Machine Learning (2021) 110:559–618
https://doi.org/10.1007/s10994-020-05912-5

Concentration bounds for temporal difference learning with linear function approximation: the case of batch data and uniform sampling

L. A. Prashanth[1] · Nathaniel Korda[2] · Rémi Munos[3]

Received: 4 August 2014 / Revised: 28 January 2020 / Accepted: 8 September 2020 /
Published online: 4 January 2021

Abstract

We propose a stochastic approximation (SA) based method with randomization of samples for policy evaluation using the least squares temporal difference (LSTD) algorithm. Our proposed scheme is equivalent to running regular temporal difference learning with linear function approximation, albeit with samples picked uniformly from a given dataset. Our method results in an $O(d)$ improvement in complexity in comparison to LSTD, where d is the dimension of the data. We provide non-asymptotic bounds for our proposed method, both in high probability and in expectation, under the assumption that the matrix underlying the LSTD solution is positive definite. The latter assumption can be easily satisfied for the pathwise LSTD variant proposed by Lazaric (J Mach Learn Res 13:3041–3074, 2012). Moreover, we also establish that using our method in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function. These rate results coupled with the low computational complexity of our method make it attractive for implementation in *big data* settings, where d is large. A similar low-complexity alternative for least squares regression is well-known as the stochastic gradient descent (SGD) algorithm. We provide finite-time bounds for SGD. We demonstrate the practicality of our method as an efficient alternative for pathwise LSTD empirically by combining it with the least squares policy iteration 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.

Editor: Csaba Szepesvari.

A portion of this work was done when the authors were at INRIA Lille - Nord Europe.

✉ L. A. Prashanth
prashla@cse.iitm.ac.in

Extended author information available on the last page of the article

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 Sect. 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 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 $V^\pi(s) \approx \theta^\mathsf{T}\phi(s)$, 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 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 characterizes this space (Tsitsiklis and Van Roy 1997). For a detailed treatment of this subject matter, the reader is referred to the classic textbooks (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998).

Batch reinforcement learning is a popular paradigm for policy learning. Here, we are provided with a (usually) large set of state transitions $\mathcal{D} \triangleq \{(s_i, r_i, s'_i), i = 1, \dots, T)\}$ obtained by simulating the underlying Markov decision process (MDP). For every $i = 1, \dots, T$, 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. Least squares policy iteration (LSPI) (Lagoudakis and Parr 2003) 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 least squares temporal difference (LSTD) (Bradtke and Barto 1996), 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

$$\begin{aligned} &\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T, \\ &\text{where } \bar{A}_T \triangleq \frac{1}{T}\sum_{i=1}^{T}\phi(s_i)(\phi(s_i) - \beta\phi(s'_i))^\mathsf{T}, \text{ and } \bar{b}_T \triangleq \frac{1}{T}\sum_{i=1}^{T} r_i\phi(s_i). \end{aligned} \tag{1}$$

We consider a special variant of LSTD called pathwise LSTD, proposed by Lazaric et al. (2012). 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 dataset $\mathcal{D}$ are linearly independent.

Our primary focus in this work is to solve the LSTD system in a computationally efficient manner. Solving (1) is computationally expensive, especially when d is large. For instance, in the case when $\bar{A}_T^{-1}$ is invertible, the complexity of the approach above 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 $\bar{A}_T$. An approach for solving (1) without explicitly inverting $\bar{A}_T$ is computationally expensive as well.

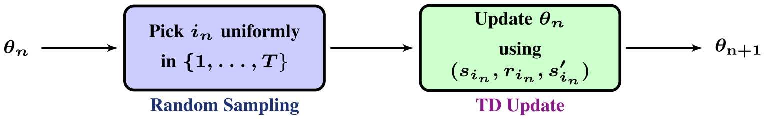

θ_n
Pick i_n uniformly in $\{1,\dots,T\}$
Random Sampling
Update θ_n using $(s_{i_n}, r_{i_n}, s'_{i_n})$
TD Update
θ_{n+1}

Fig. 1 Overall flow of the the batchTD algorithm

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. We propose the batchTD algorithm to alleviate the high computation cost of LSTD in high dimensions. The batchTD algorithm replaces the inversion of the $\bar{A}_T$ matrix by the following iterative procedure that performs a fixed point iteration (see Fig. 1 for an illustration): Set θ_0 arbitrarily and update

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

where each i_n is chosen uniformly at random from the set $\{1,\dots,T\}$, and γ_n are step-sizes that satisfy standard stochastic approximation conditions. The random sampling is sufficient to ensure convergence to the LSTD solution. The update iteration (2) is of order $O(d)$, and our bounds show that after T iterations, the iterate θ_T is very close to LSTD solution, with high probability. The advantage of the scheme above is that it incurs a computational cost of $O(dT)$, while a traditional LSTD solver based on Sherman–Morrison lemma would require $O(d^2T)$.

The update rule in (2) resembles that of TD(0) with linear function approximation, justifying the nomenclature 'batchTD'. Note that regular TD(0) with linear function approximation uses a sample path from the Markov chain underlying the policy considered. In contrast, the batchTD algorithm performs the update iteration using a sample picked uniformly at random from a dataset. We establish, through non-asymptotic bounds, that using batchTD in place of LSTD does not impact the convergence rate of LSTD to the true value function. The advantage with batchTD is the low computational cost in comparison to LSTD.

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 (1951). Iterate averaging is a standard approach to accelerate the convergence of SA schemes and was proposed independently by Ruppert (1991) and Polyak and Juditsky (1992). Non asymptotic bounds for Robbins Monro schemes have been provided by Frikha and Menozzi (2012) and extended to incorporate iterate averaging by Fathi and Frikha (2013). The reader is referred to Kushner and Yin (2003) 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 (Silver et al. 2007), with dimension $d = 10^6$, 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 a TD-like algorithm on batch data is meaningful. For other algorithms treating this complexity problem, see GTD (Sutton et al. 2009a), GTD2 (Sutton et al. 2009b), iLSTD (Geramifard et al.

2007) and the references therein. In particular, iLSTD is suitable for settings where the features admit a sparse representation.

In the context of improving the complexity of LSTD, our contributions can be summarized as follows: First, through finite sample bounds, we show that our batchTD algorithm (2) converges to the pathwise LSTD solution at the optimal rate of $O(n^{-1/2})$ in expectation (see Theorem 4.2 in Sect. 4). By projecting the iterate (2) onto a compact and 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 batchTD 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 the symmetric part of $\bar{A}_T$. We overcome the latter dependence on the knowledge of the minimum eigenvalue through iterate averaging. As an aside, we note that using completely parallel arguments to those used in arriving at non-asymptotic bounds for batchTD, one could derive bounds for the regular TD algorithm with linear function approximation, albeit for the special case when the underlying samples arrive in an i.i.d. fashion. Second, through a performance bound, we establish that using our batchTD algorithm in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function.

Third, we investigate the rates when larger step sizes ($\Theta(n^{-\alpha})$ where $\alpha \in (1/2, 1)$) are used in conjunction with averaging of the iterates, i.e., the well known Polyak-Ruppert averaging scheme. The rate obtained in high probability for the iterate-averaged variant is of the order $O(n^{-\alpha/2})$, with the added advantage that, unlike non-averaged case, the step-size choice does not require knowledge of the minimum eigenvalue of the symmetric part of $\bar{A}_T$. Further, with iterate averaging the complexity of the algorithm stays at $O(d)$ per iteration, as before. Fourth, we consider a traffic control application, and implement a variant of LSPI which uses the batchTD algorithm in place of LSTD. In particular, for the experiments we employ step-sizes that were used to derive the non-asymptotic bounds mentioned above. We demonstrate that running batchTD 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.

We now turn our attention to solving least squares regression problems via the popular stochastic gradient descent (SGD) method. 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 ML literature as SGD algorithms.

We also analyze the low-complexity SGD alternative for the classic least squares parameter estimation problem. Using the same template as for the results of batchTD, we derive non-asymptotic bounds, which hold 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. We describe a fast variant of the LinUCB (Li et al. 2010) 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 (Webscope 2011).

We observe that, using the step-size suggested by our bounds, 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 batchTD 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) (Lagoudakis and Parr 2003), and linear bandits (Dani et al. 2008; Li et al. 2010), which use LSTD and regression in their inner loops.

The rest of the paper is organized as follows: In Sect. 2, we discuss related work. In Sect. 2.2 we present the batchTD algorithm, and in Sect. 4 we provide the non-asymptotic bounds for this algorithm. In Sect. 5, we analyze a variant of our algorithm that incorporates iterate averaging. In Sect. 6, we compare our bounds to those in recent work. In Sect. 7, we describe a variant of LSPI that uses batchTD in place of LSTD. Next, in Sect. 8, we provide detailed proofs of convergence, and derivation of rates. We provide experiments on a traffic signal control application in Sect. 9. In Sect. 10, we provide extensions to solve the problem of least squares regression and in Sect. 11, we provide a set of experiments that tests a variant of the LinUCB algorithm using a SGO subroutine for least squares regression. Finally, in Sect. 12 we provide the concluding remarks.

2 Literature review

2.1 Previous work related to LSTD

In Chapter 6 of Konda (2002), the authors establish that LSTD has the optimal asymptotic convergence rate, while by Antos et al. (2008) and Lazaric et al. (2012), the authors provide a finite time analysis for LSTD and LSPI. Recent work by Tagorti and Scherrer (2015) provides sample complexity bounds for LSTD(λ). LSPE(λ), which is an algorithm that is closely related to LSTD(λ), is analyzed by Yu and Bertsekas (2009). The authors there provide asymptotic rate results for LSPE(λ), and show that it matches that of LSTD(λ). Also related is the work by Pires and Szepesvári (2012), where the authors study linear systems in general, and as a special case, provide error bounds for LSTD with improved dependence on the underlying feature dimension.

A closely related contribution that is geared towards improving the computational complexity of LSTD is iLSTD (Geramifard et al. 2007). 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. Another line of related previous work is GTD (Sutton et al. 2009a), and its later enhancement GTD2 (Sutton et al. 2009b). The latter algorithms feature an update iteration that can be viewed as gradient descent and operate in the online setting similar to the regular TD algorithm with function approximation. However, the advantage with GTD/GTD2 is that these algorithms 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 by Liu et al. (2015) 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. This 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 have not been proposed before in the literature. The high probability bounds that we derive for batchTD do not directly follow from earlier work on LSTD algorithms. Concentration bounds for SA schemes have been derived by Frikha and Menozzi (2012). While we use their technique for proving the high-probability bound on batchTD 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 the symmetric part of $\bar{A}_T$, we depart from the argument by Frikha and Menozzi (2012).

Finite sample analysis of TD with linear function approximation has received more attention in recent works (cf. Dalal et al. 2018; Bhandari et al. 2018; Lakshminarayanan and Szepesvari 2018). A detailed comparison of our bounds to those in the aforementioned references is provided in Sect. 6.

This paper is an extended version of an earlier work (see Prashanth et al. 2014). This work corrects the errors in the earlier work by using significant deviations in the proofs, and includes additional simulation experiments. Finally, by Narayanan and Szepesvári (2017), the authors list a few problems with the results and proofs in the conference version (Prashanth et al. 2014), and the corrections incorporated in this work address the comments by Narayanan and Szepesvári (2017).

2.2 Previous work related to SGD

Finite time analysis of SGD methods have been provided by Bach and Moulines (2011). While the bounds by Bach and Moulines (2011) 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 by Zinkevich (2003); Hazan and Kale (2011). The gradient descent algorithm by Zinkevich (2003) is in the setting of optimising the average of convex loss functions whose gradients are available, while that by Hazan and Kale (2011) is for strongly convex loss functions.

In comparison to previous work w.r.t. least squares regression, we highlight the following differences:

Earlier works on strongly convex optimization (cf. Hazan and Kale 2011) 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 $\bar{A}_T$ have 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 non-asymptotic bounds, both in high probability and in expectation, for iterate-averaged SGD.

Our analysis is for the classic SGD scheme that is anytime, whereas the epoch-GD algorithm by Hazan and Kale (2011) requires the knowledge of the time horizon.

While the algorithm by Bach and Moulines (2013) 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 Bach and Moulines (2011) 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.

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 TD with uniform sampling on batch data (batchTD)

We propose here a stochastic approximation variant of the 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 batchTD, is a simple stochastic approximation scheme that updates incrementally using samples picked uniformly from batch data. The results that we present establish that the batchTD algorithm 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 the batchTD algorithm.

In the following section, we provide a brief background on LSTD and pathwise LSTD. In the subsequent section, we present our batchTD algorithm.

3.1 Background

Consider an MDP with state space $\mathcal{S}$ and action space $\mathcal{A}$, both assumed to be finite. Let $p(s,a,s'), s,s' \in \mathcal{S}, a \in \mathcal{A}$ denote the probability of transitioning from state s to s' on action 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

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

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

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

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 $s \in \mathcal{S}$, approximate $V^\pi(s) \approx \phi(s)^\mathsf{T}\theta$, where $\phi(s)$ is a d-dimensional feature vector for state s with $d \ll |\mathcal{S}|$, and θ is a tunable parameter. With this approach, the idea is to find the best approximation to the value function V^{π} in $\mathcal{B} = \{\Phi\theta \mid \theta \in \mathbb{R}^d\}$, which is a vector subspace of $\mathbb{R}^{|S|}$. In this setting, it is no longer feasible to find the fixed point $V^{\pi} = \mathcal{T}^{\pi} V^{\pi}$. Instead, one can approximate V^{π} within $\mathcal{B}$ by solving the following projected system of equations:

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

In the above, Φ denotes the feature matrix with rows $\phi(s)^{\mathsf{T}}, \forall s \in \mathcal{S}$, and Π is the orthogonal projection onto $\mathcal{B}$. Assuming that the matrix Φ has full column rank, it is easy to derive that $\Pi = \Phi(\Phi^{\mathsf{T}}\Psi\Phi)^{-1}\Phi^{\mathsf{T}}\Psi$, where Ψ is the diagonal matrix whose diagonal elements form the stationary distribution (assuming it exists) of the Markov chain associated with the policy π.

The solution θ^* of (5) can be re-written as follows (cf. Bertsekas 2012, Section 6.3):

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

where $P = [P(s,s')]_{s,s' \in \mathcal{S}}$ is the transition probability matrix with components $P(s,s') = \sum_{a \in \mathcal{A}} \pi(s,a)p(s,a,s')$, $\mathcal{R}$ 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.

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, $\mathcal{D} = \{(s_i, r_i, s'_i), i = 1, \dots, T\}$, 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:

$$\begin{aligned} &\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T, \\ &\text{where } \bar{A}_T \triangleq \frac{1}{T}\sum_{i=1}^{T} \phi(s_i)(\phi(s_i) - \beta\phi(s'_i))^{\mathsf{T}}, \text{ and } \bar{b}_T \triangleq \frac{1}{T}\sum_{i=1}^{T} r_i\phi(s_i). \end{aligned} \tag{7}$$

Denoting the current state feature $(T \times d)$-matrix by $\Phi \triangleq (\phi(s_1)^{\mathsf{T}}, \dots, \phi(s_T))$, next state feature $(T \times d)$-matrix by $\Phi' \triangleq (\phi(s'_1)^{\mathsf{T}}, \dots, \phi(s'_T))$, and reward $(T \times 1)$-vector by $\mathcal{R} = (r_1, \dots, r_T)^{\mathsf{T}}$, we can rewrite $\bar{A}_T$ and $\bar{b}_T$ as follows[1]:

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

It is not clear whether $\bar{A}_T$ is invertible for an arbitrary dataset $\mathcal{D}$. One way to ensure invertibility is to adopt the approach of pathwise LSTD, proposed by Lazaric et al. (2012). The pathwise LSTD algorithm 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 $i = 1, \dots, T-1$. 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 by Lazaric et al. (2012), 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 Lazaric et al. (2012).

[1] 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.

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

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

Let $\hat{\mathcal{R}}$ be a $T \times 1$ vector with entries r_i, $i = 1, \dots, T$ 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 $\mathcal{G}_T \triangleq \{(\phi(s_1)^\mathsf{T}\theta, \dots, \phi(s_T)^\mathsf{T}\theta)^\mathsf{T} \mid \theta \in \mathbb{R}^d\} \subset \mathbb{R}^T$ be the vector sub-space of $\mathbb{R}^T$ within which pathwise LSTD approximates the true values of the value function corresponding to the states $s_1, \dots, s_T$, and it is the empirical analogue of $\mathcal{B}$ defined earlier. 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: $\|f\|_T^2 \triangleq T^{-1} \sum_{i=1}^T f(s_i)^2$, for any function f. Notice that $\hat{\Pi}\hat{T}$ is a contraction mapping, since

$$\left\|\hat{\Pi}\hat{T}y - \hat{\Pi}\hat{T}z\right\|_T \le \left\|\hat{T}y - \hat{T}z\right\|_T = \beta\left\|\hat{\mathcal{V}}y - \hat{\mathcal{V}}z\right\|_T \le \beta\|y - z\|_T.$$

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

Suppose 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. 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

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

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

Remark 1 (Regular versus Pathwise LSTD) For a large dataset $\mathcal{D}$ 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 $\frac{1}{T}$. As mentioned earlier, zeroing out the next state in the last transition of $\mathcal{D}$ together with a full-rank Φ makes the system of equations in (7) well-posed. As an aside, the batchTD algorithm, which we describe 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 the batchTD algorithm

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, in the case of pathwise LSTD, the dataset corresponds to those along a sample path simulated from the underlying MDP for a given policy π, i.e., $s'_i = s_{i+1}$, $i = 1, \dots, T-1$ and $s'_T = 0$.

The full pseudocode for batchTD is given in Algorithm 1. Starting with an arbitrary θ_0, we update the parameter θ_n as follows:

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

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, i.e., $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2 < \infty$. The operator Υ projects the iterate θ_n onto the nearest point in a closed ball $\mathcal{C} \subset \mathbb{R}^d$ with a radius H that is large enough to include $\hat{\theta}_T$. Note that projection via Υ amounts to scaling down the ℓ_2-norm of the iterate θ_n so that it does not exceed H, and is a computationally inexpensive operation.

In the next section, we present non-asymptotic bounds for the error $\left\|\theta_n - \hat{\theta}_T\right\|_2$ that hold with high probability, and in expectation, for the projected iteration in (10). Further, we also provide an error bound that holds in expectation for a variant of (10) without involving the projection operation. From the bounds presented below, we can infer that, for a step size choice that is inversely proportional to the number n of iterations, obtaining the optimal $O\left(1/\sqrt{n}\right)$ requires the knowledge of the minimum eigenvalue μ of $\frac{1}{2}\left(\bar{A}_T + \bar{A}_T^{\mathsf{T}}\right)$, where $\bar{A}_T$ is a matrix made from the features used in the linear approximation (see assumption (A1) below). Subsequently, in Sect. 5, we present non-asymptotic bounds for a variant of the batchTD algorithm, which employs iterate averaging. The bounds for iterate-averaged batchTD establish that the knowledge of eigenvalue μ is not needed to obtain a rate of convergence that can be made arbitrarily close to $O\left(1/\sqrt{n}\right)$.

