-SA as a subroutine to approximate $\hat{\theta}_n$. In particular, at any instant n of the LinUCB algorithm, we run the update (46) for τ steps and use the resulting θ_τ to derive the UCB values for each arm.

UCB confidence term approximation. Here we use an SGD scheme, originally proposed in [16], for approximating the confidence term of the UCB values (54). For a given arm $k = 1, \ldots, K$, let $\hat{\phi}_k(n) = A_n^{-1}x_k(n)$ denote the confidence estimate in the UCB

value (54). Recall that $A_n = \sum_{i=1}^{n} x_i x_i^\intercal$. It is easy to see that $\hat{\phi}_k(n)$ is the solution to the following problem:

$$\min_{\phi} \sum_{i=1}^{n} \frac{(x_i^\intercal \phi)^2}{2} - \frac{x_k(n)^\intercal \phi}{n}. \tag{55}$$

Solving the above problem incurs a complexity of $O(d^2)$. An SGD alternative with a per-iteration complexity of $O(d)$ approximates the solution to (55) by using the following iterative scheme:

$$\phi_k(l) = \phi_k(l-1) + \gamma_l(n^{-1}x_k(n) - (\phi_k(l-1)^\intercal x_{i_l})x_{i_l}), \tag{56}$$

where i_l is chosen uniformly at random in the set $\{1, \ldots, n\}$.

For fLinUCB-SA in both the simulation setups presented subsequently, we set λ to 1, κ to 1, τ, τ' to 100 and θ_0 to the $d = 136$ **0** vector. Further, the step-sizes γ_k are chosen as $c/(2(c+k))$, with $c = 1.33n$ and this choice is motivated by Theorem 9.2.

Remark 10 The choice of the number of steps τ, τ' for SGD schemes in fLinUCB-SA is an arbitrary one. Our aim is simply to show that using a stochastic approximation iterates in place of an exact solution to the least squares and confidence estimates does not significantly decrease performance of LinUCB, while it does drastically decrease the complexity.

10.3 Experiments on Yahoo! dataset

The motivation in this experimental setup is to establish the usefulness of fLS-SA in a higher level machine learning algorithm such as LinUCB. In other words, the objective is to test the performance of LinUCB with SGD approximating least squares and show that the resulting algorithm gains in runtime, while exhibiting comparable performance to that of regular LinUCB.

For conducting the experiments, we use the framework provided by the ICML exploration and exploitation challenge [25], based on the user click log dataset [40] for the Yahoo! front page today module (see Fig. 7). We run each algorithm on several data files corresponding to different days in October, 2011.

Each data file has an average of nearly two million records of user click information. Each record in the data file contains various information obtained from a user visit. These include the displayed article, whether the user clicked on it or not, user features and a list of available articles that could be recommended. The precise format is described in [25]. The evaluation of the algorithms in this framework is done in an off-line manner using a procedure described in [23].

Results. We report the tracking error and runtimes from our experimental runs in Figs. 8 and 9, respectively. As in the case of fLSTDQ-SA, the tracking error is the distance in ℓ^2 norm between the fLS-SA iterate θ_n and the RLS solution $\hat{\theta}_n$ at each instant n of the LinUCB algorithm. The runtimes in Fig. 9 are for five different data files corresponding to five days in October, 2009 of the dataset [40] and compare the classic RLS solver time against fLS-SA time for each day of the dataset considered.

From Fig. 8, we observe that, in iteration $n = 165$ of the LinUCB algorithm, fLS-SA algorithm iterate $\theta_\tau(n)$ converges rapidly to the corresponding RLS solution $\hat{\theta}_n$. The

Fig. 7: The *Featured* tab in Yahoo! Today module (src: [22])