Algorithm 1 The batchTD algorithm

```
Input: Sample path-based dataset $\mathcal{D} \triangleq \{(s_i, r_i, s'_i), i = 1, \ldots, T)\}$ such that $s'_i = s_{i+1}$, $i = 1, \ldots, T-1$ and $s'_T = 0$; a choice of step-size sizes, $\gamma_k$; a time horizon $n$.
Initialization: Set $\theta_0$.
Run:
for $k = 1 \ldots n$ do
    Get a random sample index: $i_k \sim U(\{1, \ldots, T\})$.
    Perform update iteration: $\theta_k = \Upsilon\left(\theta_{k-1} + \gamma_k\left(r_{i_k} + \beta\theta_{k-1}^{\mathsf{T}}\phi(s'_{i_k}) - \theta_{k-1}^{\mathsf{T}}\phi(s_{i_k})\right)\phi(s_{i_k})\right)$.
end for
Output: $\theta_n$
```

4 Main results for the batchTD algorithm

Map of the results: Theorem 4.1 proves almost sure convergence of batchTD iterate θ_n to LSTD solution $\hat{\theta}_T$, with and without projection. 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 $\boldsymbol{\Phi}\theta_n$ to the true value function, i.e., a variant of Theorem 1 by Lazaric et al. (2012) for the case of the batchTD algorithm. Theorem 4.5 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 batchTD 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 Sects. 4.2–4.5 and provide detailed proofs of all the claims in Sect. 8. Also, all the results are by default for the projected version of the batchTD algorithm, i.e., θ_n given by (10), while Sect. 4.4 presents the results for the projection-free batchTD variant. In particular, the latter section provides both asymptotic convergence and a bound in expectation for the error $\|\theta_n - \hat{\theta}_T\|_2$ for the projection-free variant of batchTD.

4.1 Assumptions

We make the following assumptions for the analysis of the batchTD algorithm:

- **(A1)** The matrix $\bar{A}_T$ is positive definite, which implies the smallest eigenvalue μ of its symmetric part $\frac{1}{2}\left(\bar{A}_T + \bar{A}_T^\mathsf{T}\right)$ is greater than zero.[2]
- **(A2)** Bounded features: $\left\|\phi(s_i)\right\|_2 \leq \Phi_{\max} < \infty$, for $i = 1, \dots, T$.
- **(A3)** Bounded rewards: $|r_i| \leq R_{\max} < \infty$ for $i = 1, \dots, T$.
- **(A4)** The set $\mathcal{C} \triangleq \{\theta \in \mathbb{R}^d \mid \|\theta\|_2 \leq H\}$ used for projection through Υ satisfies $H > \frac{\|\bar{b}_T\|_2}{\mu}$, where μ is as defined in (A1).

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 that the matrix $\bar{A}_T$ is positive definite (see (A1)). For pathwise LSTD, (A1) can be replaced by the following assumption:

- **(A1')** The matrix Φ is full rank.

Recall that the pathwise LSTD algorithm perturbs the data set slightly, as discussed in Sect. 3.1 above. Thus, from (9), we have

$$\mu \geq \frac{(1-\beta)}{T}\mu', \text{ where } \mu' \triangleq \lambda_{\min}(\Phi^\mathsf{T}\Phi). \tag{11}$$

The inequality above holds because $\left\|\hat{P}v\right\|_2 \leq \|v\|_2$, and $\left\|\hat{P}^\mathsf{T}v\right\|_2 \leq \|v\|_2$, leading to the fact that $\lambda_{\min}\left(I - \frac{\beta}{2}\left(\hat{P} + \hat{P}^\mathsf{T}\right)\right) \geq (1-\beta)$. Thus, it is easy to infer that (A1') implies (A1), using (11) in conjunction with the fact that a full rank Φ implies $\mu' > 0$.

Note that the dataset is assumed to be fixed for all the results presented below.

4.2 Asymptotic convergence

Theorem 4.1 *Assume (A1)–(A4), and also that the step sizes $\gamma_n \in \mathbb{R}_+$ satisfy $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2 < \infty$. Then, for the iterate θ_n updated according to (10), we have*

$$\theta_n \to \hat{\theta}_T \text{ a.s. as } n \to \infty. \tag{12}$$

Proof See Sect. 8.1. □

[2] A real matrix A is positive definite if and only if the symmetric part $\frac{1}{2}(A + A^\mathsf{T})$ is positive definite.

4.3 Non-asymptotic bounds

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

Theorem 4.2 (Error bounds for batchTD) *Assume (A1)–(A4). Set* $\gamma_n = \frac{c_0 c}{(c+n)}$ *such that* $c_0 \in (0, \mu((1+\beta)^2 \Phi^4_{\max})^{-1}]$ *and* $c_0 c > \frac{1}{\mu}$. *Then, for any* $\delta > 0$, *we have*

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1(n)}{\sqrt{n+c}}, \quad \text{and} \tag{13}$$

$$\mathbb{P}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2(n)}{\sqrt{n+c}}\right) \geq 1 - \delta. \tag{14}$$

In the above, $K_1(n)$ *and* $K_2(n)$ *are functions of order O(1), defined by*[3]:

$$K_1(n) \triangleq \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2 \sqrt{(c+1)^{c_0 c \mu}}}{\sqrt{(n+c)^{c_0 c \mu - 1}}} + \frac{2ec_0 c\left(R_{\max} + (1+\beta)H\Phi^2_{\max}\right)}{\sqrt{2c_0 c\mu - 1}}, \text{ and}$$

$$K_2(n) \triangleq 2\sqrt{e} c_0 c\left(R_{\max} + (1+\beta)H\Phi^2_{\max}\right)\sqrt{\frac{\log \delta^{-1}}{c_0 c\mu - 1}} + K_1(n).$$

Proof See Sect. 8.2. □

A few remarks are in order.

Remark 2 (Initial versus sampling error) The bound in expectation above can be re-written as

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2 \sqrt{(c+1)^{c_0 c\mu}}}{(n+c)^{c_0 c\mu/2}} + \frac{2ec_0 c\left(R_{\max} + (1+\beta)H\Phi^2_{\max}\right)}{\sqrt{2c_0 c\mu - 1}\sqrt{n+c}}. \tag{15}$$

The first term on the RHS above is the initial error, while the second term is the 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. From (15), it is evident that the initial error is forgotten at the rate $O\left(\frac{1}{n^{c_0 c\mu/2}}\right)$. Since $c_0 c\mu > 1$, the former rate is faster than the rate $O(1/\sqrt{n})$ at which the sampling error decays.

Remark 3 (Rate dependence on the minimum eigenvalue μ) 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:

[3] For notational convenience, we have chosen to ignore the dependence of K_1 and K_2 on the confidence parameter δ.

$$K_1(n) = \frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2 \sqrt{(c+1)^{\eta}}}{\sqrt{(n+c)^{(\eta-1)}}} + \frac{2e\eta}{\mu\sqrt{(2\eta-1)}}\left(R_{\max} + (1+\beta)H\Phi^2_{\max}\right), \text{ and}$$

$$K_2(n) = 2\sqrt{e}\frac{\eta}{\mu}\left(R_{\max} + (1+\beta)H\Phi^2_{\max}\right)\sqrt{\frac{\log\delta^{-1}}{(\eta-1)}} + K_1(n).$$

So both the bounds in expectation and high probability have a linear dependence on the reciprocal of μ. Note also that the constant $(R_{\max} + (1+\beta)H\Phi^2_{\max})$ is nothing more than a bound on the size of the random innovations made by the algorithm at each time step.

Remark 4 (Eigenvalue dependence on β) Notice that the eigenvalue μ is implicitly dependent on β:

$$\mu \triangleq \frac{1}{2}\lambda_{\min}(\bar{A}_T + \bar{A}_T^{\top}) = \frac{1}{2T}\lambda_{\min}\left(2\Phi^{\top}\Phi - \beta\left(\Phi'^{\top}\Phi + \Phi^{\top}\Phi'\right)\right).$$

Clearly, as β increases, it is harder to satisfy the assumption that $\mu > 0$. Moreover, for pathwise LSTD (see Sect. 3.1), the inequality in (11) underlines an *implicit* linear dependence of the rates on the reciprocal of $(1-\beta)$. However, the bounds' exact sensitivity to this reciprocal is data-dependent.

Remark 5 (Regularization) To obtain the best performance from the batchTD algorithm, we need to know the value of μ. However, we can get rid of this dependency easily by explicitly regularizing the problem. In other words, instead of the LSTD solution (7), we obtain the following regularized variant:

$$\hat{\theta}_T^{reg} = (\bar{A}_T + \mu I)^{-1}\bar{b}_T, \tag{16}$$

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

$$\theta_n^{reg} = (1-\gamma_n\mu)\theta_{n-1} + \gamma_n\left(r_{i_n} + \beta\theta_{n-1}^{\top}\phi(s'_{i_n}) - \theta_{n-1}^{\top}\phi(s_{i_n})\right)\phi(s_{i_n}). \tag{17}$$

This algorithm retains all the properties of the non-regularized batchTD algorithm, except that it converges to the solution of (16) rather than to that of (7). In particular, the conclusions of Theorem 4.2 hold without requiring assumption (A1), but measuring $||\theta_n - \hat{\theta}_T^{reg}||_2$, the error to the regularized fixed point $\hat{\theta}_T^{reg}$.

Remark 6 (Computational complexity) Our theoretical results in Theorem 4.2 show that, with probability $1-\delta$, batchTD constructs an ϵ-approximation of the pathwise LSTD solution with $O(d\ln(1/\delta)/\epsilon^2)$ complexity. In other words, for the batchTD estimate to be within a distance $\epsilon > 0$ of the LSTD solution, the number of iterations of (10) would be proportional to $\frac{d\ln(1/\delta)}{\epsilon^2}$. This observation coupled with the fact that each iteration of (10) is of order $O(d)$ establishes the advantage of batchTD over pathwise LSTD from a time-complexity viewpoint.

However, batchTD requires storing the entire dataset for the purpose of random sampling. To reduce the storage requirement of batchTD, one could uses mini-batching of the dataset, i.e., store smaller subsets of the dataset and run batchTD updates on these mini-batches. It is an interesting direction for future work to analyze such an approach and recommend appropriate mini-batch sizes based on the parameters of the underlying policy evaluation problem. For the case of regression, such an approach has been recommended in earlier works, cf. Roux et al. (2012).

Remark 7 (TD with linear function approximation) One could use completely parallel arguments to that in the proof of Theorem 4.2 to obtain rate results for TD(0) with linear function approximation under i.i.d. samples. A similar observation holds for the bounds presented below for the projection-free variant of batchTD in Theorem 4.4, and for the iterate-averaged variant of batchTD in Theorem 5.1.

The bounds for TD with linear function approximation under i.i.d. sampling would be a side benefit, while the primary message from our work is that one could run TD(0) on a batch, and obtain a computational advantage, with performance comparable to that of LSTD. We have used pathwise LSTD to drive home this point.

Finally, note that the regular TD with linear function approximation is under non i.i.d. sampling (or involving a Markov noise component), and deriving non-asymptotic bounds for such a setting is beyond the scope of this paper.

4.4 Projection-free variant of the batchTD algorithm

Here we consider a projection-free variant of batchTD that updates according to (10), but with $\Upsilon(\theta) = \theta$, $\forall \theta \in \mathbb{R}^d$. We now present the results for batchTD without a non-trivial projection, under assumptions similar to the projected variant of batchTD, i.e., bounded rewards, features, and a positive lower bound on the minimum eigenvalue μ of the symmetric part of $\bar{A}_T$. The results include asymptotic convergence and a bound in expectation on the error $\|\theta_n - \hat{\theta}_T\|_2$. 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.

Theorem 4.3 *Assume (A1)–(A3), and also that the step sizes $\gamma_n \in \mathbb{R}_+$ satisfy $\sum_n \gamma_n = \infty$, and $\sum_n \gamma_n^2 < \infty$. Then, for the iterate θ_n updated according to (10) without projection (i.e., Υ is the identity map), we have*

$$\theta_n \to \hat{\theta}_T \text{ a.s. as } n \to \infty. \tag{18}$$

Proof See Sect. 8.2. □

Using a slightly different proof technique, we are able to give a bound in expectation for the error of the non-projected batchTD, in the result below.

Theorem 4.4 (Expectation error bound for batchTD without projection) *Assume (A2)–(A4). Set* $\gamma_n = \frac{c_0 c}{(c+n)}$ *such that* $c_0 \in (0, \mu((1+\beta)^2 \Phi^4_{\max})^{-1}]$ *and* $c_0 c\mu \in (1, \infty)$. *Then, we have*

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \le \frac{K_1(n)}{\sqrt{n+c}}, \tag{19}$$

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

$$K_1(n) \triangleq \frac{\sqrt{3}\left\|\theta_0 - \hat{\theta}_T\right\|_2 \sqrt{(c+1)^{c_0 c\mu}}}{\sqrt{(n+c)^{c_0 c\mu - 1}}} + \frac{2\sqrt{3} e c_0 c\left(R_{\max} + (1+\beta)\left\|\hat{\theta}_T\right\|_2 \Phi^2_{\max}\right)}{\sqrt{2c_0 c\mu - 1}}.$$

Proof See Sect. 8.3. □

4.5 Performance bound

We can combine our error bounds above with the performance bound derived by Lazaric et al. (2012) for pathwise LSTD. The theorem below shows that using batchTD in place of pathwise LSTD does not impact the overall convergence rate.

Theorem 4.5 (Performance bound) *Let* $\tilde{v}_n \triangleq \Phi\theta_n$ *denote the approximate value function obtained after n steps of batchTD, and let v denote the true value function, evaluated at the states* $s_1, \dots, s_T$ *along the sample path. Then, under the assumptions (A1)–(A4), with probability* $1 - 2\delta$ *(taken w.r.t. the random path sampled from the MDP, and the randomization in batchTD), we have*

$$\|v - \tilde{v}_n\|_T \le \underbrace{\frac{\|v - \Pi v\|_T}{\sqrt{1-\beta^2}}}_{\textbf{approximation error}} + \underbrace{\frac{\beta R_{\max}\Phi_{\max}}{(1-\beta)}\sqrt{\frac{d}{\mu'}}\left(\sqrt{\frac{8\ln\frac{2d}{\delta}}{T}} + \frac{1}{T}\right)}_{\textbf{estimation error}} + \underbrace{\frac{\Phi_{\max} K_2(n)}{\sqrt{n+c}}}_{\textbf{computational error}}. \tag{20}$$

where $\|f\|^2_T \triangleq \frac{1}{T}\sum_{i=1}^{T} f(s_i)^2$, *for any function f and* μ' *is the minimum eigenvalue of* $\frac{1}{T}\Phi^\top\Phi$ *(see also* (11)*).*

Proof The result follows by combining Theorem 4.2 above with Theorem 1 of Lazaric et al. (2012) using a triangle inequality. □

Remark 8 The approximation and estimation errors (first and second terms in the RHS of (20)) are artifacts of function approximation and least squares methods, respectively. The third term is a consequence of using batchTD 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.

Further, the performance bound in Theorem 4.5, considering only the dimension d, minimum eigenvalue μ and sample size T, is of the order $O\left(\frac{\sqrt{d}}{\mu\sqrt{T}}\right)$, which is better than the order $O\left(\frac{d}{\mu T^{1/4}}\right)$ on-policy performance bound for GTD/GTD2 in Proposition 4 of Liu et al. (2015).

Remark 9 *(Generalization bounds)* While Theorem 4.5 holds for only states along the sample path $s_1, \dots, s_T$, it is possible to generalize the result to hold for states outside the sample path. This approach has been adopted by Lazaric et al. (2012) 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 (see Lemma 4 by Lazaric et al. (2012)). In the light of the result in Theorem 4.5 above, it is straightforward to provide generalization bounds similar to Theorems 5 and 6 of Lazaric et al. (2012) for batchTD as well, and the resulting rates from these generalization bound variants for batchTD are the same as that for regular LSTD. We omit these obvious generalizations, and refer the reader to Section 5 of Lazaric et al. (2012) 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 (A1) in Sect. 4) to obtain the (optimal) approximation error of order $O(n^{-1/2})$. Introduced independently by Ruppert (1991) and Polyak and Juditsky (1992), the idea here is to use a larger step-size $\gamma_n \triangleq c_0(c/(c+n))^\alpha$, and then use the averaged iterate, defined as follows:

$$\bar{\theta}_n \triangleq \frac{1}{n+1} \sum_{i=0}^{n} \theta_i, \tag{21}$$

where θ_n is the iterate of the batchTD algorithm, presented earlier. The following result bounds the the distance of the averaged iterate to the LSTD solution.

Theorem 5.1 (Error Bound for iterate averaged batchTD) *Assume (A1)–(A4). Set* $\gamma_n = c_0\left(\frac{c}{c+n}\right)^\alpha$, *with* $\alpha \in (1/2, 1)$ *and* $c, c_0 > 0$. *Then, for any* $\delta > 0$, *and any* $n > n_0 \triangleq \max\{\lfloor\left(\frac{2c_0(1+\beta^2)\Phi^4_{\max}}{\mu})^{1/\alpha} - 1\right)c\rfloor, 0\}$, *we have*

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1^{IA}(n)}{(n+c)^{\alpha/2}}, \quad \text{and} \tag{22}$$

$$\mathbb{P}\left(\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2^{IA}(n)}{(n+c)^{\alpha/2}}\right) \geq 1 - \delta, \tag{23}$$

where

$$\begin{aligned}
K_1^{IA}(n) \triangleq {} & C_0\left[C_1C_2\left\|\theta_0-\hat{\theta}_T\right\|_2+\sqrt{e}\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{2(1-\alpha)}}\right.\\
&\left.+\underbrace{2c_0C_1C_2\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}_{\max}^2\right)\sqrt{n_0}}_{(E1)}\right]\frac{1}{(n+1)(n+c)^{-\frac{\alpha}{2}}}\\
&+\underbrace{\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}_{\max}^2\right)c^\alpha c_0\left(2c_0\mu c^\alpha\right)^{\frac{\alpha}{2(1-\alpha)}}}_{E2},
\end{aligned}$$

$$C_0 \triangleq \sum_{n=1}^{\infty}\exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right),\ C_1 \triangleq \exp\left(2c_0(1+\beta)\boldsymbol{\Phi}_{\max}^2(n_0+1)\right),$$

$$C_2 \triangleq \exp\left(c_0\mu c^\alpha(n_0+c+1)^{1-\alpha}\right), \text{and}$$

$$\begin{aligned}
K_2^{IA}(n) \triangleq {} & \left\{\underbrace{\frac{4\sqrt{\log\delta^{-1}}}{\mu^2c_0^2}\frac{1}{\mu}\left[2^\alpha+\left[\frac{2\alpha}{c_0\mu c^\alpha}\right]^{\frac{1}{1-\alpha}}+\frac{2(1-\alpha)(c_0\mu)^\alpha}{\alpha}\right]}_{(E3)}\right.\\
&\left.+\underbrace{\frac{\sqrt{n_0}e^{(1+\beta)\boldsymbol{\Phi}_{\max}^2c_0(2n_0+1)}}{(1+\beta)\boldsymbol{\Phi}_{\max}^2(n+1)}}_{(E4)}\right\}\frac{1}{(n+1)(n+c)^{-\frac{\alpha}{2}}}+K_1^{IA}(n).
\end{aligned}$$

Proof The proof of both the high probability bound as well as the bound in expectation proceed by splitting the analysis into the error before and after n_0. The individual terms in the definition of $K_2^{IA}(n)$ can be classified based on whether they are bounding the error before or after n_0. In particular, the term labelled (E4) in the definition of $K_2^{IA}(n)$ is a bound on the error before n_0, while the terms collected under (E3) are a bound on the error after n_0.

While the proof of the bound in expectation involves splitting the analysis before and after n_0, the resulting bound via $K_1^{IA}(n)$ does not have a clear split into additive terms that directly correspond to before or after n_0. However, from the proof presented later, it is apparent that C_1 arises out of a bound on the initial error before n_0, the term involving the factor labelled (E1) in the definition of $K_1^{IA}(n)$ arises out of a bound on the sampling error before n_0. Further, C_0 arises out of a bound on the initial error after n_0, and the term labelled (E2) in $K_1^{IA}(n)$ is used to bound the sampling error after n_0.

For a detailed proof, the reader is referred to Sect. 8.4. □

A few remarks are in order.

Remark 10 (Explicit constants) Unlike Fathi and Frikha (2013), where the authors provide concentration bounds for general stochastic approximation schemes, our results provide an explicit n_0, after which the error of iterate averaged batchTD is nearly of the order $O(1/n)$.

Remark 11 (Rate dependence on eigenvalue) From the bounds in Theorem 5.1, it is evident that the dependency on the knowledge of μ for the choice of c can be removed through averaging of the iterates, while obtaining a rate that is close to $1/\sqrt{n}$. In particular, iterate averaging results in a rate that is of the order $O\left(1/n^{(1-\alpha)/2}\right)$, where the exponent α has to be chosen strictly less than 1. Setting $\alpha = 1$ causes the constant C_0 as well as $K_1^{IA}(n), K_2^{IA}(n)$ to blowup and hence, there is a loss of $\alpha/2$ in the rate, when compared to non-averaged batchTD. However, unlike the latter, iterate averaged batchTD does not need the knowledge of μ in setting the step size γ_n.

Remark 12 (Decay rate of initial error) The bound in expectation in Theorem 5.1 can be re-written as follows:

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{C_0 C_1 C_2 \left\|\theta_0 - \hat{\theta}_T\right\|_2}{(n+1)} + \frac{const}{(n+c)^{\alpha/2}}.$$

Thus, the initial error is forgotten at the rate $O(1/n)$, and this is slower than the corresponding rate obtained for the case of non-averaged batchTD (see Remark 2). Hence, as suggested by earlier works on stochastic approximation (cf. Fathi and Frikha 2013), it is preferred to average after a few iterations since the initial error is not forgotten faster than the sampling error with averaging.

Remark 13 (Computational cost vs. accuracy) Let $\epsilon, \delta > 0$. Then, the number of iterations n requires to achieve an accuracy ϵ, i.e., $\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \epsilon$ with probability $1 - \delta$, is of the order $O\left(\frac{1}{\epsilon^{2/\alpha}} \log\left(\frac{1}{\delta}\right)\right)$. On the other hand, the corresponding number of iterations for the non-averaged case (see Theorem 4.2) is $O\left(\frac{1}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right)$.

6 Recent works: a comparison

Non-asymptotic bounds for TD(0) with linear function approximation are derived in three recent works—see Dalal et al. (2018); Bhandari et al. (2018); Lakshminarayanan and Szepesvari (2018). In Dalal et al. (2018); Lakshminarayanan and Szepesvari (2018), the authors consider the i.i.d. sampling case, while the authors by Bhandari et al. (2018) provide bounds in the i.i.d. as well as the more general Markov noise settings. As noted earlier in Remark 7, our analysis could be re-used to derive bounds for TD with linear function approximation in the i.i.d. sampling scenario, while the case of Markov noise is not handled by us. This observation justifies a comparison of the bounds that we derive for batchTD to those in the aforementioned references for TD under i.i.d. sampling, and we provide this comparison below.

In comparison to the references Bhandari et al. (2018) and Lakshminarayanan and Szepesvari (2018), we would like to point out that we derive non-asymptotic bounds that hold with high probability, in addition to bounds that hold in expectation. The aforementioned references provide bounds that hold in expectation only.

The bound in expectation that we derived in Theorem 4.2 matches the bound derived in Bhandari et al. (2018), up to constants. Note that our result in Theorem 4.2, as well as those in (Bhandari et al. 2018) are for the projected variant of TD(0). In addition, we also provide a bound in expectation in Theorem 4.4 for the projection-free variant of TD(0).

Continuing the comparison with Bhandari et al. (2018), the bounds in their work require the knowlege of the minimum eigenvalue μ, which is unknown in a typical RL setting. We get rid of this problematic eigenvalue dependence through iterate averaging, while obtaining a nearly optimal rate of the order $O\big(n^{\alpha/2}\big)$, where $\frac{1}{2} < \alpha < 1$.

The bounds by Dalal et al. (2018) are for TD(0) with linear function approximation under the i.i.d. sampling case, allowing a comparison of bounds for batchTD with their results. The bound in expectation on the error $\left\|\theta_n - \theta^*\right\|_2$ in Theorem 3.1 of Dalal et al. (2018) is $O(\frac{1}{n^\sigma})$, where $0 < \sigma < \frac{1}{2}$. Here θ_n is the TD(0) iterate, and θ^* is the TD fixed point. In contrast, the bound we obtain in Theorem 4.3 is $O(\frac{1}{\sqrt{n}})$. Both results are for the projection-free variant. However, our bound involves a stepsize that requires the knowledge of μ (see (A1)), while their stepsize is $\Theta(\frac{1}{n^{2\sigma}})$. Our results for the iterate-averaged variant in Theorem 5.1 get rid of this stepsize dependence, and the rate we obtain for this variant are comparable to that in Theorem 3.1 of Dalal et al. (2018).

Continuing the comparison with Dalal et al. (2018), we first note that the high-probability bound in 4.2 in our work, which is for the case when μ is known, has a rate of order $O\bigg(\frac{1}{\sqrt{n}}\bigg)$, while the iterate averaged variant in Theorem 5.1 exhibits a rate $O\bigg(\frac{1}{n^{\alpha/2}}\bigg)$, where $0 < \alpha < \frac{1}{2}$. On the other hand, the rate from the bounds in Theorem 3.6 of Dalal et al. (2018), is limited by a problem-dependent parameter λ that is below the minimum eigenvalue (which is μ in our notation). Further, our high probability bound in Theorem 4.2 applies for all n, while that in Theorem 5.1 is for all $n \geq n_0$, with n_0 explicitly specified (as a function of the underlying parameters). In contrast, the bound in Theorem 3.6 of Dalal et al. (2018) applies to sufficiently large n, where the threshold beyond which the bound applies is not explicitly specified. Finally, we project the iterates to keep it bounded, while the bounds by Dalal et al. (2018) do not involve a projection operator. Note that we require projection for the high-probability bounds, while we derive a bound in expectation for the projection-free variant (see Theorem 4.4).

In Lakshminarayanan and Szepesvari (2018), the authors derive non-asymptotic bounds in expectation, which could be applied for TD(0) with linear function approximation, or even to our batchTD algorithm. Lakshminarayanan and Szepesvari (2018) derive lower bounds, while we focus on Theorem 1, which contains the upper bound. Our bound in expectation in Theorem 4.2 is comparable to that in Theorem 1 there, since the overall rate is $O(\frac{1}{\sqrt{n}})$ in either case, and both results assume knowledge about underlying dynamics (through the minimum eigenvalue μ in our case, while through a certain distribution constant for setting the stepsize there). Further, unlike Lakshminarayanan and Szepesvari (2018), we derive bounds for the iterate-averaged variant, which gets rid of the problematic stepsize dependence, at a compromise in the rate, which turns out to be $O(\frac{1}{n^\alpha})$, with $\alpha < \frac{1}{2}$.

7 Fast LSPI using batchTD *(fLSPI)*

LSPI (Lagoudakis and Parr 2003) is a well-known algorithm for control based on the policy iteration procedure for MDPs. We propose a computationally efficient variant of LSPI, which we shall henceforth refer to as fLSPI. The latter algorithm works by substituting the regular LSTDQ with batchTDQ—an algorithm that is quite similar to batchTD described earlier. We first briefly describe the LSPI algorithm and later provide a detailed description of fLSPI.

7.1 Background for LSPI

We are given a set of samples $\mathcal{D} \triangleq \{(s_i, a_i, r_i, s_i'), i = 1, \dots, T)\}$, 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 Sect. 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:

$$Q^\pi(s,a) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t r(s_t, \pi(s_t)) \mid s_0 = s, a_0 = a\right]. \tag{24}$$

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,

$$Q^\pi(s,a) \approx \theta^\mathsf{T} \phi(s,a), \tag{25}$$

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. 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

$$\begin{aligned} &\hat{\theta}_T = \bar{A}_T^{-1} \bar{b}_T, \text{ where} \\ &\bar{A}_T = \frac{1}{T} \sum_{i=1}^{T} \phi(s_i, a_i)(\phi(s_i, a_i) - \beta \phi(s_i', \pi(s_i')))^\mathsf{T}, \text{ and } \bar{b}_T = \frac{1}{T} \sum_{i=1}^{T} r_i \phi(s_i, a_i). \end{aligned} \tag{26}$$

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:

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} \theta^\mathsf{T} \phi(s,a).$$

Since this policy is provably better than π, iterating this procedure allows LSPI to find an approximately optimal policy.

7.2 fLSPI algorithm

The fLSPI algorithm works by substituting the regular LSTDQ with its computationally efficient variant batchTDQ. The overall structure of fLSPI is given in Algorithm 2.

For a given policy π, batchTDQ approximates LSTDQ solution (26) by an iterative update scheme as follows (starting with an arbitrary θ_0):

$$\theta_k = \theta_{k-1} + \gamma_k\Big(r_{i_k} + \beta\theta_{k-1}^{\mathsf{T}}\phi(s'_{i_k}, \pi(s'_{i_k})) - \theta_{k-1}^{\mathsf{T}}\phi(s_{i_k}, a_{i_k})\Big)\phi(s_{i_k}, a_{i_k}) \tag{27}$$

From Sect. 2.2, it is evident that the claims in Proposition 8.1 and Theorem 4.2 hold for the above scheme as well.

Algorithm 2 fLSPI

```
Input: Sample set $D \triangleq \{s_i, a_i, r_i, s'_i\}_{i=1}^T$, obtained from an initial (arbitrary) policy.
Initialization: $\epsilon$, $\tau$, step-sizes $\{\gamma_k\}_{k=1}^T$, initial policy $\pi_0$ (given as $\theta_0$).
$\pi \leftarrow \pi_0, \theta \leftarrow \theta_0$.
repeat
    Policy Evaluation
        Approximate LSTDQ$(D, \pi)$ using batchTDQ$(D, \pi)$ as follows:
        for $k = 1 \ldots \tau$ do
            Get random sample index: $i_k \sim U(\{1, \ldots, T\})$.
            Update batchTDQ iterate $\theta_k$ using (27).
        end for
    $\theta' \leftarrow \theta_\tau, \Delta = \|\theta - \theta'\|_2$.
    Policy Improvement
        Obtain a greedy policy $\pi'$ as follows: $\pi'(s) = \arg\max_{a \in \mathcal{A}} \theta'^{\mathsf{T}} \phi(s, a)$.
    $\theta \leftarrow \theta', \pi \leftarrow \pi'$.
until $\Delta < \epsilon$
```

Remark 14 Error bounds for fLSPI can be derived along the lines of those for regular on-policy LSPI by Lazaric et al. (2012), and we omit the details.

8 Convergence proofs

Let $\mathcal{F}_n$ denotes the σ-field generated by $\theta_0, \ldots, \theta_n$, $n \geq 0$. Let

$$f_n(\theta) \triangleq \Big(r_{i_n} + \beta\theta^{\mathsf{T}}\phi(s'_{i_n}) - \theta^{\mathsf{T}}\phi(s_{i_n})\Big)\phi(s_{i_n}). \tag{28}$$

Recall that we denote the current state feature $(T \times d)$-matrix by $\boldsymbol{\Phi} \triangleq (\phi(s_1)^{\mathsf{T}}, \ldots, \phi(s_T))$, the next state feature $(T \times d)$-matrix by $\boldsymbol{\Phi'} \triangleq (\phi(s'_1)^{\mathsf{T}}, \ldots, \phi(s'_T))$, and the reward $(T \times 1)$-vector by $\mathcal{R} = (r_1, \ldots, r_T)^{\mathsf{T}}$. Recall also that the LSTD solution is given by

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

Finally we note also that the pathwise LSTD solution has the same form as above, except that $\boldsymbol{\Phi'} \triangleq \hat{P}\boldsymbol{\Phi} = (\phi(s'_1)^{\mathsf{T}}, \ldots, \phi(s'_{T-1})^{\mathsf{T}}, \mathbf{0}^{\mathsf{T}})$, where $\mathbf{0}$ is the $d \times 1$-zero-vector.

8.1 Proof of asymptotic convergence

Proof of Theorem 4.3 (batchTD without projection):

Proof We first rewrite (10) as follows:

$$\theta_n = \theta_{n-1} + \gamma_n\big(-\bar{A}_T\theta_{n-1} + \bar{b}_T + \Delta M_n\big), \tag{29}$$

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 sequence, with $f_n(\cdot)$ as defined in (28).

The ODE associated with (29) is

$$\dot{\theta}(t) = q(\theta(t)), t \geq 0. \tag{30}$$

In the above, $q(\theta(t)) \triangleq -\bar{A}_T\theta(t) + \bar{b}_T$.

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 Borkar and Meyn (2000), provided we verify assumptions (A1)–(A2) there. These assumptions are as follows:

(a1) The function q is Lipschitz. For any $\eta \in \mathbb{R}$, 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 a globally asymptotically stable equilibrium for the ODE

$$\dot{\theta}(t) = -q_\infty(\theta(t)). \tag{31}$$

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

$$\mathbb{E}[\|\Delta M_{n+1}\|_2^2 \mid \mathcal{F}_n] \leq C_0(1 + \|\theta_n\|_2^2),\ n \geq 0,$$

for some $C_0 < \infty$.

We now verify (a1) and (a2) in our context. Notice that $q_\eta(\theta) \triangleq -\bar{A}_T\theta + \bar{b}_T/\eta$ 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 (A1), the aforementioned ODE has the origin as its globally asymptotically stable equilibrium. This verifies (a1).

For verifying (a2), notice that

$$\begin{aligned}\mathbb{E}[\|\Delta M_{n+1}\|_2^2 \mid \mathcal{F}_n] \leq & \mathbb{E}[\|f_{n+1}(\theta_2)\|_2^2 \mid \mathcal{F}_n] \\ \leq & (R_{\max}\Phi_{\max} + (1+\beta)\Phi_{\max}^2\|\theta_n\|_2)^2\end{aligned}$$

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

Proof of Theorem 4.1 (batchTD with projection):

Proof We first rewrite (10) as follows:

$$\theta_n = \Upsilon\big(\theta_{n-1} + \gamma_n\big(-\bar{A}_T\theta_{n-1} + \bar{b}_T + \Delta M_n\big)\big), \tag{32}$$

where ΔM_n, $\mathcal{F}_n$ and $f_n(\theta)$ are as defined in (28).

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 $\varDelta M_n$ is upper bounded by $2\big(R_{\max}\varPhi_{\max} + (1+\beta)H\varPhi^2_{\max}\big)$. Thus, (32) can be seen as a discretization of the ODE

$$\dot{\theta}(t) = \check{\varUpsilon}(-\bar{A}_T\theta(t) + \bar{b}_T),\ t \geq 0, \tag{33}$$

where $\check{\varUpsilon}(\theta) = \lim_{\tau\to 0}\left[(\varUpsilon(\theta + \tau f(\theta)) - \theta)/\tau\right]$, for any bounded continuous f. The operator $\check{\varUpsilon}$ ensures that θ governed by (33) evolves within the set $\mathcal{C}$ that contains $\hat{\theta}_T$. As in the proof of Lemma 4.1 by Yu (2015), we have

$$0 = \langle \hat{\theta}_T, -\bar{A}_T\hat{\theta}_T + \bar{b}_T\rangle \leq -\mu\left\|\hat{\theta}_T\right\|_2^2 + \left\|\bar{b}_T\right\|_2\left\|\hat{\theta}_T\right\|_2,$$

where the inequality follows from (A1). From the foregoing, we have that $\left\|\hat{\theta}_T\right\|_2 \leq \frac{\|\bar{b}_T\|_2}{\mu} < H \Rightarrow \hat{\theta}_T \in \mathcal{C}$. Following similar arguments as before, it can be inferred that at any boundary point θ of $\mathcal{C}$, $\langle\theta, -\bar{A}_t\theta + \bar{b}_T\rangle < 0$, and hence the ODE (33) has the origin as its globally asymptotically stable equilibrium. The claim now follows from Theorem 2 in Chapter 2 of Borkar (2008) (or even Theorem 5.3.1 on pp. 191–196 of Kushner and Clark (1978)). □

8.2 Proofs of finite-time error bounds for batchTD

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 (34) below), and the size of its mean itself (see (35) 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).

Proposition 8.1 below gives these results for general step sequences. The proof involves two martingale analyses, which also form the template for the proofs for the least squares regression extension (see Sect. 10), and the iterate averaged variant of batchTD (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 8.1 *Let $z_n = \theta_n - \hat{\theta}_T$, where θ_n is given by* (10). *Under (A1)–(A4), we have* $\forall\epsilon > 0$,

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

$$\mathbb{P}\big(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon\big) \leq \exp\left(-\frac{\epsilon^2}{4\big(R_{\max} + (1+\beta)H\varPhi^2_{\max}\big)^2 \sum\limits_{k=1}^{n} L_k^2}\right), \tag{34}$$

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

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

$$
\begin{aligned}
\mathbb{E}\big(\|z_n\|_2\big)^2 \le{} & \underbrace{\left[\prod_{k=1}^{n}\big(1-\gamma_k(2\mu-\gamma_k(1+\beta)^2\Phi^4_{\max})\big)\|z_0\|_2\right]^2}_{\textbf{initial error}} \\
& + \underbrace{4\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\Phi^4_{\max})\right]^2}_{\textbf{sampling error}}\big(R_{\max}+(1+\beta)H\Phi^2_{\max}\big)^2.
\end{aligned}
\tag{35}
$$

As mentioned earlier, the initial error relates to the starting point θ_0 of the algorithm, while the sampling error arises out of a martingale difference sequence (see Step 1 in Sect. 8.2.2 below for a precise definition).

We establish later, in Sect. 8.2.3, that under a suitable choice of step sizes, the initial error is forgotten faster than the sampling error.

We claim that the terms of the form $1-\gamma_j(2\mu-\gamma_j\Phi^4_{\max}(1+\beta)^2)$, which go into a product in the Lipschitz constant L_i as well as in the initial/sampling error terms of the expectation bound, are positive. This claim can be seen as follows:

$$
\begin{aligned}
1-\gamma_j\big(2\mu-\gamma_j\Phi^4_{\max}(1+\beta)^2\big) &\ge 1-2\gamma_j(1+\beta)\Phi^2_{\max}+\gamma_j^2\Phi^4_{\max}(1+\beta)^2 \\
&= \big(1-\gamma_j(1+\beta)\Phi^2_{\max}\big)^2 \ge 0,
\end{aligned}
\tag{36}
$$

where the inequality above follows from the fact that $\mu \le (1+\beta)\Phi^2_{\max}$.

In Sect. 8.2.3, to establish the rates of Theorem 4.2, we first prove that $\sum_{i=1}^{n} L_i$ is an order $1/n$ term, and the claim of positivity of L_i is necessary for the aforementioned proof.

8.2.1 Proof of Proposition 8.1 part (1)

Proof 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 $z_n \triangleq \theta_n - \hat{\theta}_T$. We rewrite $\|z_n\|_2 - \mathbb{E}\|z_n\|_2$ as follows:

$$
\|z_n\|_2 - \mathbb{E}\|z_n\|_2 = \sum_{k=1}^{n}\big(g_k - g_{k-1}\big) = \sum_{k=1}^{n} D_k,
\tag{37}
$$

where $g_k \triangleq \mathbb{E}[\|z_n\|_2 | \mathcal{F}_k]$, $D_k \triangleq g_k - \mathbb{E}[g_k|\mathcal{F}_{k-1}]$, and $\mathcal{F}_k$ denotes the σ-field generated by the random variables $\{\theta_i, i \le k\}$ for $k \ge 0$.