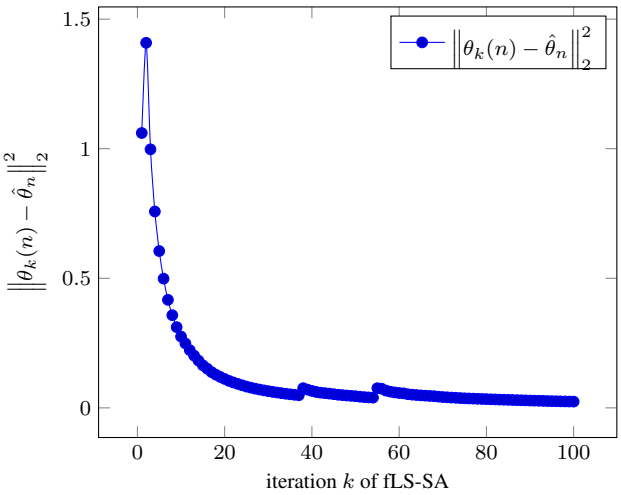


Fig. 8: Distance between fLS-SA iterate $\theta_k(n)$ and $\hat{\theta}_n$ in iteration $n = 165$ of fLinUCB-SA, with day 2's data file as input.

choice 165 for the iteration is arbitrary, as we observed similar behaviour across iterations of LinUCB.

The CTR score value is the ratio of the number of clicks that an algorithm gets to the total number of iterations it completes, multiplied by 10000 for ease of visualization. We observed that the CTR score for the regular LinUCB algorithm with day 2's data file as input was 470, while that of fLinUCB-SA was 390, resulting in about 20% loss in performance. Considering that the dataset contains very sparse features and also the fact that the rewards are binary, with a reward of 1 occurring rarely, we believe LinUCB has not seen enough data to have converged UCB values and hence the observed loss in CTR may not be conclusive.

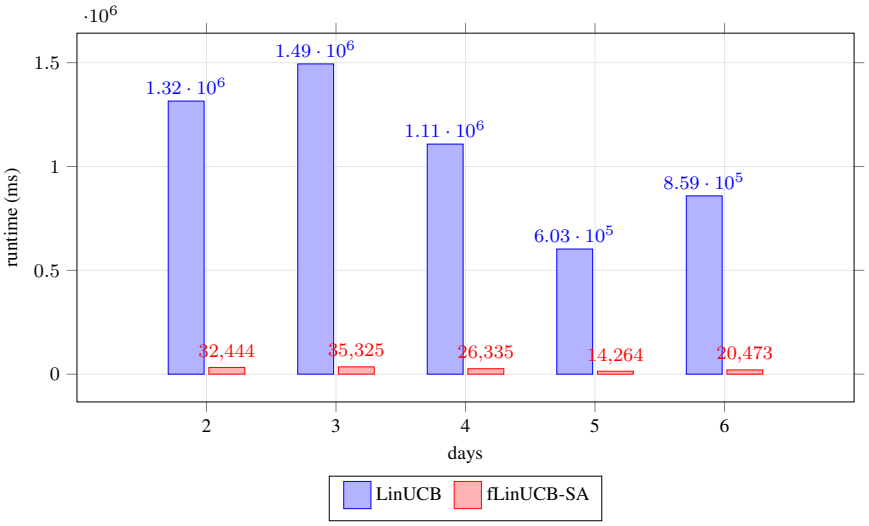


Fig. 9: Performance comparison of the algorithms using runtimes on various days of the dataset.

11 Conclusions and Future Work

We analysed a stochastic approximation based algorithm with randomised samples for policy evaluation by the method of LSTD. We provided convergence rate results for this algorithm, both in high probability and in expectation. Furthermore, we also established that using this scheme in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function and hence a low-complexity LSPI variant that uses our SA based scheme has the same order of the performance bounds as that of regular LSPI. These results coupled with the fact that the SA based scheme possesses lower computational complexity in comparison to traditional techniques makes it attractive for implementation in *big data* settings, where the feature dimension is large, regardless of the density of the feature vectors. On a traffic signal control application, we demonstrated the practicality of a low-complexity alternative to LSPI that uses our SA based scheme in place of LSTDQ for policy evaluation. We also extended our analysis for bounding the error of an SGD scheme for least squares regression and conducted a set of experiments that combines the SGD scheme with the LinUCB algorithm on a news-recommendation platform.