Recall that $f_k(\theta) \triangleq (\theta^{\top}\phi(s_{i_k}) - (r_{i_k} + \beta\theta^{\top}\phi(s'_{i_k})))\phi(s_{i_k})$ denotes the random innovation at time k, given that $\theta_{k-1} = \theta$.

Step 2 (Showing that g_k is a Lipschitz function of the random innovation f_k)[4]

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

[4] For notational convenience, we have not chosen to make the dependence of g_k on the random innovation f_k explicit. The Lipschitzness of g_k as a function of f_k is clear from equation (43) presented below.

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 Frikha and Menozzi (2012), 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:

Let $\Theta^k_j(\theta)$ 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 non-expansive,

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

Expanding the random innovation terms, we have

$$\begin{aligned}
&\Theta^k_{j-1}(\theta)-\Theta^k_{j-1}(\theta')-\gamma_j[f_j(\Theta^k_{j-1}(\theta))-f_j(\Theta^k_{j-1}(\theta'))]\\
&\quad=\Theta^k_{j-1}(\theta)-\Theta^k_{j-1}(\theta')-\gamma_j[\phi(s_{i_j})\phi(s_{i_j})^\top-\beta\phi(s_{i_j})\phi(s'_{i_j})^\top](\Theta^k_{j-1}(\theta)-\Theta^k_{j-1}(\theta'))\\
&\quad=[I-\gamma_j a_j](\Theta^k_{j-1}(\theta)-\Theta^k_{j-1}(\theta')),
\end{aligned}\tag{38}$$

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

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

Recall that $\boldsymbol{\Phi}^\top \triangleq (\phi(s_1),\dots,\phi(s_T))$, and $\boldsymbol{\Phi}'^\top \triangleq (\phi(s_1)',\dots,\phi(s_T)')$. Let $\Delta \triangleq \operatorname{diag}(\|\phi(s_1)\|_2^2,\dots,\|\phi(s_T)\|_2^2)$. Then, for any vector θ, we have

$$\begin{aligned}
&\mathbb{E}\left(\theta^\top(I-\gamma_j a_j)^\top(I-\gamma_j a_j)\theta \mid \mathcal{F}_{j-1}\right)\\
&\quad=\theta^\top\mathbb{E}(I-\gamma_j[a_j^\top+a_j-\gamma_j a_j^\top a_j])\theta \mid \mathcal{F}_{j-1})\\
&\quad=\|\theta\|_2^2-\gamma_j\theta^\top\frac{1}{T}\Big[2\boldsymbol{\Phi}^\top\boldsymbol{\Phi}-\beta\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}'+\boldsymbol{\Phi}'^\top\boldsymbol{\Phi}\right)\\
&\qquad-\gamma_j\left(\boldsymbol{\Phi}^\top\Delta\boldsymbol{\Phi}-\beta\left(\boldsymbol{\Phi}'^\top\Delta\boldsymbol{\Phi}+\boldsymbol{\Phi}^\top\Delta\boldsymbol{\Phi}'\right)+\beta^2\boldsymbol{\Phi}'^\top\Delta\boldsymbol{\Phi}'\right)\Big]\theta
\end{aligned}\tag{39}$$

$$\le\|\theta\|_2^2-\gamma_j 2\mu\|\theta\|_2^2+\gamma_j^2\theta^\top\frac{1}{T}\left(\boldsymbol{\Phi}^\top\Delta\boldsymbol{\Phi}-\beta\left(\boldsymbol{\Phi}'^\top\Delta\boldsymbol{\Phi}+\boldsymbol{\Phi}^\top\Delta\boldsymbol{\Phi}'\right)\right)\theta+\beta^2\|\theta\|_2^2\boldsymbol{\Phi}^4_{\max}\tag{40}$$

$$\le(1-\gamma_j(2\mu-\gamma_j\boldsymbol{\Phi}^4_{\max}(1+\beta)^2))\|\theta\|_2^2.\tag{41}$$

For the equality in (39), we have used that $\sum_{k=1}^T\phi(s_k)\phi(s_k)^\top=\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$, and similar identities. Further, the inequality in (40) can be inferred using the following fact:

$$\lambda_{\min}\Big(2\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} - \beta\Big(\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}'\Big)\Big) = \lambda_{\min}\Big((\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} - \beta\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Phi}) + (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} - \beta\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Phi})^{\mathsf{T}}\Big)$$
$$= \lambda_{\min}\big(T\big(\bar{A}_T + \bar{A}_T^{\mathsf{T}}\big)\big) \geq 2T\mu,$$

where we have used assumption (A1) for the last inequality above. The last term in (40) follows from $|\theta^{\mathsf{T}}\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}'\theta| \leq \|\theta\|_2^2\boldsymbol{\Phi}_{\max}^4$, where we have used assumption (A2) that ensures features are bounded. The inequality in (41) can be inferred as follows:

$$\begin{aligned}
&|\theta\Big(\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}'\Big)\theta| \leq 2\|\theta\|_2^2\boldsymbol{\Phi}_{\max}^4\\
&\quad\Rightarrow -2\|\theta\|_2^2\boldsymbol{\Phi}_{\max}^4 \leq \theta^{\mathsf{T}}\Big(\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}'\Big)\theta\\
&\quad\Rightarrow\ \theta^{\mathsf{T}}(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi} - \beta\Big(\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}'\Big) + \beta^2\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}')\theta\\
&\quad\leq \|\theta\|_2^2(1+2\beta+\beta^2)\boldsymbol{\Phi}_{\max}^4 = (1+\beta)^2\boldsymbol{\Phi}_{\max}^4\|\theta\|_2^2.
\end{aligned}$$

In the above, we have used the boundedness of features to infer $|\theta^{\mathsf{T}}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}\theta| \leq \|\theta\|_2^2\boldsymbol{\Phi}_{\max}^4$, and $|\theta^{\mathsf{T}}\boldsymbol{\Phi}'^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{\Phi}'\theta| \leq \|\theta\|_2^2\boldsymbol{\Phi}_{\max}^4$.

Hence, from the tower property of conditional expectations, it follows that:

$$\begin{aligned}
\mathbb{E}\Big[\Big\|\Theta_n^k(\theta) - \Theta_n^k(\theta')\Big\|_2^2\Big] &= \mathbb{E}\Big[\mathbb{E}\Big(\Big\|\Theta_n^k(\theta) - \Theta_n^k(\theta')\Big\|_2^2 \mid \mathcal{F}_{n-1}\Big)\Big]\\
&\leq \big(1-\gamma_n\big(2\mu - \gamma_n\boldsymbol{\Phi}_{\max}^4(1+\beta)^2\big)\big)\mathbb{E}\Big[\Big\|\Theta_{n-1}^k(\theta) - \Theta_{n-1}^k(\theta')\Big\|_2^2\Big]\\
&\leq \Bigg[\prod_{j=k+1}^{n}\big(1-\gamma_j\big(2\mu - \gamma_j\boldsymbol{\Phi}_{\max}^4(1+\beta)^2\big)\big)\Bigg]\|\theta-\theta'\|_2^2
\end{aligned} \tag{42}$$

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

$$\begin{aligned}
&\Big|\mathbb{E}\Big[\Big\|\theta_n - \hat{\theta}_T\Big\|_2 \big|\theta_k = \theta\Big] - \mathbb{E}\Big[\Big\|\theta_n - \hat{\theta}_T\Big\|_2 \big|\theta_k = \theta'\Big]\Big|\\
&\quad\leq \mathbb{E}\Big[\Big\|\Theta_n^k(\theta) - \Theta_n^k\big(\theta'\big)\Big\|_2\Big] \leq L_k\|f - f'\|_2,
\end{aligned} \tag{43}$$

which proves that the functions g_k are L_k-Lipschitz in the random innovations at time k. Recall that $D_k = g_k - g_{k-1}$, and hence, the Lipschitz constant of D_k is $\max\big(L_k, L_{k-1}\big)$. However, from (36), we have $L_k > L_{k-1}$, leading to a Lipschitz constant of L_k for D_k.

Step 3 (Applying a sub-Gaussian concentration inequality)

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

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

The last equality above follows from (37), while the first inequality follows from Markov's inequality.

Let Z be a zero-mean random variable (r.v) satisfying $|Z| \leq B$ w.p. 1, and g be a L-Lipschitz function g. Letting Z' denote an independent copy of Z and ε a Rademacher r.v., we have

$$\begin{aligned}\mathbb{E}(\exp(\lambda g(Z))) &= \mathbb{E}\Big(\exp\big(\lambda\big(g(Z) - \mathbb{E}(g(Z'))\big)\big)\Big)\\ &\leq \mathbb{E}\Big(\exp\big(\lambda\big(g(Z) - g(Z')\big)\big)\Big)\end{aligned} \tag{44}$$

$$= \mathbb{E}\Big(\exp\big(\lambda\varepsilon\big(g(Z) - g(Z')\big)\big)\Big) \tag{45}$$

$$\leq \mathbb{E}\Big(\exp\Big(\lambda^2\big(g(Z) - g(Z')\big)^2/2\Big)\Big) \tag{46}$$

$$\leq \mathbb{E}\Big(\exp\Big(\lambda^2 L^2\big(Z - Z'\big)^2/2\Big)\Big) \tag{47}$$

$$\leq \exp\big(\lambda^2 B^2 L^2/2\big). \tag{48}$$

In the above, we have used Jensen's inequality in (44), the fact that distribution of $g(Z) - g(Z')$ is the same as $\varepsilon(g(Z) - g(Z'))$ in (45), a result from Example 2.2 in Wainwright (2019) in (46), the fact that g is L-Lipschitz in (47), and the boundedness of Z in (48).

Note that by (A3), and the projection step of the algorithm, we have that $|f_k(\theta_{k-1})| < (R_{\max} + (1+\beta)H\Phi^2_{\max})$ 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. Hence, we obtain

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

leading to

$$\mathbb{P}(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon) \leq \exp(-\lambda\epsilon)\exp\left(\frac{\lambda^2\big(R_{\max} + (1+\beta)H\Phi^2_{\max}\big)^2}{2}\sum_{k=1}^{n} L_k^2\right). \tag{49}$$

The proof of Proposition 8.1 part (1) follows by optimizing over λ in (49). □

8.2.2 Proof of Proposition 8.1 part (2)

Proof 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 $\bar{A}_T = \mathbb{E}((\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^\top \mid \mathcal{F}_{n-1})$ 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's inequality to the square of the norm)

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

$$\begin{aligned}&\mathbb{E}\left(\|z_n\|_2 \mid \mathcal{F}_{n-1}\right)^2 \leq \mathbb{E}(\langle z_n, z_n\rangle \mid \mathcal{F}_{n-1})\\ &\quad\leq \mathbb{E}(\langle \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1}), \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1})\rangle \mid \mathcal{F}_{n-1})\\ &\quad= \mathbb{E}(\langle \left(I - \gamma_n \bar{A}_T\right) z_{n-1} - \gamma_n \Delta M_n, \left(I - \gamma_n \bar{A}_T\right) z_{n-1} - \gamma_n \Delta M_n\rangle \mid \mathcal{F}_{n-1})\\ &\quad= z_{n-1}^\top \left(I - \gamma_n \bar{A}_T\right)^\top \left(I - \gamma_n \bar{A}_T\right) z_{n-1} + \gamma_n^2 \mathbb{E}\left(\langle \Delta M_n, \Delta M_n\rangle \mid \mathcal{F}_{n-1}\right)\\ &\quad\leq \|z_{n-1}\|_2^2 \left\|\left(I - \gamma_n \bar{A}_T\right)^\top \left(I - \gamma_n \bar{A}_T\right)\right\|_2 + \gamma_n^2 \mathbb{E}\Big(\|\Delta M_n\|_2^2 \mid \mathcal{F}_{n-1}\Big).\end{aligned} \tag{50}$$

Note that the cross-terms have vanished in (50) since ΔM_n is martingale difference, independent of the other terms, given $\mathcal{F}_{n-1}$.

Step 3 (Unroll the iteration)

Using assumptions (A1) and (A2)

$$\|(I - \gamma_n \bar{A}_T)^\top (I - \gamma_n \bar{A}_T)\|_2 = \|(I - \gamma_n((\bar{A}_T^\top + \bar{A}_T) - \gamma_n \bar{A}_T^\top \bar{A}_T)\|_2 \quad \leq 1 - \gamma_n(2\mu - \gamma_n(1+\beta)^2 \Phi_{\max}^4) \tag{51}$$

Furthermore, by assumption (A3), and the projection step, the martingale differences ΔM_n are bounded in norm by $2(R_{\max} + (1+\beta)H\Phi_{\max}^2)$. By applying the tower property of conditional expectations repeatedly together with (51) we arrive at the following bound:

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

□

8.2.3 Derivation of rates given in Theorem 4.2

Proof To obtain the rates specified in the bound in expectation in Theorem 4.2, we simplify the bound in expectation in Proposition 8.1 using the choice $\gamma_n = \frac{c_0 c}{(c+n)}$, with $c_0 \in (0, \mu((1+\beta)^2\Phi_{\max}^4)^{-1}]$ and $2c_0 c\mu \in (1, \infty)$. Consider the sampling error term in (35) under the aforementioned choice for the step size.

$$\begin{aligned}&\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k+1}^{n}(1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\boldsymbol{\Phi}_{\max}^4)\right]^2\\&\quad=\sum_{k=1}^{n}\gamma_k^2\exp\left(2\sum_{j=k+1}^{n}\ln\left(1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\boldsymbol{\Phi}_{\max}^4)\right)\right)\end{aligned}\tag{52}$$

$$=\sum_{k=1}^{n}\frac{c_0^2c^2}{(c+k)^2}\exp\left(2\sum_{j=k+1}^{n}\ln\left(1-\frac{c_0c}{c+j}\left(2\mu-\frac{c_0c}{c+j}(1+\beta)^2\boldsymbol{\Phi}_{\max}^4\right)\right)\right)$$

$$\leq\sum_{k=1}^{n}\frac{c_0^2c^2}{(c+k)^2}\exp\left(2\sum_{j=k+1}^{n}\ln\left(1-\frac{c_0c\mu}{c+j}\right)\right)\tag{53}$$

$$\leq\sum_{k=1}^{n}\frac{c_0^2c^2}{(c+k)^2}\exp\left(-2c_0c\mu\left(\sum_{j=k+1}^{n}\frac{1}{c+j}\right)\right)\tag{54}$$

$$\leq c_0^2c^2(c+n+1)^{-2c_0c\mu}\sum_{k=1}^{n}(c+k+1)^{2c_0c\mu}(c+k)^{-2}\tag{55}$$

$$\leq\frac{c_0^2c^2e^2}{(2c_0c\mu-1)(n+c+1)}\tag{56}$$

In the above, the inequality in (52) uses the fact that $1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\boldsymbol{\Phi}_{\max}^4>0$, a claim that was established earlier in (36). The inequality in (53) uses $c_0\in(0,\mu((1+\beta)^2\boldsymbol{\Phi}_{\max}^4)^{-1}]$. The inequality in (54) follows by using $\ln(1+u)\leq u$. To infer the inequality in (55), we use $\sum_{j=k+1}^{n}(c+j)^{-1}\geq\int_{x=c+k+1}^{n+c+1}x^{-1}dx$, which holds because the LHS is the upper Riemann sum of RHS. Now, evaluating the integral of x^{-1}, the exponential term inside the summand of (54) becomes:

$$\begin{aligned}\exp\left(-2c_0c\mu\sum_{j=k+1}^{n}(c+j)^{-1}\right)&\leq\exp\left(-2c_0c\mu[\ln(c+n+1)-\ln(c+k+1)]\right)\\&=(c+n+1)^{-2c_0c\mu}(c+k+1)^{2c_0c\mu},\end{aligned}$$

and the inequality in (55) follows by substituting the bound on the RHS above. We obtain the final inequality, (56), by upper bounding the term $\sum_{k=1}^{n}(k+c+1)^{2c_0c\mu}(k+c)^{-2}$ on the RHS of (55) as follows:

$$\begin{aligned}\sum_{k=1}^{n}(k+c+1)^{2c_0c\mu}(k+c)^{-2}&=\sum_{k=1}^{n}(((k+c)(1+1/(k+c)))^{2c_0c\mu}(k+c)^{-2}\\&\leq\sum_{k=1}^{n}(1+1/c)^{2c}(k+c)^{2c_0c\mu}(k+c)^{-2}\end{aligned}\tag{57}$$

$$\leq e^2 \sum_{k=1}^{n} (k+c)^{2(c_0 c\mu - 1)} \tag{58}$$

$$\begin{aligned} &\leq e^2 \int_{x=0}^{n+1} (x+c)^{2(c_0 c\mu-1)} dx \\ &= \frac{e^2 (n+c+1)^{-(1-2c_0 c\mu)}}{(2c_0 c\mu - 1)}, \end{aligned} \tag{59}$$

where the inequality in (58) holds because

$$c_0\mu \leq \frac{\mu^2}{\boldsymbol{\Phi}_{\max}^4 (1+\beta)^2} \leq \left(\frac{\mu}{\boldsymbol{\Phi}_{\max}^2}\right)^2 \leq 1.$$

Further, the inequality in (58) follows from the fact that $(1+1/c)^{2c} \leq e^2$ for all $c > 0$, and the inequality in (59) follows by comparison of a sum with an integral together with the assumption that $c_0 c\mu > 1$.

Similarly, the initial error term in (35) can be simplified from the hypothesis that $c_0 c\mu \in (1, \infty)$ and $c_0 \in (0, \mu((1+\beta)^2 \boldsymbol{\Phi}_{\max}^4)^{-1}]$ as follows:

$$\begin{aligned} &\prod_{k=1}^{n} (1 - \gamma_k (2\mu - \gamma_k (1+\beta)^2 \boldsymbol{\Phi}_{\max}^4) \\ &\quad \leq \exp\left(-c_0 c\mu \sum_{j=1}^{n} \frac{1}{c+j}\right) \leq \left(\frac{c+1}{n+c}\right)^{c_0 c\mu} \end{aligned} \tag{60}$$

The last inequality above follows again from a comparison with an integral: $\sum_{j=1}^{n} \frac{1}{c+j} \geq \int_{c+1}^{c+n} x^{-1} dx = \ln \frac{n+c}{c+1}$. Hence, we obtain

$$\begin{aligned} \mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq &\left(\frac{\left\|\theta_0 - \hat{\theta}_T\right\|_2 \sqrt{(c+1)^{c_0 c\mu}}}{\sqrt{(n+c)^{c_0 c\mu - 1}}} + \frac{2ec_0 c(R_{\max} + (1+\beta) H \boldsymbol{\Phi}_{\max}^2)}{\sqrt{2c_0 c\mu - 1}} \right) \\ &\times \sqrt{\frac{1}{n+c}}, \end{aligned} \tag{61}$$

and the result concerning the bound in expectation in Theorem 4.2 now follows.

We now derive the rates for the high-probability bound in Theorem 4.2. With $\gamma_n = \frac{c_0 c}{(c+n)}$, and $c_0 \in (0, \mu((1+\beta)^2 \boldsymbol{\Phi}_{\max}^4)^{-1}]$, we have

$$\begin{aligned} \sum_{i=1}^{n} L_i^2 &= \sum_{i=1}^{n} \frac{c_0^2 c^2}{(c+i)^2} \prod_{j=i+1}^{n} \left(1 - \frac{c_0 c}{(c+j)} \left(2\mu - (1+\beta)^2 \boldsymbol{\Phi}_{\max}^4 \frac{c_0 c}{(c+j)}\right)\right) \\ &\leq \sum_{i=1+1}^{n} \frac{c_0^2 c^2}{(c+i)^2} \prod_{j=i+1}^{n} \left(1 - \frac{c_0 c\mu}{(c+j)}\right) \end{aligned} \tag{62}$$

$$\leq \sum_{i=1}^{n} \frac{c_0^2 c^2}{(c+i)^2} \exp\left(-c_0 c\mu \sum_{j=i+1}^{n} \frac{1}{(c+j)}\right) \tag{63}$$

$$\leq \frac{c_0^2 c^2}{(n+c)^{c_0 c\mu}} \sum_{i=1}^{n} (i+c+1)^{c_0 c\mu} (i+c)^{-2}. \tag{64}$$

$$\leq \frac{c_0^2 c^2 e}{(n+c)^{c_0 c\mu}} \sum_{i=1}^{n} (i+c)^{-(2-c_0 c\mu)}. \tag{65}$$

Inequality (62) follows from the assumption on c_0. To obtain the inequality (63), as in the rates for the bound in expectation, we have taken the exponential of the logarithm of the product, brought the product outside the logarithm as a sum, and applied the inequality $\ln(1-x) \leq x$ which holds for $x \in [0, 1)$. The inequality in (64) can be inferred in a manner analogous to that in (55), while that in (65) follows in a similar manner as (58).

We now find three regimes for the rate of convergence, based on the choice of c. Each case is again derived from a comparison of the sum in (65) with an appropriate integral:

(i) $\sum_{i=1}^{n} L_i^2 = O((n+c)^{c_0 c\mu})$ 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) $\sum_{i=1}^{n} L_i^2 \leq \frac{c_0^2 c^2 e}{(c_0 c\mu - 1)} (n+c)^{-1}$ when $c_0 c\mu \in (1, \infty)$.

Thus, setting $c \in (1/(c_0\mu), \infty)$, the high probability bound from Proposition 8.1 gives

$$\mathbb{P}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2 - \mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2 (n+c)}{4K_{\mu,c,c_0,\beta}}\right), \tag{66}$$

where $K_{\mu,c,c_0,\beta} \triangleq \dfrac{c_0^2 c^2 e\left(R_{\max} + (1+\beta) H \Phi_{\max}^2\right)^2}{(c_0 c\mu - 1)}$. The high probability bound in Theorem 4.2 now follows. □

8.3 Proof of expectation bound for batchTD 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 8.1 part 2. The proof of the following proposition differs from that of Proposition 8.1 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 8.2 *Let $z_n = \theta_n - \hat{\theta}_T$, where θ_n is given by* (10) with $\Upsilon(\theta) = \theta$, $\forall \theta \in \mathbb{R}^d$. *Under (A1)–(A4), we have* $\forall \epsilon > 0$,

$$\begin{aligned}\mathbb{E}\left(\|z_n\|_2\right)^2 \le{}& 3\underbrace{\left[\prod_{k=1}^{n}\left(1-\gamma_k(2\mu-\gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right)\|z_0\|_2\right]^2}_{\textbf{initial error}}\\ &+\underbrace{3\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2\left(R_{\max}+(1+\beta)\left\|\hat{\theta}_T\right\|_2\boldsymbol{\Phi}^2_{\max}\right)^2}_{\textbf{sampling error}}\end{aligned} \tag{67}$$

Proof The proof involves two steps.

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:

$$\begin{aligned}z_n &= \theta_n - \hat{\theta}_T = \theta_{n-1} - \hat{\theta}_T - \gamma_n f_n(\theta_{n-1})\\ &= \left(I - \gamma_n(\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^{\mathsf{T}}\right)z_{n-1} - \gamma_n f_n(\hat{\theta}_T)\\ &= \Pi_1^n z_0 - \sum_{k=1}^{n}\gamma_k \Pi_{k+1}^n f_k(\hat{\theta}_T).\end{aligned}$$

where $\Pi_k^n \triangleq \prod_{j=k}^{n}\left(I - \gamma_j(\phi(s_{i_j}) - \beta\phi(s'_{i_j}))\phi(s_{i_j})^{\mathsf{T}}\right)$ for $1 \le k \le n$, and $\Pi_k^n = I$ for $k > n$.[5] In the above, we have used that the random increment at time n has the form $f_n(\theta) = (\theta^{\mathsf{T}}\phi(s_{i_n}) - (r_{i_n} + \beta\theta^{\mathsf{T}}\phi(s'_{i_n})))\phi(s_{i_n})$. 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

$$\mathbb{E}\left(\|z_n\|_2\right)^2 \le 3z_0^{\mathsf{T}}\mathbb{E}\left(\Pi_1^{n\,\mathsf{T}}\Pi_1^n\right)z_0 + 3\sum_{k=1}^{n}\gamma_k^2\mathbb{E}\left(f_k(\hat{\theta}_T)^{\mathsf{T}}\Pi_{k+1}^{n}{}^{\mathsf{T}}\Pi_{k+1}^n f_k(\hat{\theta}_T)\right), \tag{68}$$

where we have used the identity $\|x-y\|_2^2 \le 3\|x\|_2^2 + 3\|y\|_2^2$ for any two vectors x, y.

Using assumptions (A1) and (A2), we have

$$\begin{aligned}&\left\|\mathbb{E}(\left(I - \gamma_n(\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^{\mathsf{T}}\right)^{\mathsf{T}}\left(I - \gamma_n(\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n}))^{\mathsf{T}}\right)\right\|_2\\ &= \Bigg\|\mathbb{E}\Bigg(I - \gamma_n((\phi(s_{i_n}) - \beta\phi(s'_{i_n}))\phi(s_{i_n})^{\mathsf{T}} - \gamma_n\phi(s_{i_n})(\phi(s_{i_n}) - \beta\phi(s'_{i_n}))^{\mathsf{T}}\\ &\quad + \gamma_n^2\left(\left\|\phi(s_{i_n})\right\|_2^2 - 2\beta\langle\phi(s'_{i_n}), \phi(s_{i_n})\rangle + \beta^2\left\|\phi(s'_{i_n})\right\|_2^2\right)\phi(s_{i_n}))\phi(s_{i_n}))^{\mathsf{T}}\Bigg)\Bigg\|_2\\ &\le 1 - \gamma_n(2\mu - \gamma_n(1+\beta)^2\boldsymbol{\Phi}^4_{\max}).\end{aligned} \tag{69}$$

[5] One usually sees terms of the form $\phi(s_{i_j})(\phi(s_{i_j}) - \beta\phi(s'_{i_j}))$, whereas we use a transposed form to simplify handling the products that get written through the Π_j^n matrices.

Furthermore, by assumption (A3), the random variables $f_n(\hat{\theta}_T)$ are bounded in norm by $(R_{\max} + (1+\beta)\left\|\hat{\theta}_T\right\|_2 \Phi^2_{\max})$. So, by applying the tower property of conditional expectations repeatedly together with (69) we arrive at the following bound:

$$\mathbb{E}(\|z_n\|_2) \leq \left(3\left[\prod_{k=1}^{n}(1-\gamma_k(2\mu-\gamma_k(1+\beta)^2\Phi^4_{\max})\|z_0\|_2\right]^2 \right.$$
$$\left. +3\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1-\gamma_j(2\mu-\gamma_j(1+\beta)^2\Phi^4_{\max})\right]^2\left(R_{\max}+(1+\beta)\left\|\hat{\theta}_T\right\|_2\Phi^2_{\max}\right)^2\right)^{\frac{1}{2}}.$$

□

Proof of Theorem 4.4 We need to prove that $\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1(n)}{\sqrt{n+c}}$, where θ_n is the batchTD iterate that is not projected, and $K_1(n)$ is as defined in Theorem 4.4. Once we have Proposition 8.2 in place, the bound mentioned before follows using a completely parallel argument to that used in Sect. 8.2.3 to prove the bound in expectation in Theorem 4.2 for projected batchTD. □

8.4 Proofs of finite time bounds for iterate averaged batchTD

For establishing the bounds in expectation and high probability, we follow the technique from Fathi and Frikha (2013), where the authors provide concentration bounds for general stochastic approximation schemes. However, unlike them, we make all the constants explicit and more importantly, we provide an explicit iteration index n_0 after which the distance between averaged iterate $\bar{\theta}_n$ and LSTD solution $\hat{\theta}_T$ is nearly of the order $O(1/n)$. For providing such a n_0, we have to deviate from Fathi and Frikha (2013) in several steps of the proof.

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, i.e.,

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{1}{n+1}\sum_{k=0}^{n}\mathbb{E}\left\|\theta_k - \hat{\theta}_T\right\|_2. \tag{70}$$

For simplifying the RHS above, we apply the bounds in expectation given in Proposition 8.1. Recall that the rates in Theorem 4.2 are for step sizes of the form $\gamma_n = \frac{c_0 c}{c+n}$, while iterate averaged batchTD uses a different step size sequence. In the following, we specialize the bound in expectation in Proposition 8.1 for the new choice of step-size sequence and subsequently, average the resulting bound using (70) to obtain the final rate in expectation in Theorem 5.1. Let $\gamma_n \triangleq c_0(c/(c+n))^\alpha$. We assume $n > n_0$, i.e.,

$$\frac{c_0 c^\alpha}{(c+n)^\alpha}(1+\beta)^2\Phi^4_{\max} < \mu. \tag{71}$$

Using Proposition 8.1 followed by a split of the individual terms into those before and after n_0, we have

$$
\begin{aligned}
\mathbb{E}\Big(\left\|\theta_n - \hat{\theta}_T\right\|_2\Big)^2 \leq{}& \left[\prod_{k=1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right)\left\|z_0\right\|_2\right]^2 \\
&+ 4\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
={}& \left[\prod_{k=1}^{n_0}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right) \times \prod_{k=n_0+1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right)\left\|z_0\right\|_2\right]^2 \\
&+ 4\sum_{k=1}^{n_0}\gamma_k^2\left[\prod_{j=k}^{n-1}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
&+ 4\sum_{k=n_0+1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
\leq{}& \left[\prod_{k=1}^{n_0}\left(1 + (1+\beta)\boldsymbol{\Phi}^2_{\max}c_0\right)^2\prod_{k=n_0+1}^{n}\left(1 - \gamma_k(2\mu - \gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right)\left\|z_0\right\|_2\right]^2 \\
&+ 4\sum_{k=1}^{n_0}c_0^2\left[\prod_{j=k}^{n_0}\left(1 + (1+\beta)\boldsymbol{\Phi}^2_{\max}c_0\right)^2\right]^2\left[\prod_{j=n_0+1}^{n-1}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2 \\
&\times\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
&+ 4\sum_{k=n_0+1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1 - \gamma_j(2\mu - \gamma_j(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right]^2\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2
\end{aligned}
\tag{72}
$$

$$
\begin{aligned}
\leq{}& \left[\left(1 + c_0(1+\beta)\boldsymbol{\Phi}^2_{\max}\right)^{2n_0}\prod_{k=n_0+1}^{n}\left(1 - \frac{\mu c_0 c^\alpha}{(c+k)^\alpha}\right)\left\|z_0\right\|_2\right]^2 \\
&+ 4n_0c_0^2\left(1 + c_0(1+\beta)\boldsymbol{\Phi}^2_{\max}\right)^{4n_0}\left[\prod_{j=n_0+1}^{n-1}\left(1 - \frac{\mu c_0 c^\alpha}{(c+j)^\alpha}\right)\right]\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
&+ 4\sum_{k=n_0+1}^{n}\frac{c_0^2c^{2\alpha}}{(c+k)^{2\alpha}}\left[\prod_{j=k}^{n-1}\left(1 - \frac{\mu c_0 c^\alpha}{(c+j)^\alpha}\right)\right]^2\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
\leq{}& \left[\exp\left(2c_0(1+\beta)\boldsymbol{\Phi}^2_{\max}n_0\right)\exp\left(-\mu c_0\sum_{k=n_0+1}^{n}\frac{c^\alpha}{(c+k)^\alpha}\right)\left\|z_0\right\|_2\right]^2 \\
&+ 4n_0c_0^2\exp\left(4c_0(1+\beta)\boldsymbol{\Phi}^2_{\max}n_0\right)\exp\left(-2\mu c_0\sum_{j=n_0+1}^{n-1}\frac{c^\alpha}{(c+j)^\alpha}\right) \\
&\times\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2 \\
&+ 4\sum_{k=n_0+1}^{n}\frac{c_0^2c^{2\alpha}}{(c+k)^{2\alpha}}\exp\left(-2\mu c_0\sum_{j=k}^{n-1}\frac{c^\alpha}{(c+j)^\alpha}\right)\left(R_{\max} + (1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)^2.
\end{aligned}
\tag{73}
$$

In the above, the inequality in (72) can be inferred from the following:

$$\begin{aligned}\left(1-\gamma_k(2\mu-\gamma_k(1+\beta)^2\boldsymbol{\Phi}^4_{\max})\right) &\le \left(1+2(1+\beta)\boldsymbol{\Phi}^2_{\max}\gamma_k+(1+\beta)^2\boldsymbol{\Phi}^4_{\max}\gamma_k^2\right)\\ &\le \left(1+(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0\right)^2,\end{aligned} \tag{74}$$

where we have used the fact that $\mu > 0$ and $\gamma_k < c_0$. To obtain the inequality in (73), we have split the product at n_0, and, when $k \le n_0$, we have used $(1+x)^{n_0} = e^{n_0 \ln(1+x)} \le e^{xn_0}$ and when $k > n_0$, we have applied (71). For the final inequality above, we have exponentiated the logarithm of the products, and used the inequality $\ln(1+x) < x$ in several places.

With C_1 and C_2 as defined in the statement of Theorem 5.1, we have that

$$\begin{aligned}\mathbb{E}\left\|\theta_n-\hat{\theta}_T\right\|_2 \le{}& C_1\exp\left(-c_0\mu c^\alpha\left((n+c)^{1-\alpha}-(n_0+c+1)^{1-\alpha}\right)\right)\left\|\theta_0-\hat{\theta}_T\right\|_2\\ &+\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}^2_{\max}\right).\Bigg(4n_0c_0^2C_1^2\exp\left(-2c_0\mu c^\alpha((n+c)^{1-\alpha}-(n_0+c+1)^{1-\alpha}\right)\\ &+\sum_{k=n_0+1}^{n}c_0^2\left(\frac{c}{k+c}\right)^{2\alpha}\exp\left(-2c_0\mu c^\alpha((n+c)^{1-\alpha}-(k+c)^{1-\alpha}\right)\Bigg)^{\frac{1}{2}}\end{aligned} \tag{75}$$

$$\begin{aligned}={}&\exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right)\\ &\times\Bigg[C_1C_2\left\|\theta_0-\hat{\theta}_T\right\|_2+\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)\\ &\times\left\{4n_0c_0^2C_1^2C_2^2+\sum_{k=n_0+1}^{n}c_0^2\left(\frac{c}{k+c}\right)^{2\alpha}\exp\left(2c_0\mu c^\alpha((k+c)^{1-\alpha}\right)\right\}^{\frac{1}{2}}\Bigg]\\ \le{}&\exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right)\\ &\times\Bigg[C_1C_2\left\|\theta_0-\hat{\theta}_T\right\|_2+\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)\\ &\quad\times\left\{4n_0c_0^2C_1^2C_2^2+c^{2\alpha}c_0^2\int_1^{n+c}x^{-2\alpha}\exp\left(2c_0\mu c^\alpha x^{1-\alpha}\right)dx\right\}^{\frac{1}{2}}\Bigg]\end{aligned} \tag{76}$$

$$\begin{aligned}\le{}&\exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right)\\ &\times\Bigg[C_1C_2\left\|\theta_0-\hat{\theta}_T\right\|_2+\left(R_{\max}+(1+\beta)H\boldsymbol{\Phi}^2_{\max}\right)\\ &\times\Bigg\{4n_0c_0^2C_1^2C_2^2+c^{2\alpha}c_0^2\left(2c_0\mu c^\alpha\right)^{\frac{2\alpha}{1-\alpha}}\\ \times&\int_{\left(2c_0\mu c^\alpha\right)^{1/(1-\alpha)}}^{(n+c)\left(2c_0\mu c^\alpha\right)^{1/(1-\alpha)}}y^{-2\alpha}\exp(y^{1-\alpha})dy\Bigg\}^{\frac{1}{2}}\Bigg].\end{aligned} \tag{77}$$

In the above, the inequality in (75) follows by an application of Jensen's Inequality together with the fact that $\sum_{j=k}^{n-1}(c+j)^{-\alpha} \ge \int_{j=k}^{n}(c+j)^{-\alpha}dj = (c+n)^{1-\alpha}-(c+k)^{1-\alpha}$. To obtain the inequality in (76), we have upper bounded the sum with an integral, the validity of which follows from the observation that $x \mapsto x^{-2\alpha}e^{x^{1-\alpha}}$ is convex for $x \ge 1$. Finally, for arriving at the inequality in (77), we have applied the change of variables $y = (2c_0\mu c^\alpha)^{1/(1-\alpha)}x$.

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

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

and furthermore, since $y \mapsto y^{-2\alpha}\exp(y^{1-\alpha})$ is non-decreasing for $y \leq \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}$, we have

$$\int_1^{\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}} y^{-2\alpha}\exp(y^{1-\alpha})dy \leq e\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}.$$

Plugging these into (77), we obtain

$$\begin{aligned}
\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \leq{}& \exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right) \\
&\times\Bigg(C_1C_2\left\|\theta_0 - \hat{\theta}_T\right\|_2 + \sqrt{e}\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{2(1-\alpha)}}c^\alpha c_0\left(2c_0\mu c^\alpha\right)^{\frac{\alpha}{(1-\alpha)}} \\
&+ 2c_0C_1C_2\left(R_{\max} + (1+\beta)H\Phi_{\max}^2\right)\sqrt{n_0}\Bigg) \\
&+ \sqrt{\frac{2}{1-\alpha}}\left(R_{\max} + (1+\beta)H\Phi_{\max}^2\right)c^\alpha c_0\left(2c_0\mu c^\alpha\right)^{\frac{\alpha}{2(1-\alpha)}}.(n+c)^{-\frac{\alpha}{2}}
\end{aligned} \tag{78}$$

The bound in expectation in the theorem statement can be inferred by using the inequality above in

$$\mathbb{E}\left\|\bar{\theta}_{n+1} - \hat{\theta}_T\right\|_2 \leq \frac{1}{n+1}\sum_{k=0}^{n}\mathbb{E}\left\|\theta_k - \hat{\theta}_T\right\|_2,$$

followed by a straightforward bound on the sum of the first exponential term on the RHS of (78), using the constant C_0. □

Proof of the high probability bound in Theorem 5.1 The proof of the high probability bound is considerably more involved than the proof of the bound in expectation in Theorem 5.1. We first state and prove a bound on the error in high probability for the averaged iterates in Proposition 8.3 below. This result is for general step-size sequences, and can be seen as the iterate average counterpart to Proposition 8.1. □

Proposition 8.3 *Let $z_n = \bar{\theta}_n - \hat{\theta}_T$. Under (A1)–(A3) we have, for all $\epsilon \geq 0$ and $\forall n \geq 1$,*

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

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

Proof Recall that z_n denotes the error of the algorithm at time n, which in this case is $z_n = \bar{\theta}_n - \hat{\theta}_T$. The proof follows the scheme of the proof of Proposition 8.1, part (1), given in Sect. 8.2:

Step 1 As before, we decompose the centered error $\|z_n\|_2 - \mathbb{E}\|z_n\|_2$ as follows:

$$\|z_n\|_2 - \mathbb{E}\|z_n\|_2 = \sum_{k=1}^{n} D_k, \tag{79}$$

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

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 8.1 in Sect. 8.2 that the random variable $\Theta_n^k(\theta)$ 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

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

Then, letting f and f' denote 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

$$\begin{aligned}\mathbb{E}\left\|\bar{\Theta}_n^k(\bar{\theta}_{k-1},\theta) - \bar{\Theta}_n^k(\bar{\theta}_{k-1},\theta')\right\|_2 &= \mathbb{E}\left\|\frac{1}{n+1}\sum_{l=k}^{n}\left(\Theta_l^k(\theta) - \Theta_l^k(\theta')\right)\right\|_2\\ &\leq \frac{1}{n+1}\sum_{l=k}^{n}\prod_{j=k+1}^{l}\left(1-\gamma_j\left(2\mu-\gamma_j(1+\beta)^2\Phi_{\max}^4\right)\right)^{1/2}\|f-f'\|_2\end{aligned} \tag{80}$$

where we have used (42) derived in Step 2 of the proof the high probability bound in Proposition 8.1. Hence, as in Step 2 of the proof of Proposition 8.1, part (1), we find that g_k is L_k -Lipschitz in the random innovation at time k, and this implies D_k is L_k-Lipschitz.