In [15], the authors derive concentration bounds for TD with function approximation, which is a natural extension of the work in this paper. Unlike LSTD, TD is an online algorithm and a finite-time analysis there would require notions of mixing time for Markov chains in addition to the solution scheme that we employed in this work. This is because the asymptotic limit for TD(0) is the fixed point of the Bellman operator, which assumes that the underlying MDP is begun from the stationary distribution, say Ψ. However, the samples provided to TD(0) come from simulations of the MDP that are not begun from Ψ, making the finite time analysis challenging and also implying that significant deviations from the proof technique used for fLSTD-SA are needed for analyzing TD. It would be interesting to (i) develop extensions of fLSTD-SA to approximate LSTD(λ) and (ii) choose a cyclic sampling scheme instead of the uniform random sampling. Cycling through the samples is

advantageous because the samples need not be stored and one can then think of fLSTD-SA with cyclic sampling as an incremental algorithm in the spirit of TD. Another orthogonal direction of future research is to develop online algorithms that track the corresponding batch solutions, efficiently and this has been partially accomplished in [17] and [38].

Appendix

A Proof of Lemma 7.1

Recall from the statement of Theorem 5.1 that we assume $n > n_0$, where n_0 satisfies,

$$\frac{c_0 c^\alpha}{(c+n_0)^\alpha}(1+\beta)^2\Phi_{\max}^2 < \mu.$$

Then, from the formula in Proposition 7.1, we have that:

$$\begin{aligned}
\sum_{i=1}^n L_i^2 &= \sum_{i=1}^n \left[\frac{\gamma_i}{n}\left(\sum_{l=i+1}^{n-1}\prod_{j=i}^{l}\left(1-\gamma_{j+1}(2\mu-(1+\beta)^2\Phi_{\max}^4\gamma_{j+1})\right)\right)^{1/2}\right]^2 \\
&\le \frac{1}{n^2}\sum_{i=1}^n\left[\gamma_i\left(\sum_{l=i+1}^{n-1}\exp\left(-\sum_{j=i}^{l}\gamma_{j+1}(2\mu-(1+\beta)^2\Phi_{\max}^4\gamma_{j+1})\right)\right)\right]^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n \underbrace{\left[c_0\left(\frac{c}{c+i}\right)^\alpha\left(\sum_{l=i+1}^{n-1}\exp\left(-c_0\mu\sum_{j=i}^{l}\left(\frac{c}{c+j}\right)^\alpha\right)\right)\right]^2}_{:=(A)} \qquad (57)
\end{aligned}$$

To produce the final bound, we bound the summand (A) highlighted in line (57) by a constant, uniformly over all values of i and n. Now, using the convexity of $e^{-\frac{c_0\mu}{2}x}$, followed by an Abel transform

$$\begin{aligned}
&\sum_{l=i+1}^{n-1}\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^\alpha\right) \\
&= \sum_{l=i+1}^{n-1}\left[\left(\frac{c}{c+l}\right)^\alpha\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^\alpha\right)\right]\left(\frac{c+l}{c}\right)^\alpha \\
&\le \sum_{l=i+1}^{n-1}\left[\frac{1}{c_0\mu}\left(\exp\left(-\sum_{j=1}^{l-1}\left(\frac{c}{c+i}\right)^\alpha\right)-\exp\left(-\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^\alpha\right)\right)\right]\left(\frac{c+l}{c}\right)^\alpha \\
&= \frac{1}{c_0\mu}\Bigg\{-\left(\frac{c}{c+n}\right)^{-\alpha}\exp\left(-\sum_{j=1}^{n}\left(\frac{c}{c+i}\right)^\alpha\right) \\
&\qquad +\left(\frac{c}{c+i+1}\right)^{-\alpha}\exp\left(-\sum_{j=1}^{i+1}\left(\frac{c}{c+i}\right)^\alpha\right) \\
&\qquad +\sum_{l=i+1}^{n-1}\exp\left(-\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^\alpha\right)\left[\left(\frac{c}{c+l+1}\right)^{-\alpha}-\left(\frac{c}{c+l}\right)^{-\alpha}\right]\Bigg\}
\end{aligned}$$

So the summand term (A) highlighted in line (57) can be bounded by

$$(A) \leq \frac{1}{\mu}\Bigg(\left(\frac{c+i+1}{c+i}\right)^{\alpha} + \frac{1}{(c+i)^{\alpha}} \sum_{l=i}^{n-1} \exp\left(-c^{\alpha}\frac{((c+l)^{1-\alpha}-(c+i)^{1-\alpha})}{1-\alpha}\right).((c+l+1)^{\alpha}-(c+l)^{\alpha})\Bigg)$$

Now, using convexity of x^{α} followed by comparison with an integral, and then a change of variables, we have

$$\sum_{l=i+1}^{n-1} \exp\left(-\frac{c^{\alpha}((c+l)^{1-\alpha}-(c+i)^{1-\alpha})}{(1-\alpha)}\right)((c+l+1)^{\alpha}-(c+l)^{\alpha}) \tag{58}$$

$$\leq \sum_{l=i+1}^{n-1} \exp\left(-\frac{c^{\alpha}((c+l)^{1-\alpha}-(c+i)^{1-\alpha})}{(1-\alpha)}\right)\alpha\,(c+l)^{-(1-\alpha)}$$

$$\leq \alpha \exp\left(\frac{c^{\alpha}(c+i)^{1-\alpha}}{(1-\alpha)}\right)\left[\int_{i}^{n-1} \exp\left(-\frac{c^{\alpha}(c+l)^{1-\alpha}}{(1-\alpha)}\right)(c+l)^{-(1-\alpha)}dl\right]$$

$$\leq \alpha \exp\left(\frac{c^{\alpha}(c+i)^{1-\alpha}}{(1-\alpha)}\right)\left[\int_{(c+i)^{1-\alpha}}^{(c+n-1)^{1-\alpha}} \exp\left(-\frac{c^{\alpha}l}{(1-\alpha)}\right) l^{\frac{2\alpha-1}{1-\alpha}}dl\right]. \tag{59}$$

For the second inequality we have used that the mapping $x \to e^{-d(c+x)^{1-\alpha}}(c+x)^{-(1-\alpha)}$ is decreasing in x for all $x > 1$.

By taking the derivative and setting it to zero, we find that $l \mapsto \exp\left(-\frac{c^{\alpha}l}{(1-\alpha)}\right) l^{\frac{2\alpha}{1-\alpha}}$ is decreasing on $[2\alpha/c^{\alpha}, \infty)$, and so we deduce that when $c+i \geq 2\alpha/c^{\alpha}$,

$$\exp\left(\frac{c^{\alpha}(c+i)^{1-\alpha}}{(1-\alpha)}\right)\int_{(c+i+1)^{1-\alpha}}^{(c+n)^{1-\alpha}} \exp\left(-\frac{c^{\alpha}l}{(1-\alpha)}\right) l^{\frac{2\alpha-1}{1-\alpha}}dl$$

$$\leq (c+i+1)^{2\alpha}\int_{(c+i+1)^{1-\alpha}}^{(c+n)^{1-\alpha}} l^{\frac{-1}{1-\alpha}}dl < \frac{1-\alpha}{\alpha}(c+i+1)^{\alpha}$$

When $c+i < 2\alpha/c^{\alpha}$ we can bound the summand of (58) by 1. Hence we can conclude that:

$$\sum_{i=1}^{n} L_i^2 \leq \frac{1}{\mu^2}\left\{2^{\alpha} + \left[\frac{2\alpha}{c^{\alpha}} + \frac{2\alpha}{1-\alpha}\right]\right\}^2 \frac{1}{n}. \tag{60}$$

The rest of the proof follows that of Theorem 4.2. ■

B Proof of Lemma 7.2

Recall from the statement of Theorem 5.1 that we assume $n > n_0$, where n_0 satisfies,

$$\frac{c_0 c^{\alpha}}{(c+n_0)^{\alpha}}(1+\beta)^2\Phi_{\max}^2 < \mu.$$

Recall that for this result we have chosen the larger step sizes, $\gamma_n = c_0 \left(c/(c+n)\right)^{-\alpha}$. Using that $x \mapsto x^{-2\alpha} e^{x^{1-\alpha}}$ is convex, we have