Step 3 follows in a similar manner to the proof of Proposition 8.1, part (1). □

We now bound the sum of squares of the Lipschitz constants L_m when the iterates are averaged, and the step-sizes are chosen to be $\gamma_n = c_0\left(\frac{c}{c+n}\right)^{\alpha}$ 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 $c_0\mu c \in (1,\infty)$.

Lemma 8.1 *Under conditions of Theorem 5.1, we have*

$$\sum_{i=1}^{n} L_i^2 \leq \frac{n_0}{(n+1)^2}\left[\frac{e^{(1+\beta)\Phi_{\max}^2 c_0(2n_0+1)}}{(1+\beta)\Phi_{\max}^2}\right]^2 \tag{81}$$

$$+\frac{1}{\mu^2}\left\{2^\alpha+\left[\left[\frac{2\alpha}{c_0\mu c^\alpha}\right]^{\frac{1}{1-\alpha}}+\frac{2(1-\alpha)(c_0\mu)^\alpha}{\alpha}\right]\right\}^2\frac{1}{n+1}. \tag{82}$$

Proof Recall from the statement of Theorem 5.1 that n satisfies,

$$\frac{c_0c^\alpha}{(c+n)^\alpha}(1+\beta)^2\boldsymbol{\Phi}^4_{\max}<\mu. \tag{83}$$

Recall also from the formula in Proposition 8.3, that:

$$L_i=\frac{\gamma_i}{n+1}\left(\sum_{l=i+1}^{n-1}\prod_{j=i}^{l}\left(1-\gamma_{j+1}(2\mu-(1+\beta)^2\boldsymbol{\Phi}^4_{\max}\gamma_{j+1}))\right)^{1/2}\right).$$

We split the bound on the sum into two terms as follows:

$$\sum_{i=1}^{n}L_i^2=\sum_{i=1}^{n_0-1}L_i^2+\sum_{i=n_0}^{n}L_i^2. \tag{84}$$

The first term in (84) is simplified as follows:

$$\begin{aligned}\sum_{i=1}^{n_0-1}L_i^2&=\sum_{i=1}^{n_0-1}\left[\frac{\gamma_i}{n+1}\left(\sum_{l=i+1}^{n_0}\prod_{j=i}^{l}\left(1-\gamma_{j+1}(2\mu-(1+\beta)^2\boldsymbol{\Phi}^4_{\max}\gamma_{j+1}))\right)^{1/2}\right)\right]^2\\&\le\frac{1}{(n+1)^2}\sum_{i=1}^{n_0-1}\left[c_0\left(\sum_{l=i+1}^{n_0}\prod_{j=i}^{l}\left(1+(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0)\right)\right)\right]^2\end{aligned} \tag{85}$$

$$\le\frac{1}{(n+1)^2}\sum_{i=1}^{n_0-1}\left[c_0(1+(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0)^{2n_0}\sum_{l=1}^{n_0}\left(1+(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0\right)^{-l}\right]^2 \tag{86}$$

$$\le\frac{1}{(n+1)^2}c_0^2n_0\left[\frac{(1+(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0)^{2n_0+1}}{(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0}\right]^2 \tag{87}$$

$$\le\frac{n_0}{(n+1)^2}\left[\frac{e^{(1+\beta)\boldsymbol{\Phi}^2_{\max}c_0(2n_0+1)}}{(1+\beta)\boldsymbol{\Phi}^2_{\max}}\right]^2. \tag{88}$$

In the above, the inequality in (85) follows from (74). , while the inequality in (85) applies the form of the step sizes. In obtaining the inequality in (86), we have replaced i with 1. For the inequality in (87), we have used the formula for the sum of a geometric series, and for the final inequality we have used that $(1+x)^{n_0}=e^{n_0\ln(1+x)}\le e^{xn_0}$.

We now analyze the second term in (84). Notice that

$$
\begin{aligned}
\sum_{i=n_0}^{n} L_i^2 &= \sum_{i=n_0}^{n}\left[\frac{\gamma_i}{n+1}\left(\sum_{l=i+1}^{n-1}\prod_{j=i}^{l}\left(1-\gamma_{j+1}(2\mu-(1+\beta)^2\boldsymbol{\Phi}_{\max}^4\gamma_{j+1}))\right)^{1/2}\right)\right]^2\\
&\le \frac{1}{(n+1)^2}\sum_{i=n_0}^{n}\left[\gamma_i\left(\sum_{l=i+1}^{n-1}\exp\left(-\sum_{j=i}^{l}\gamma_{j+1}(2\mu-(1+\beta)^2\boldsymbol{\Phi}_{\max}^4\gamma_{j+1}))\right)\right)\right]^2\\
&< \frac{1}{(n+1)^2}\sum_{i=n_0}^{n}\underbrace{\left[c_0\left(\frac{c}{c+i}\right)^{\alpha}\left(\sum_{l=i+1}^{n-1}\exp\left(-c_0\mu\sum_{j=i}^{l}\left(\frac{c}{c+j}\right)^{\alpha}\right)\right)\right]^2}_{\triangleq(A)}.
\end{aligned}
\tag{89}
$$

To produce the final bound, we bound the summand (A) highlighted in line (89) by a constant, uniformly over all values of i and n, as follows:

$$
\begin{aligned}
&\sum_{l=i+1}^{n-1}\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^{\alpha}\right)\\
&\quad=\sum_{l=i+1}^{n-1}\left[\left(\frac{c}{c+l}\right)^{\alpha}\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^{\alpha}\right)\right]\left(\frac{c+l}{c}\right)^{\alpha}\\
&\quad\le\sum_{l=i+1}^{n-1}\left[\frac{1}{c_0\mu}\left(\exp\left(-c_0\mu\sum_{j=1}^{l-1}\left(\frac{c}{c+i}\right)^{\alpha}\right)\right.\right.\\
&\quad\left.\left.-\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^{\alpha}\right)\right)\right]\left(\frac{c+l}{c}\right)^{\alpha}
\end{aligned}
\tag{90}
$$

$$
\begin{aligned}
&=\frac{1}{c_0\mu}\left\{-\left(\frac{c}{c+n}\right)^{-\alpha}\exp\left(-c_0\mu\sum_{j=1}^{n}\left(\frac{c}{c+i}\right)^{\alpha}\right)\right.\\
&\quad+\left(\frac{c}{c+i+1}\right)^{-\alpha}\exp\left(-c_0\mu\sum_{j=1}^{i+1}\left(\frac{c}{c+i}\right)^{\alpha}\right)\\
&\quad\left.+\sum_{l=i+1}^{n-1}\exp\left(-c_0\mu\sum_{j=1}^{l}\left(\frac{c}{c+i}\right)^{\alpha}\right)\left[\left(\frac{c}{c+l+1}\right)^{-\alpha}-\left(\frac{c}{c+l}\right)^{-\alpha}\right]\right\},
\end{aligned}
\tag{91}
$$

where the inequality in (90) follows from the convexity of $e^{-\frac{c_0\mu}{2}x}$, while that in (91) follows by applying an Abel transform.

From the foregoing, the summand term (A) highlighted in (89) can be bounded by

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

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

$$\sum_{l=i+1}^{n-1} \exp\left(-c_0\mu \frac{c^\alpha((c+l)^{1-\alpha}-(c+i)^{1-\alpha})}{(1-\alpha)}\right)((c+l+1)^\alpha-(c+l)^\alpha) \tag{92}$$

$$\begin{aligned}
&\le \sum_{l=i+1}^{n-1} \exp\left(-c_0\mu \frac{c^\alpha((c+l)^{1-\alpha}-(c+i)^{1-\alpha})}{(1-\alpha)}\right)\alpha(c+l)^{-(1-\alpha)} \\
&\le \alpha \exp\left(c_0\mu \frac{c^\alpha(c+i)^{1-\alpha}}{(1-\alpha)}\right)\left[\int_i^{n-1} \exp\left(-c_0\mu \frac{c^\alpha(c+l)^{1-\alpha}}{(1-\alpha)}\right)(c+l)^{-(1-\alpha)}dl\right] \qquad (93) \\
&= \alpha \exp\left(c_0\mu \frac{c^\alpha(c+i)^{1-\alpha}}{(1-\alpha)}\right)\left[\int_{c_0\mu(c+i)^{1-\alpha}}^{c_0\mu(c+n-1)^{1-\alpha}} \exp\left(-\frac{c^\alpha l}{(1-\alpha)}\right)l^{\frac{2\alpha-1}{1-\alpha}}dl\right].
\end{aligned}$$

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 $c_0\mu(c+i+1)^{1-\alpha} \ge 2\alpha/c^\alpha$,

$$\begin{aligned}
&\exp\left(\frac{c^\alpha(c+i)^{1-\alpha}}{(1-\alpha)}\right)\int_{c_0\mu(c+i+1)^{1-\alpha}}^{c_0\mu(c+n)^{1-\alpha}} \exp\left(-\frac{c^\alpha l}{(1-\alpha)}\right)l^{\frac{2\alpha-1}{1-\alpha}}dl \\
&\le (c_0\mu)^{\frac{2\alpha}{1-\alpha}}(c+i+1)^{2\alpha}\int_{c_0\mu(c+i+1)^{1-\alpha}}^{c_0\mu(c+n)^{1-\alpha}} l^{\frac{-1}{1-\alpha}}dl \le \frac{1-\alpha}{\alpha}((c_0\mu(c+i+1))^\alpha.
\end{aligned}$$

When $c_0\mu(c+i+1)^{1-\alpha} < 2\alpha/c^\alpha$ we can bound the summand of (92) by 1, and

$$\begin{aligned}
c_0\mu(c+i+1)^{1-\alpha} < \frac{2\alpha}{c^\alpha} &\implies (c+i+1)^{1-\alpha} < \frac{2\alpha}{c_0\mu c^\alpha} \\
&\implies i < \left[\frac{2\alpha}{c_0\mu c^\alpha}\right]^{\frac{1}{1-\alpha}} - c - 1.
\end{aligned}$$

Hence, we conclude that

$$\sum_{i=n_0}^{n} L_i^2 \le \frac{1}{\mu^2}\left\{2^\alpha + \left[\left[\frac{2\alpha}{c_0\mu c^\alpha}\right]^{\frac{1}{1-\alpha}} + \frac{2(1-\alpha)(c_0\mu)^\alpha}{\alpha}\right]\right\}^2 \frac{1}{n+1}.$$

□

Proof (High probability bound in Theorem 5.1) Once we have established the bound in expectation for batchTD with iterate averaging, and the bound on sum of squares of Lipschitz constants in the lemma above, the proof of the high probability bound is straightforward, and follows by arguments similar to that used in establishing the corresponding claim for non-averaged batchTD (see Sect. 8.2.3). □

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	0.01
	Green	0.06
$q_i < \mathcal{L}_1$ and $t_i \geq \mathcal{T}_1$	Red	0.02
	Green	0.05
$\mathcal{L}_1 \leq q_i < \mathcal{L}_2$ and $t_i < \mathcal{T}_1$	Red	0.03
	Green	0.04
$\mathcal{L}_1 \leq q_i < \mathcal{L}_2$ and $t_i \geq \mathcal{T}_1$	Red	0.04
	Green	0.03
$q_i \geq \mathcal{L}_2$ and $t_i < \mathcal{T}_1$	Red	0.05
	Green	0.02
$q_i \geq \mathcal{L}_2$ and $t_i \geq \mathcal{T}_1$	Red	0.06
	Green	0.01

9 Traffic control application

9.1 Simulation setup

The idea behind the experimental setup is to study both LSPI and the variant of LSPI, fLSPI, where we use batchTDQ 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 Prashanth and Bhatnagar (2011), the traffic signal control MDP is formulated as follows:

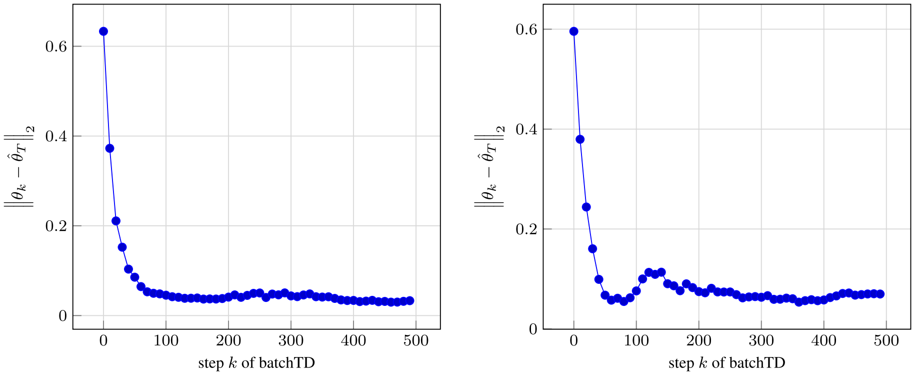
State $\quad s_t = \big(q_1(t), \ldots, q_L(t), t_1(t), \ldots, t_L(t)\big),$

Action $\quad a_t$ 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.

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 Prashanth and Bhatnagar (2012), 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,



(a) Tracking error on 7x9-grid network **(b)** Tracking error on 14x9-grid network

Fig. 2 Tracking error of batchTDQ in iteration 1 of fLSPI on two grid networks

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.

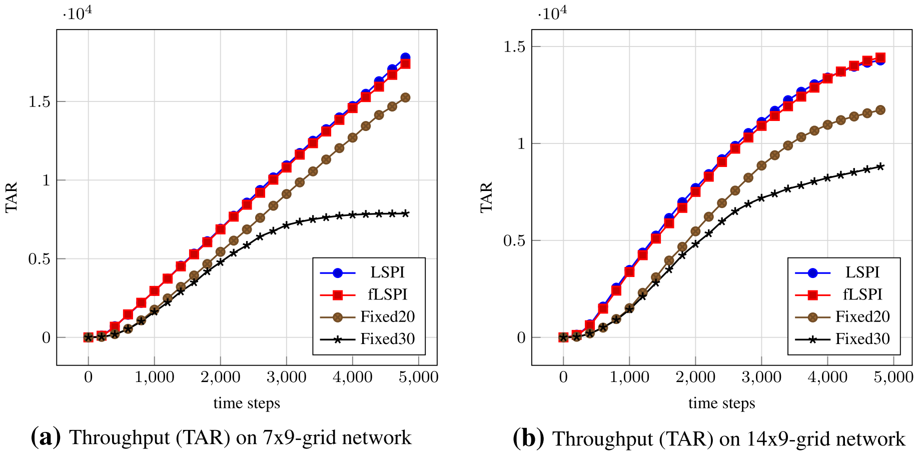
Note that the feature selection scheme depends on certain thresholds $\mathcal{L}_1$ and $\mathcal{L}_2$ on the queue length and $\mathcal{T}_1$ on the elapsed times. The motivation for using such graded thresholds is owing to the fact that queue lengths are difficult to measure precisely in practice. 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, by Prashanth and Bhatnagar (2012).

We implement both LSPI as well as fLSPI for the above problem. The experiments involve two stages—an initial training stage where LSPI/fLSPI is run to find an approximately optimal policy, and a test stage where ten independent simulations are run using the policy that LSPI/fLSPI converged to in the training stage. In the training stage, for both LSPI and fLSPI, we collect $T = 10000$ samples from an exploratory policy that picks the actions in a uniformly random manner. For both LSPI and fLSPI, we set $\beta = 0.9$ and $\epsilon = 0.1$. We set τ, the number of batchTDQ iterations in fLSPI, to 500. This choice is motivated by an experiment where we observed that at 500 steps, batchTD is already very close to LSTDQ, and taking more steps did not result in any significant improvements for fLSPI. 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 batchTDQ is set as recommended by Theorem 4.2.

9.2 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. For batchTDQ, which is the policy evaluation algorithm in fLSPI, we also report the tracking error, which measures the distance in ℓ^2 norm between the batchTD iterate θ_k, $k = 1, \dots, \tau$ and LSTDQ solution $\hat{\theta}_T$.

We report the tracking error and total arrived road users (TAR) in Figs. 2 and 3, respectively. The run-times obtained from our experimental runs for LSPI and fLSPI is presented in Fig. 4. Iteration 1 for fLSPI is used for reporting the tracking error and we observed similar behavior across iterations, i.e., we observed that batchTD iterate θ_τ is close to the

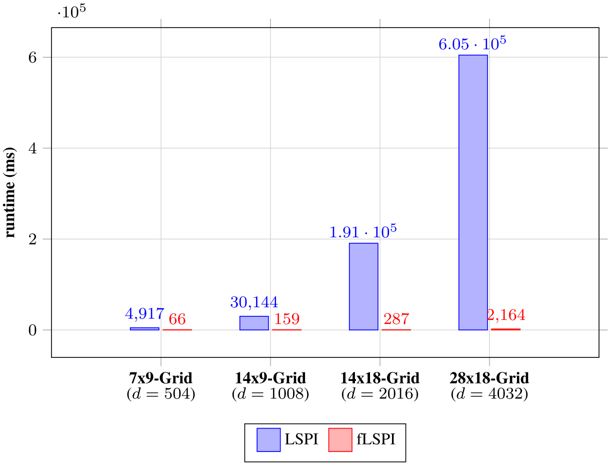


(a) Throughput (TAR) on 7x9-grid network

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

Fig. 3 Performance comparison of LSPI and fLSPI using throughput (TAR) on two grid networks

Fig. 4 Run-times of LSPI and fLSPI on four road networks



corresponding LSTDQ solution in each iteration of fLSPI. The experiments are performed for four different grid networks of increasing size and hence, increasing feature dimension.

From Fig. 2a, b, we observe that batchTD algorithm converges rapidly to the corresponding LSTDQ solution. Further, from the runtime plots (see Fig. 4), we notice that fLSPI is several orders of magnitude faster than regular LSPI. From a traffic application standpoint, we observe in Fig. 3a, b that fLSPI results in a throughput (TAR) performance that is on par with LSPI. Moreover, the throughput observed for LSPI/fLSPI 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 Fig. 3a, b 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 gives performance comparable to that of LSPI, but at a lower computational cost.

10 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.

10.1 Least squares regression and SGD

In this setting, we are given a set of samples $\mathcal{D} \triangleq \{(x_i, y_i), i = 1, \ldots, T\}$ with the underlying observation model $y_i = x_i^\top \theta^* + \xi_i$ (ξ_i is a bounded, zero-mean random variable, and θ^* is an unknown parameter). The least squares estimate $\hat{\theta}_T$ minimizes $\sum_{i=1}^T (y_i - \theta^\top x_i)^2$. It can be shown that $\hat{\theta}_T = \bar{A}_T^{-1} b_T$, where $\bar{A}_T = T^{-1} \sum_{i=1}^T x_i x_i^\top$ 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 $O(d^2 T)$. As in the case of the batchTD algorithm, we update the SGD iterate θ_n using a SA scheme as follows (starting with an arbitrary θ_0),

$$\theta_n = \theta_{n-1} + \gamma_n (y_{i_n} - \theta_{n-1}^\top x_{i_n}) x_{i_n}, \tag{94}$$

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

Unlike batchTD which is a fixed point iteration, the above is a stochastic gradient descent procedure. Nevertheless, using the same proof template as for batchTD 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.