$$\begin{aligned}
\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 &\leq \exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right)\left\|\theta_0 - \hat{\theta}_T\right\|_2 \\
&\quad + \left(\sum_{k=1}^{n-1} \left(R_{\max} + (1+\beta)H\Phi_{\max}\right)^2 \right. \\
&\qquad \left. .c_0^2 \left(\frac{c}{k+c+1}\right)^{2\alpha} \exp\left(-2c_0\mu c^\alpha ((n+c)^{1-\alpha} - (k+1+c)^{1-\alpha}\right)\right)^{\frac{1}{2}} \\
&\leq \exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right)\left[\left\|\theta_0 - \hat{\theta}_T\right\|_2 \right. \\
&\quad \left. + \left(R_{\max} + (1+\beta)H\Phi_{\max}\right) c^\alpha c_0 \left\{e + \int_1^{n+c} x^{-2\alpha} \exp\left(2c_0\mu c^\alpha x^{1-\alpha}\right) dx\right\}^{\frac{1}{2}}\right] \\
&\leq \exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right) \\
&\qquad \left[\left\|\theta_0 - \hat{\theta}_T\right\|_2 + \left(R_{\max} + (1+\beta)H\Phi_{\max}\right) c^\alpha \right. \\
&\qquad\quad \left. .\left\{e + \left(\frac{c_0\mu c^\alpha}{2}\right)^{-2\alpha} . \int_1^{(n+c)(2c_0\mu c^\alpha)^{1/(1-\alpha)}} y^{-2\alpha} \exp(y^{1-\alpha}) dy\right\}^{\frac{1}{2}}\right]
\end{aligned}$$

Now, since $y^{-2\alpha} \leq \frac{2}{1-\alpha}((1-\alpha)y^{-2\alpha} - \alpha y^{-(1+\alpha)})$ when $y > \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}$, we have

$$\begin{aligned}
&\int_{\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}}^{(n+c)(2c_0\mu c^\alpha)^{1/(1-\alpha)}} y^{-2\alpha} \exp(y^{1-\alpha}) dy \\
&\quad \leq \frac{2}{1-\alpha} \int_{\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}}^{(n+c)(2c_0\mu c^\alpha)^{1/(1-\alpha)}} ((1-\alpha)y^{-2\alpha} - \alpha y^{-(1+\alpha)}) \exp(y^{1-\alpha}) dy \\
&\quad \leq \exp\left(2c_0\mu c^\alpha n^{1-\alpha}\right)(n+c)^{-\alpha}
\end{aligned}$$

and so we have that

$$\begin{aligned}
&\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \\
&\leq \exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right) \\
&\quad . \left(\|\theta_0 - \theta_T\|_2 + e\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}} + \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}} \exp\left(\frac{2\alpha}{1-\alpha}\right)\right) \\
&\qquad + \left(R_{\max} + (1+\beta)H\Phi_{\max}\right) c^\alpha c_0 \left(c_0\mu c^\alpha\right)^{-\alpha\frac{1+2\alpha}{2(1-\alpha)}} (n+c)^{-\frac{\alpha}{2}}
\end{aligned}$$

So we have

$$\begin{aligned}
\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 &\leq \left(\sum_{n=1}^{\infty} \exp\left(-c_0\mu c^\alpha (n+c)^{1-\alpha}\right)\right) \\
&\qquad . \left(\|\theta_0 - \theta_T\|_2 + e + \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}} \exp\left(\frac{2\alpha}{1-\alpha}\right)\right) \frac{1}{n} \\
&\qquad + 2\left(R_{\max} + (1+\beta)H\Phi_{\max}\right) c^\alpha \left(c_0\mu c^\alpha\right)^{-\alpha\frac{1+2\alpha}{2(1-\alpha)}} (n+c)^{-\frac{\alpha}{2}}.
\end{aligned}$$

■

References

1. 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
2. Bach F, Moulines E (2011) Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: NIPS
3. Bach F, Moulines E (2013) Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In: Advances in Neural Information Processing Systems, pp 773–781
4. Bertsekas DP (2012) Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming. Athena Scientific
5. Bertsekas DP, Tsitsiklis JN (1996) Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3), vol 7. Athena Scientific
6. Borkar V (2008) Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press
7. Borkar VS, Meyn SP (2000) The ode method for convergence of stochastic approximation and reinforcement learning. SIAM Journal on Control and Optimization 38(2):447–469
8. Bradtke S, Barto A (1996) Linear least-squares algorithms for temporal difference learning. Machine Learning 22:33–57
9. Dani V, Hayes TP, Kakade SM (2008) Stochastic linear optimization under bandit feedback. In: Proceedings of the 21st Annual Conference on Learning Theory (COLT), pp 355–366
10. Fathi M, Frikha N (2013) Transport-entropy inequalities and deviation estimates for stochastic approximation schemes. arXiv preprint arXiv:13017740
11. Frikha N, Menozzi S (2012) Concentration Bounds for Stochastic Approximations. Electron Commun Probab 17:no. 47, 1–15
12. Geramifard A, Bowling M, Zinkevich M, Sutton RS (2007) iLSTD: Eligibility traces and convergence analysis. In: NIPS, vol 19, p 441
13. Hazan E, Kale S (2011) Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In: COLT, pp 421–436
14. Konda VR (2002) Actor-critic algorithms. PhD thesis, Department of Electrical Engineering and Computer Science, MIT
15. Korda N, LA P (2015) On TD (0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In: ICML
16. Korda N, Prashanth L, Munos R (2014) Fast gradient descent for drifting least squares regression, with application to bandits. arXiv preprint arXiv:13073176v3
17. Korda N, LA P, Munos R (2015) Fast Gradient Descent for Drifting Least Squares Regression, with Application to Bandits. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp 2708–2714
18. Kushner H, Clark D (1978) Stochastic approximation methods for constrained and unconstrained systems. Springer-Verlag
19. Kushner HJ, Yin G (2003) Stochastic approximation and recursive algorithms and applications, vol 35. Springer Verlag
20. Lagoudakis MG, Parr R (2003) Least-squares policy iteration. The Journal of Machine Learning Research 4:1107–1149
21. Lazaric A, Ghavamzadeh M, Munos R (2012) Finite-sample analysis of least-squares policy iteration. Journal of Machine Learning Research 13:3041–3074
22. Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web, ACM, pp 661–670
23. Li L, Chu W, Langford J, Wang X (2011) Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, pp 297–306
24. Liu B, Liu J, Ghavamzadeh M, Mahadevan S, Petrik M (2015) Finite-Sample Analysis of Proximal Gradient TD Algorithms. In: Proc. The 31st Conf. Uncertainty in Artificial Intelligence, Amsterdam, Netherlands
25. Mary J, Garivier A, Li L, Munos R, Nicol O, Ortner R, Preux P (2012) Icml exploration and exploitation 3 - new challenges
26. Nemirovsky A, Yudin D (1983) Problem complexity and method efficiency in optimization
27. Pires BA, Szepesvári C (2012) Statistical linear estimation with penalized estimators: an application to reinforcement learning. arXiv preprint arXiv:12066444
28. Polyak BT, Juditsky AB (1992) Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization 30(4):838–855

29. Prashanth L, Bhatnagar S (2011) Reinforcement Learning with Function Approximation for Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems 12(2):412–421
30. Prashanth L, Bhatnagar S (2012) Threshold Tuning using Stochastic Optimization for Graded Signal Control. IEEE Transactions on Vehicular Technology 61(9):3865–3880
31. Robbins H, Monro S (1951) A stochastic approximation method. The annals of mathematical statistics pp 400–407
32. Ruppert D (1991) Stochastic approximation. Handbook of Sequential Analysis pp 503–529
33. Silver D, Sutton RS, Müller M (2007) Reinforcement Learning of Local Shape in the Game of Go. In: IJCAI, vol 7, pp 1053–1058
34. Sutton RS, Barto AG (1998) Reinforcement learning: An introduction. Cambridge Univ Press
35. Sutton RS, Szepesvári C, Maei HR (2009) A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. In: NIPS, pp 1609–1616
36. Sutton RS, et al (2009) Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: ICML, ACM, pp 993–1000
37. Tagorti M, Scherrer B (2015) On the Rate of Convergence and Error Bounds for LSTD(λ). In: ICML
38. Tarrès P, Yao Y (2011) Online learning as stochastic approximation of regularization paths. arXiv preprint arXiv:11035538
39. Tsitsiklis JN, Van Roy B (1997) An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control 42(5):674–690
40. Webscope Y (2011) Yahoo! Webscope dataset ydata-frontpage-todaymodule-clicks-v2_0. URL "`http://research.yahoo.com/Academic_Relations`"
41. Yu H, Bertsekas DP (2009) Convergence results for some temporal difference methods based on least squares. IEEE Transactions on Automatic Control 54(7):1515–1531
42. Zinkevich M (2003) Online convex programming and generalized infinitesimal gradient ascent. In: ICML, pp 928–925