10.2 Main results

10.2.1 Assumptions

As in the case of batchTD, 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., $\|x_i\|_2 \le \Phi_{\max}$, for $i = 1, \ldots, T$.
(A3) The noise $\{\xi_i\}$ is i.i.d., zero mean and $|\xi_i| \le \sigma$, 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 batchTD, in cases when the fourth assumption is not satisfied we can employ either explicit regularization or iterate averaging to produce similar results.

10.2.2 Asymptotic convergence

An analogue of Theorem 4.1 holds as follows:

Theorem 10.1 *Under (A1)–(A4), the iterate $\theta_n \to \hat{\theta}_T$ a.s. as $n \to \infty$, where θ_n is given by* (96) *and $\hat{\theta}_T = \bar{A}_T^{-1}\bar{b}_T$.*

Proof Follows in a similar manner as the proof of Theorem 4.1. □

10.2.3 Finite time bounds

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

Theorem 10.2 (Error Bound for iterates of SGD) *Assume (A1)–(A4). Choosing $\gamma_n = \frac{c_0 c}{(c+n)}$, and c such that $c_0 \boldsymbol{\Phi}^2_{\max} \in (0,1)$ 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

$$K_1^{LS}(n) \triangleq \frac{\sqrt{c^{c_0 c \mu}}\left\|\theta_0 - \hat{\theta}_T\right\|_2}{(n+c)^{\mu c_0 c - \frac{1}{2}}} + \frac{2ec_0 c h(n)}{2c_0 c\mu - 1},$$

$$K_2^{LS}(n) \triangleq 2\sqrt{ec_0 c h(n)}\sqrt{\frac{\log \delta^{-1}}{\mu c_0 c - 1}} + K_1(n).$$

In the above, $h(n) \triangleq \left(\|\theta^*\|_2 + \left\|\theta_0\right\|_2 + \sigma \boldsymbol{\Phi}_{\max}\Gamma_n\right)\boldsymbol{\Phi}^2_{\max} + \sigma\boldsymbol{\Phi}_{\max}$.

Proof See Sect. 10.4. □

With step-sizes specified in Theorem 10.2, we see that the initial error is forgotten faster than the sampling error, which vanishes at the rate $\tilde{O}\left(n^{-1/2}\right)$, where $\tilde{O}(\cdot)$ is like $O(\cdot)$ with the log factors discarded. Thus, the rate derived in Theorem 10.2 matches the asymptotically optimal convergence rate for SGD type schemes (cf. Nemirovsky and Yudin 1983).

10.3 Iterate averaging

The expectation and high-probability bounds in Theorem 10.2 as well as earlier works on SGD (cf. Hazan and Kale 2011) 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 10.3 (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 $c_0\boldsymbol{\Phi}^2_{\max} \in (0,1)$, we have, for any $\delta > 0$,*

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_1^{IA}(n)}{(n+c)^{\alpha/2}} \text{ and } \mathbb{P}\left(\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{K_2^{IA}(n)}{(n+c)^{\alpha/2}}\right) \geq 1-\delta, \tag{95}$$

where, writing $C_0 \triangleq \sum_{n=1}^{\infty} \exp(-\mu c_0 c^\alpha n^{1-\alpha}$ *and* $C_1 \triangleq (\exp\left(c_0 \mu c^\alpha (1+c)^{1-\alpha}\right)$,

$$\begin{aligned} K_1^{IALS}(n) \triangleq C_0 &\left(C_1 \left\|\theta_0 - \theta_T\right\|_2 + 2h(n) c^\alpha c_0 \left(2 c_0 \mu c^\alpha\right)^{\frac{\alpha}{(1-\alpha)}} \sqrt{e} \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{2(1-\alpha)}} \right) \\ &+ 2h(n) c^\alpha c_0 \left(2 c_0 \mu c^\alpha\right)^{\frac{\alpha}{2(1-\alpha)}} (n+c)^{1-\frac{\alpha}{2}}, \end{aligned}$$

and

$$K_2^{IALS}(n) \triangleq \frac{4\sqrt{\log \delta^{-1}}}{\mu^2 c_0^2} \frac{\frac{1}{\mu}\left\{2^\alpha + \left[\left[\frac{2\alpha}{c_0 \mu c^\alpha}\right]^{\frac{1}{1-\alpha}} + \frac{2(1-\alpha)(c_0\mu)^\alpha}{\alpha}\right]\right\}}{(n+c)^{(1-\alpha)/2}} + K_1^{IALS}(n).$$

Proof The proof is similar to that of Theorem 5.1 and is provided in "Appendix 1". □

Remark 15 Note that, unlike in the case of Theorem 5.1, there is no dependence on a quantity n_0 which defines a time when the step sizes have become sufficiently small. This is because for the regression setting here, the assumption that $c_0 \Phi_{\max}^2 \in (0, 1)$ already ensures that the step sizes are sufficiently small. If it was not possible to set c_0 in this way, then a similar bound including a dependence on the smallest n such that $\gamma_n \Phi_{\max}^2 < 1$ would be derivable.

10.4 Proofs for least squares regression extension

The overall schema of the proof here is the same as that used to prove Theorem 4.2. Proposition 10.1 below is an analogue of Proposition 8.1 for the least squares setting. From this proposition the derivation of the rates in Theorem 10.2 is essentially the same as for Theorem 4.2 and $\hat{\theta}_T = \bar{A}_T^{-1} b_T$.

Proposition 10.1 *Let* $z_n = \theta_n - \hat{\theta}_T$*, where* θ_n *is given by* (94)*, Under (A1)–(A4), and assuming that* $\gamma_n \Phi_{\max}^2 \leq 1$ *for all n, we have* $\forall \epsilon > 0$,

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

$$\mathbb{P}\left(\left\|z_n\right\|_2 - \mathbb{E}\left\|z_n\right\|_2 \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{4h(n)^2 \sum_{i=1}^n L_i^2}\right), \tag{96}$$

where

$$L_i \triangleq \gamma_i \prod_{j=i}^{n-1} (1 - \gamma_{j+1} \mu (2 - \Phi_{\max}^2 \gamma_{j+1}))^{1/2},$$

$$h(n) \triangleq \left(\|\theta^*\|_2 + \left\|\theta_0\right\|_2 + \sigma \Phi_{\max} \Gamma_n\right) \Phi_{\max}^2 + \sigma \Phi_{\max},$$

and $\Gamma_n \triangleq \sum_{i=1}^n \gamma_i$.

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

$$\mathbb{E}\left(\left\|z_n\right\|_2\right)^2 \leq \underbrace{\prod_{j=1}^{n}\left(1-\mu\gamma_j\right)\left\|\theta_0-\hat{\theta}_T\right\|_2}_{\textbf{initial error}} + \underbrace{\left(\sum_{k=1}^{n-1} 4h(k)^2\gamma_{k+1}^2\left[\prod_{j=k+1}^{n}\left(1-\mu\gamma_j\right)\right]^2\right)^{\frac{1}{2}}}_{\textbf{sampling error}}. \tag{97}$$

The proof of the Proposition 10.1 has the same scheme as the proof of Proposition 8.1. 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 8.1:

High-probability bound There are two alterations to the proof of the high probability bound in Proposition 8.1: slightly different Lipschitz constants are derived according to the different form of the random innovation (Step 2 of the proof of Proposition 8.1); 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 8.1).

Bound in expectation The overall scheme of this proof is similar to that used in proving the expectation bound in Proposition 8.2. 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 10.1 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 8.1.

Let $\Theta_j^i(\theta)$ denote the mapping that returns the value of the iterate updated according to (94) 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

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

Using Jensen's inequality, the tower property of conditional expectations, and Cauchy-Schwarz inequality, we can deduce that

$$\begin{aligned}&\mathbb{E}\left[\|\Theta_n^i(\theta) - \Theta_n^i(\theta')\|_2 \mid \Theta_{n-1}^i(\theta), \Theta_{n-1}^i(\theta')\right]\\ &\quad\leq \left[\|I - \gamma_n(2 - \Phi_{\max}^2\gamma_n)\bar{A}_T\|_2^2\|\Theta_{n-1}^i(\theta) - \Theta_{n-1}^i(\theta')\|_2^2\right]^{1/2}.\end{aligned} \tag{98}$$

Notice that since $\gamma_n\Phi_{\max}^2 \in (0, 1)$, the largest eigenvalue of $\gamma_n\bar{A}_T$ must be less than 1. Hence, a repeated application of (98), together with (A1) yields the following

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

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

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

Finally we need to bound the size of the random innovations. Recall that in Proposition 8.1, the bound on the size of the iterates followed from the projection step in the algorithm. In this case, we can derive a bound for the iterates directly:

$$\begin{aligned}
\|\theta_n\|_2 =& \left\|\left[\prod_{k=1}^{n}(I-\gamma_k x_{i_k}x_{i_k}^\top)\right]\theta_0+\sum_{k=1}^{n}\gamma_k\left[\prod_{j=k}^{n}(I-\gamma_j x_{i_j}x_{i_j}^\top)\right]\xi_k x_k\right\|_2 \\
\le& \|\theta_0\|_2+\sigma\Phi_{\max}\sum_{j=1}^{n}\gamma_j,
\end{aligned} \tag{99}$$

where we have used that $\gamma_j x_{i_j} x_j^\top$ is a positive semi-definite matrix. Now, we can bound the random innovation by

$$\begin{aligned}
\left\|(y_{i_n}-\theta_{n-1}^\top x_{i_n})x_{i_n}\right\|_2 &= \left\|(x_{i_n}^\top\theta^*+\xi_{i_n}-\theta_{n-1}^\top x_{i_n})x_{i_n}\right\|_2 \\
&\le \left(\|\theta^*\|_2+\|\theta_0\|_2+\sigma\Phi_{\max}\Gamma_n\right)\Phi_{\max}^2+\sigma\Phi_{\max}=h(n).
\end{aligned}$$

The proof now follows just as in Proposition 8.1. □

Proof of Proposition 10.1 part (2) First we extract a martingale difference from the update rule (94). Let $f_n(\theta)\triangleq(x_{i_n}-(\theta-\hat{\theta}_T)^\top x_{i_n})x_{i_n}$, and let $F(\theta)\triangleq\mathbb{E}(f_n(\theta)\mid\mathcal{F}_{n-1})$, where $\mathcal{F}_{n-1}$ is the σ-field generated by the random variables $\{i_1,\dots,i_{n-1}\}$ 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 the following recursion:

$$z_n=z_{n-1}-\gamma_n\left(z_{n-1}\bar{A}_T-\Delta M_n\right)=\Pi_1^n z_0-\sum_{k=1}^{n}\gamma_k\Pi_{k+1}^n\Delta M_k,$$

where $\Pi_k^n\triangleq\prod_{j=k}^{n}\left(I-\gamma_j\bar{A}_T\right)$.

By Jensen's inequality, we have

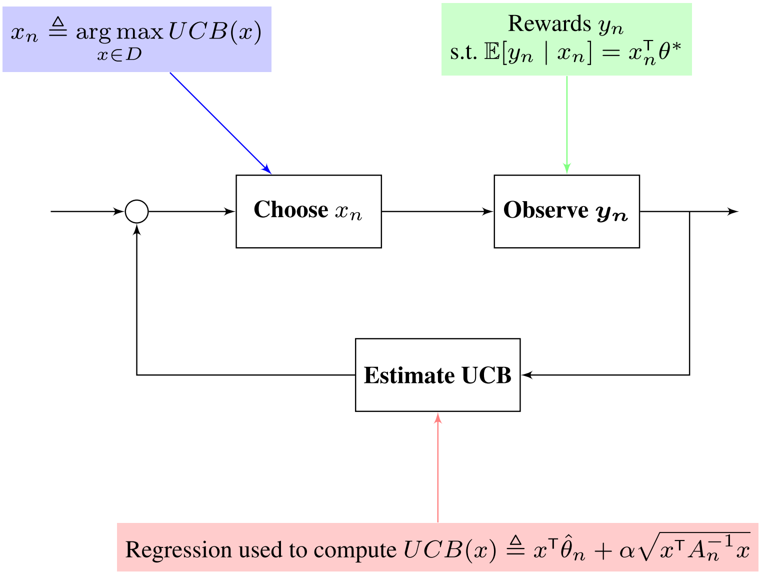

$x_n \triangleq \arg\max_{x\in D} UCB(x)$
Rewards y_n
s.t. $\mathbb{E}[y_n \mid x_n] = x_n^\mathsf{T}\theta^*$
Choose x_n
Observe $\boldsymbol{y_n}$
Estimate UCB
Regression used to compute $UCB(x) \triangleq x^\mathsf{T}\hat{\theta}_n + \alpha\sqrt{x^\mathsf{T}A_n^{-1}x}$

Fig. 5 Operational model of LinUCB

$$\mathbb{E}(\|z_n\|_2) \le \left(\mathbb{E}(\langle z_n, z_n\rangle)\right)^{\frac{1}{2}} = \left(\mathbb{E}\left\|\Pi_1^n z_0\right\|_2^2 + \sum_{k=1}^{n}\gamma_k^2\mathbb{E}\left\|\Pi_{k+1}^n \Delta M_k\right\|_2^2\right)^{\frac{1}{2}} \tag{100}$$

Notice that the largest eigenvalue of $\gamma_n\bar{A}_T$ is smaller than 1, since $\gamma_n\Phi^2_{\max} \in (0,1)$. So, $I - \gamma_n\bar{A}_T$ is positive definite, and, by (A1), has largest eigenvalue $1 - \gamma_n\mu$. Hence

$$\left\|\Pi_{k+1}^n\right\|_2 = \left\|\prod_{j=k+1}^{n}\left(I - \gamma_j\bar{A}_T\right)\right\|_2 \le \prod_{j=k+1}^{n}(1 - \gamma_j\mu). \tag{101}$$

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

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

where we have used the bound in (99). Hence $\mathbb{E}[\|\Delta M_n\|_2^2] \le 4h(n)^2$.

The result now follows from (100) and (101). □

11 Fast LinUCB using SA and application to news-recommendation

11.1 Background for LinUCB

As illustrated in Fig. 5, at each iteration n, the objective is to choose an article from a pool of K articles with respective features $x_1(n), \dots, x_K(n)$. Let x_n denote the chosen article at time n. LinUCB computes a regularized least squares (RLS) solution $\hat{\theta}_n$ based on the chosen arms x_i and rewards y_i seen so far, $i = 1, \dots, n-1$ as follows:

$$\hat{\theta}_n = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta^\top x_i)^2 + \lambda \|\theta\|_2^2. \tag{102}$$

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:

$$\text{UCB}(x_k(n)) \triangleq x_k(n)^\top \hat{\theta}_n + \kappa \sqrt{x_k(n)^\top A_n^{-1} x_k(n)}, k = 1, \dots, K. \tag{103}$$

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

Algorithm 3 fLinUCB-SA

```
Initialization: Set $\theta_0$, $\lambda > 0$ - the regularization parameter, $\gamma_k$ - the step-size sequence.
for $n = 1, 2, \dots$ do
    Observe article features $x_1(n), \dots, x_K(n)$
    Approximate Least Squares Regression using fLS-SA
    for $l = 1 \dots \tau$ do
        Get random sample index: $i_l \sim U(\{1, \dots, n-1\})$
        Update fLS-SA iterate $\theta_l(n)$ as follows:
        $\theta_l(n) = \theta_{l-1}(n) + \gamma_l (y_{i_l} - \theta_{l-1}(n)^\top x_{i_l}) x_{i_l} - \gamma_l \frac{\lambda}{n} \theta_{l-1}(n)$
    end for
    UCB computation using SGD
    for $k = 1 \dots K$ do
        for $l = 1 \dots \tau'$ do
            Get random sample index: $i_l \sim U(\{1, \dots, n-1\})$
            Update SGD iterate $\phi_k(n)$ as follows:
            $\phi_k(l) = \phi_k(l-1) + \gamma_l (n^{-1} x_k(n) - (\phi_k(l-1)^\top x_{i_l}) x_{i_l})$,
        end for
    end for
    Choose article achieving $\arg\max_{k=1,\dots,K} \theta_\tau(n)^\top x_k(n) + \kappa \sqrt{\phi_k(\tau')^\top x_k(n)}$
    Observe the reward $y_n$.
end for
```

11.2 Fast LinUCB using SA (fLinUCB-SA)

We implement a fast 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 (94) 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 for approximating the confidence term of the UCB values (103). For a given arm $k = 1, \dots, K$, let $\hat{\phi}_k(n) = A_n^{-1} x_k(n)$

denote the confidence estimate in the UCB value (103). Recall that $A_n = \sum_{i=1}^{n} x_i x_i^\mathsf{T}$. It is easy to see that $\hat{\phi}_k(n)$ is the solution to the following problem:

$$\min_{\phi} \sum_{i=1}^{n} \frac{(x_i^\mathsf{T}\phi)^2}{2} - \frac{x_k(n)^\mathsf{T}\phi}{n}. \tag{104}$$

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 (104) by using the following iterative scheme:

$$\phi_k(l) = \phi_k(l-1) + \gamma_l(n^{-1}x_k(n) - (\phi_k(l-1)^\mathsf{T} x_{i_l})x_{i_l}), \tag{105}$$

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 $d = 136$-dimensional zero 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 10.2.

Remark 16 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 an SGO 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.

11.3 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 slightly weaker performance as compared to regular LinUCB.

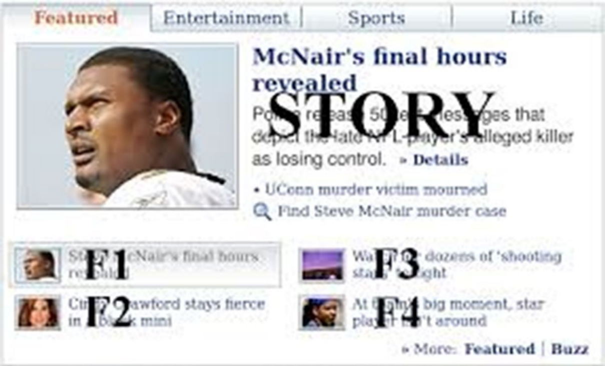
For conducting the experiments, we use the framework provided by the ICML exploration and exploitation challenge (Mary et al. 2012), based on the user click log dataset (Webscope 2011) for the Yahoo! front page today module (see Fig. 6). We run each algorithm on several data files corresponding to different 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 Mary et al. (2012). The evaluation of the algorithms in this framework is done in an off-line manner using a procedure described in Li et al. (2011).

We report the tracking error and runtimes from our experimental runs in Figs. 7 and 8, respectively. As in the case of batchTDQ, 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. 8 are for five different data files corresponding to five days in October, 2009 of the dataset (Webscope 2011), and we compare the classic RLS solver time against fLS-SA time for each day of the dataset considered.

From Fig. 7, 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. 6 The *Featured* tab in Yahoo! Today module (src: Li et al. 2010)


Featured
Entertainment
Sports
Life
McNair's final hours revealed
STORY
as losing control. » Details
• UConn murder victim mourned
Find Steve McNair murder case
F1
F2
F3
F4
» More: Featured | Buzz

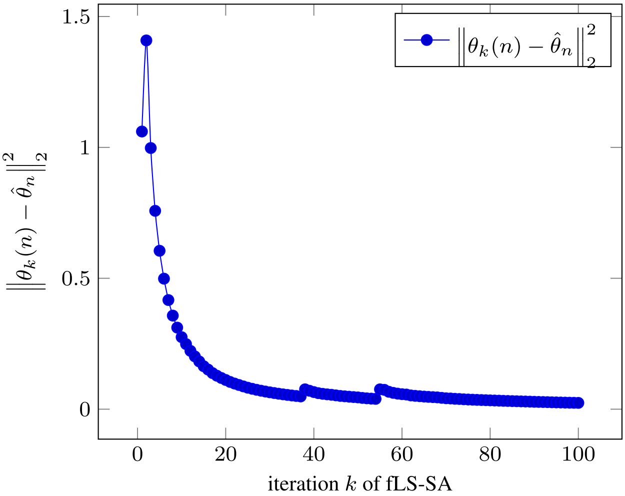


Fig. 7 Distance between fLS-SA iterate $\theta_k(n)$ and $\hat{\theta}_n$ in iteration $n = 165$ of fLinUCB-SA, with day 2's data file as input

choice 165 for the iteration is arbitrary, as we observed similar behavior 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.

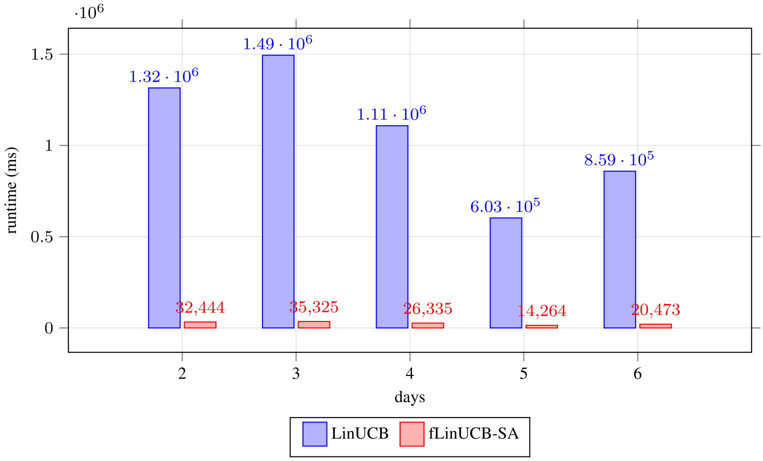


Fig. 8 Performance comparison of the algorithms using runtimes on various days of the dataset

12 Conclusions and future work

We analyzed the TD algorithm with linear function approximation, under uniform sampling from a dataset. We provided convergence rate results for this algorithm, both in high probability and in expectation. Furthermore, we also established that using our batchTD scheme in place of LSTD does not impact the rate of convergence of the approximate value function to the true value function. These results coupled with the fact that the batchTD algorithm 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 batchTDQ in place of LSTDQ for policy evaluation. We also extended our analysis to bound 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.

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. It would be an interesting future research direction to use the proof technique employed to analyze batchTD, and incorporate the necessary deviations to handle the more general Markov noise.

We outline a few future research directions for improving batchTD algorithm developed here: (i) develop extensions of batchTD to approximate LSTD(λ); (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 batchTD with cyclic sampling as an incremental algorithm in the spirit of TD; and (iii) leverage recent enhancements to SGD in the context of least squares regression, cf. Dieuleveut et al. (2016). An orthogonal direction of future research is to develop online algorithms that track the corresponding batch solutions, efficiently and this has been partially accomplished by Korda et al. (2015), and Tarrès and Yao (2011).

Appendix 1: Proof of Theorem 10.3

The proof of Theorem 10.3 relies on a general rate result built from Proposition 10.1

Proposition 13.1 *Under (A1)–(A3) we have, for all $\epsilon \geq 0$ and $\forall n \geq 1$,*

$$\mathbb{P}(\|z_n\|_2 - \mathbb{E}\|z_n\|_2 \geq \epsilon) \leq \exp\left(-\frac{\epsilon^2}{4h(n)^2 \sum\limits_{m=1}^{n} L_m^2}\right),$$

where $L_i \triangleq \frac{\gamma_i}{n}\left(\sum_{l=i+1}^{n-1} \prod\limits_{j=i}^{l} \left(1 - \mu\gamma_{j+1}(2 - \Phi_{\max}^2 \gamma_{j+1}))\right)^{1/2}\right)$, and h(n) is as in Proposition 10.1.

Proof This proof follows exactly the proof of Proposition 8.3, except that it uses the form of L_i for non-averaged iterates as derived in Proposition 10.1 part (1), rather than as derived in Proposition 8.1 part (1). □

We specialise this result with the choice of step size $\gamma_n \triangleq (c_0 c^\alpha)/(n+c)^\alpha$. First, we prove the form of the L_i constants for this choice of step size in the lemma below.

Lemma 13.1 *Under conditions of Theorem 10.3, we have*

$$\sum_{i=1}^{n} L_i^2 \leq \frac{1}{\mu^2}\left\{2^\alpha + \left[\left[\frac{2\alpha}{c_0 \mu c^\alpha}\right]^{\frac{1}{1-\alpha}} + \frac{2(1-\alpha)(c_0\mu)^\alpha}{\alpha}\right]\right\}^2 \frac{1}{n}.$$

Second, we bound the expected error by directly averaging the errors of the non-averaged iterates:

$$\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left\|\theta_k - \hat{\theta}_T\right\|_2, \tag{106}$$

and directly applying the bounds in expectation given in Proposition 8.1.

Lemma 13.2 *Under conditions of Theorem 10.3, we have*

$$\begin{aligned}\mathbb{E}\left\|\bar{\theta}_n - \hat{\theta}_T\right\|_2 \leq & C_0\left(C_1\|\theta_0 - \theta_T\|_2 + 2h(n)c^\alpha c_0\left(2c_0\mu c^\alpha\right)^{\frac{\alpha}{(1-\alpha)}}\sqrt{e}\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{2(1-\alpha)}}\right)\frac{1}{n}\\ & + h(n)c^\alpha c_0\left(2c_0\mu c^\alpha\right)^{\frac{\alpha}{2(1-\alpha)}}(n+c)^{-\frac{\alpha}{2}},\end{aligned}$$

where C_0 and C_1 are as defined in Theorem 10.3.

Proof of Lemma 13.1

Recall from the statement of Theorem 10.3 that

$$0 < c_0 \boldsymbol{\Phi}_{\max}^2 < 1. \tag{107}$$

Recall also from the formula in Proposition 13.1, that

$$L_i = \frac{\gamma_i}{n}\left(\sum_{l=i+1}^{n-1} \prod_{j=i}^{l} \left(1 - \mu\gamma_{j+1}(2 - \boldsymbol{\Phi}_{\max}^2 \gamma_{j+1}))\right)^{1/2}\right).$$

Notice that

$$\begin{aligned}\sum_{i=1}^{n} L_i^2 &= \sum_{i=1}^{n}\left[\frac{\gamma_i}{n}\left(\sum_{l=i+1}^{n-1} \prod_{j=i}^{l} \left(1 - \mu\gamma_{j+1}(2 - \boldsymbol{\Phi}_{\max}^2 \gamma_{j+1}))\right)^{1/2}\right)\right]^2 \\ &\leq \frac{1}{n^2}\sum_{i=1}^{n}\left[\gamma_i\left(\sum_{l=i+1}^{n-1} \exp\left(-\sum_{j=i}^{l} \mu\gamma_{j+1}(2 - \boldsymbol{\Phi}_{\max}^2 \gamma_{j+1}))\right)\right)\right]^2 \\ &< \frac{1}{n^2}\sum_{i=1}^{n}\underbrace{\left[c_0\left(\frac{c}{c+i}\right)^{\alpha}\left(\sum_{l=i+1}^{n-1} \exp\left(-c_0\mu\sum_{j=i}^{l}\left(\frac{c}{c+j}\right)^{\alpha}\right)\right)\right]^2}_{\triangleq(A)}.\end{aligned}$$

To produce the final bound, we bound the summand (A) highlighted in line (91) by a constant, uniformly over all values of i and n, exactly as in the proof of Lemma 8.1. Thus, we have

$$\sum_{i=1}^{n} L_i^2 \leq \frac{1}{\mu^2}\left\{2^{\alpha} + \left[\left[\frac{2\alpha}{c_0\mu c^{\alpha}}\right]^{\frac{1}{1-\alpha}} + \frac{2(1-\alpha)(c_0\mu)^{\alpha}}{\alpha}\right]\right\}^2 \frac{1}{n}.$$

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

Proof of Lemma 13.2

Recall that $\gamma_n \triangleq c_0\left(\frac{c}{(c+n)}\right)^{\alpha}$. Recall that in Theorem 10.3 we have assumed that

$$0 < c_0 \boldsymbol{\Phi}_{\max}^2 < 1. \tag{108}$$

Using (99), we have

$$\begin{aligned}
&\mathbb{E}\left(\left\|\theta_n - \hat{\theta}_T\right\|_2\right)^2 \\
&\le \left[\prod_{k=1}^{n}\left(1-\mu\gamma_k(2-\gamma_k\boldsymbol{\Phi}^2_{\max}\right)\|z_0\|_2\right]^2 + 4\sum_{k=1}^{n}\gamma_k^2\left[\prod_{j=k}^{n-1}(1-\mu\gamma_j(2-\gamma_j\boldsymbol{\Phi}^2_{\max})\right]^2 h(k)^2 \\
&\le \left[\prod_{k=1}^{n}\left(1-\frac{\mu c_0 c^\alpha}{(c+k)^\alpha}\right)\|z_0\|_2\right]^2 + 4\sum_{k=1}^{n}\frac{c_0^2c^{2\alpha}}{(c+k)^{2\alpha}}\left[\prod_{j=k}^{n-1}\left(1-\frac{\mu c_0 c^\alpha}{(c+j)^\alpha}\right)\right]^2 h(k)^2 \\
&\le \left[\exp\left(-\mu c_0\sum_{k=1}^{n}\frac{c^\alpha}{(c+k)^\alpha}\right)\|z_0\|_2\right]^2 + 4h(n)^2\sum_{k=1}^{n}\frac{c_0^2c^{2\alpha}}{(c+k)^{2\alpha}}\exp\left(-2\mu c_0\sum_{j=k}^{n-1}\frac{c^\alpha}{(c+j)^\alpha}\right).
\end{aligned} \tag{109}$$

To obtain (109), we have applied (108). For the final inequality, we have exponentiated the logarithm of the products, and used the inequality $\ln(1+x) < x$ in several places.

Continuing the derivation, we have

$$\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 \tag{110}$$

$$\begin{aligned}
&\le \exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha} - c_0\mu c^\alpha(1+c)^{1-\alpha}\right)\left\|\theta_0-\hat{\theta}_T\right\|_2 \\
&\quad + 2h(n)\left(\sum_{k=1}^{n}c_0^2\left(\frac{c}{k+c}\right)^{2\alpha}\exp\left(-2c_0\mu c^\alpha((n+c)^{1-\alpha}-(k+c)^{1-\alpha}\right)\right)^{\frac{1}{2}}
\end{aligned} \tag{111}$$

$$\begin{aligned}
&= \exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right) \\
&\quad\times\left[\exp\left(c_0\mu c^\alpha(1+c)^{1-\alpha}\right)\left\|\theta_0-\hat{\theta}_T\right\|_2\right. \\
&\quad\left. + 2h(n)\left\{\sum_{k=1}^{n}c_0^2\left(\frac{c}{k+c}\right)^{2\alpha}\exp\left(2c_0\mu c^\alpha((k+c)^{1-\alpha}\right)\right\}^{\frac{1}{2}}\right] \\
&\le \exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right) \\
&\quad\times\left[\exp\left(c_0\mu c^\alpha(1+c)^{1-\alpha}\right)\left\|\theta_0-\hat{\theta}_T\right\|_2\right. \\
&\quad\left. + 2h(n)\left\{c^{2\alpha}c_0^2\int_1^{n+c}x^{-2\alpha}\exp\left(2c_0\mu c^\alpha x^{1-\alpha}\right)dx\right\}^{\frac{1}{2}}\right]
\end{aligned} \tag{112}$$

$$\begin{aligned}
&\le \exp\left(-c_0\mu c^\alpha(n+c)^{1-\alpha}\right) \\
&\quad\times\left[\exp\left(c_0\mu c^\alpha(1+c)^{1-\alpha}\right)\left\|\theta_0-\hat{\theta}_T\right\|_2\right. \\
&\quad\left. + 2h(n)\left\{c^{2\alpha}c_0^2\left(2c_0\mu c^\alpha\right)^{\frac{2\alpha}{1-\alpha}}\times\int_{\left(2c_0\mu c^\alpha\right)^{1/(1-\alpha)}}^{(n+c)\left(2c_0\mu c^\alpha\right)^{1/(1-\alpha)}}y^{-2\alpha}\exp(y^{1-\alpha})dy\right\}^{\frac{1}{2}}\right]
\end{aligned} \tag{113}$$

As in the proof of Theorem 5.1, for arriving at (111), we have used Jensen's inequality, and that $\sum_{j=k}^{n-1}(c+j)^{-\alpha} \geq \int_{j=k}^{n}(c+j)^{1-\alpha}dj = (c+n)^{1-\alpha} - (c+k)^{1-\alpha}$. To obtain (112), we have upper bounded the sum with an integral, the validity of which follows from the observation that $x \mapsto x^{-2\alpha}e^{x^{1-\alpha}}$ is convex for $x \geq 1$. Finally, for (113), we have applied the change of variables $y = (2c_0\mu c^{\alpha})^{1/(1-\alpha)}x$.

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

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

and furthermore, since $y \mapsto y^{-2\alpha}\exp(y^{1-\alpha})$ is decreasing for $y \leq \left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}$, we have

$$\int_1^{\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}} y^{-2\alpha}\exp(y^{1-\alpha})dy \leq e\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{1-\alpha}}.$$

Plugging these into (113), we obtain

$$\begin{aligned}
\mathbb{E}\left\|\theta_n - \hat{\theta}_T\right\|_2 &\leq \exp\left(-c_0\mu c^{\alpha}(n+c)^{1-\alpha}\right) \\
&\times\left(\exp\left(c_0\mu c^{\alpha}(1+c)^{1-\alpha}\right)\left\|\theta_0 - \theta_T\right\|_2 + 2h(n)c^{\alpha}c_0\left(2c_0\mu c^{\alpha}\right)^{\frac{\alpha}{(1-\alpha)}}\sqrt{e}\left(\frac{2\alpha}{1-\alpha}\right)^{\frac{1}{2(1-\alpha)}}\right) \\
&+ 2h(n)c^{\alpha}c_0\left(2c_0\mu c^{\alpha}\right)^{\frac{\alpha}{2(1-\alpha)}}(n+c)^{-\frac{\alpha}{2}}.
\end{aligned}$$

Hence, we obtain

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

□

References

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.

Bach, F., & Moulines, E. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in neural information processing systems* (pp. 451–459).

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).

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control, Approximate Dynamic Programming*, (4th ed., Vol. II). Belmont: Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3)*, (Vol. 7). Belmont: Athena Scientific.

Bhandari, J., Russo, D., & Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory pp. 1691–1692.*

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

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

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

Dalal, G., Szörényi, B., Thoppe, G., & Mannor, S. (2018). Finite sample analyses for td (0) with function approximation. In *Thirty-second AAAI conference on artificial intelligence*.

Dani, V., Hayes, T. P., & Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st annual conference on learning theory (COLT)* (pp. 355–366).

Dieuleveut, A., Flammarion, N., & Bach, F. (2016). Harder, better, faster, stronger convergence rates for least-squares regression. arXiv preprint arXiv:160205419.

Fathi, M., & Frikha, N. (2013). Transport-entropy inequalities and deviation estimates for stochastic approximation schemes. arXiv preprint arXiv:13017740.

Frikha, N., & Menozzi, S. (2012). Concentration bounds for stochastic approximations. *Electronic Communications in Probability*, *17*(47), 1–15.

Geramifard, A., Bowling, M., Zinkevich, M., & Sutton, R. S. (2007). iLSTD: Eligibility traces and convergence analysis. In *NIPS* (Vol. 19, p. 441).

Hazan, E., & Kale, S. (2011). Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *COLT* (pp. 421–436).

Konda, V. R. (2002). Actor-critic algorithms. PhD thesis, Department of Electrical Engineering and Computer Science, MIT.

Korda, N., Prashanth, L. A., & 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).

Kushner, H., & Clark, D. (1978). *Stochastic approximation methods for constrained and unconstrained systems*. Berlin: Springer-Verlag.

Kushner, H. J., & Yin, G. (2003). *Stochastic approximation and recursive algorithms and applications*, (Vol. 35). Berlin: Springer Verlag.

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

Lakshminarayanan, C., & Szepesvari, C. (2018). Linear stochastic approximation: How far does constant step-size and iterate averaging go? *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, *84*, 1347–1355.

Lazaric, A., Ghavamzadeh, M., & Munos, R. (2012). Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, *13*, 3041–3074.

Li, L., Chu, W., Langford, J., & Schapire, R. E. (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).

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).

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., & Petrik, M. (2015). Finite-sample analysis of proximal gradient TD algorithms. In: *Proceedings of the 31st conference on uncertainty in artificial intelligence, Amsterdam, Netherlands*

Mary, J., Garivier, A., Li, L., Munos, R., Nicol, O., Ortner, R., & Preux, P. (2012). ICML exploration and exploitation 3—new challenges. http://explochallenge.inria.fr.

Narayanan, C., & Szepesvári, C. (2017). Finite time bounds for temporal difference learning with function approximation: Problems with some "state-of-the-art" results. Technical report, https://sites.ualberta.ca/~szepesva/papers/TD-issues17.pdf.

Nemirovsky, A., & Yudin, D. (1983). Problem complexity and method efficiency in optimization. NY: Wiley-Interscience.

Pires, BA., & Szepesvári, C. (2012). Statistical linear estimation with penalized estimators: An application to reinforcement learning. arXiv preprint arXiv:12066444.

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

Prashanth, L. A., & Bhatnagar, S. (2011). Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 412–421.

Prashanth, L. A., & Bhatnagar, S. (2012). Threshold tuning using stochastic optimization for graded signal control. *IEEE Transactions on Vehicular Technology*, *61*(9), 3865–3880.

Prashanth, L. A., Korda, N., & Munos, R. (2014). Fast LSTD using stochastic approximation: Finite time analysis and application to traffic control. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 66–81).

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*, 400–407

Roux, N. L., Schmidt, M., & Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in neural information processing systems* (pp. 2663–2671).

Ruppert, D. (1991). Stochastic approximation. In B. K. Ghosh & P. K. Sen (Eds.), *Handbook of sequential analysis* (pp. 503–529).

Silver, D., Sutton, R. S., & Müller, M. (2007). Reinforcement learning of local shape in the game of go. *IJCAI*, *7*, 1053–1058.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge: Cambridge University Press.

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

Sutton, R. S., et al.(2009b). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML ACM* (pp. 993–1000).

Tagorti, M., & Scherrer, B. (2015). On the Rate of Convergence and Error Bounds for LSTD(λ). In *ICML*.

Tarrès, P., & Yao, Y. (2011). Online learning as stochastic approximation of regularization paths. arXiv preprint arXiv:11035538.

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

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint* (Vol. 48). Cambridge: Cambridge University Press.

Webscope, Y. (2011). Yahoo! Webscope dataset ydata-frontpage-todaymodule-clicks-v2_0. http://research.yahoo.com/Academic_Relations.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *COLT* (pp. 1724–1751).

Yu, H., & Bertsekas, D. P. (2009). Convergence results for some temporal difference methods based on least squares. *IEEE Transactions on Automatic Control*, *54*(7), 1515–1531.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *ICML* (pp. 928–925).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

L. A. Prashanth[1] · Nathaniel Korda[2] · Rémi Munos[3]

Nathaniel Korda
nathaniel.korda@eng.ox.ac.uk

Rémi Munos
remi.munos@gmail.com

[1] Indian Institute of Technology Madras, Chennai, India

[2] Oxford University, Oxford, UK

[3] Google Deepmind, Paris, France