Proximal Gradient Temporal Difference Learning Algorithms

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, Marek Petrik
UMass Amherst, U. of Rochester, Adobe & INRIA Lille, UMass Amherst, IBM Research
{boliu, mahadeva}@cs.umass.edu, jliu@cs.rochester.edu,
Mohammad.ghavamzadeh@inria.fr, marekpetrik@gmail.com

Abstract

In this paper, we describe proximal gradient temporal difference learning, which provides a principled way for designing and analyzing true stochastic gradient temporal difference learning algorithms. We show how gradient TD (GTD) reinforcement learning methods can be formally derived, not with respect to their original objective functions as previously attempted, but rather with respect to primal-dual saddle-point objective functions. We also conduct a saddle-point error analysis to obtain finite-sample bounds on their performance. Previous analyses of this class of algorithms use stochastic approximation techniques to prove asymptotic convergence, and no finite-sample analysis had been attempted. An accelerated algorithm is also proposed, namely GTD2-MP, which use proximal "mirror maps" to yield acceleration. The results of our theoretical analysis imply that the GTD family of algorithms are comparable and may indeed be preferred over existing least squares TD methods for off-policy learning, due to their linear complexity. We provide experimental results showing the improved performance of our accelerated gradient TD methods.

1 Introduction

Designing a true stochastic gradient unconditionally stable temporal difference (TD) method with finite-sample convergence analysis has been a longstanding goal of reinforcement learning (RL) [Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998]. It was discovered more than two decades ago that the original TD method was unstable in many off-policy scenarios where the target behavior being learned and the exploratory behavior producing samples differ by Baird and others [Baird, 1995]. Sutton *et al.* [2008, 2009] proposed the family of gradient-based temporal difference (GTD) algorithms to address the limitations of the standard TD algorithm. A key property of this class of GTD algorithms is that they are asymptotically off-policy convergent, which was shown using the framework of stochastic approximation [Borkar, 2008]. Many RL algorithms, especially those that are based on stochastic approximation, such as TD(λ), do not have convergence guarantees in the off-policy setting. Unfortunately, this class of GTD algorithms are *not true stochastic gradient methods with respect to their original objective functions*, as pointed out in Szepesvári [2010]. The reason is not surprising: the gradient of the objective functions used involve products of terms, which cannot be sampled directly, and was decomposed by an ad-hoc splitting of terms. In this paper, we show a principled way of designing true stochastic gradient TD algorithms by using a primal-dual saddle point objective function, derived from the original objective functions, coupled with the principled use of *operator splitting* [Bauschke and Combettes, 2011].

2 Preliminaries

Reinforcement Learning (RL) [Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998] is a class of learning problems in which an agent interacts with an unfamiliar, dynamic and stochastic environment, where the agent's goal is to optimize some measure of its long-term performance. This interaction is conventionally modeled as a Markov decision process (MDP). A MDP is defined as the tuple $(\mathcal{S}, \mathcal{A}, P^a_{ss'}, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the sets of states and actions, the transition kernel $P^a_{ss'}$ specifying the probability of transition from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ by taking action $a \in \mathcal{A}$, $R(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function bounded by $R_{\max}$., and $0 \leq \gamma < 1$ is a discount factor. A stationary policy $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ is a probabilistic mapping from states to actions. The main objective of a RL algorithm is to find an optimal policy. In order to achieve this goal, a key step in many algorithms is to calculate the value function of a given policy π, i.e., $V^\pi : \mathcal{S} \to \mathbb{R}$, a process known as *policy evaluation*. It is known that V^π is the unique fixed-point of the *Bellman operator* T^π, i.e.,

$$V^\pi = T^\pi V^\pi = R^\pi + \gamma P^\pi V^\pi, \tag{1}$$

where R^π and P^π are the reward function and transition kernel of the Markov chain induced by policy π. In Eq. 1, we may imagine V^π as a $|\mathcal{S}|$-dimensional vector and write everything in vector/matrix form. In the following, to simplify the notation, we often drop the dependence of T^π, V^π, R^π, and P^π to π.

We denote by π_b, the behavior policy that generates the data, and by π, the target policy that we would like to evalu-

ate. They are the same in the on-policy setting and different in the off-policy scenario. For each state-action pair (s_i, a_i), such that $\pi_b(a_i|s_i) > 0$, we define the importance-weighting factor $\rho_i = \pi(a_i|s_i)/\pi_b(a_i|s_i)$ with $\rho_{\max} \geq 0$ being its maximum value over the state-action pairs.

When $\mathcal{S}$ is large or infinite, we often use a linear approximation architecture for V^π with parameters $\theta \in \mathbb{R}^d$ and L-bounded basis functions $\{\varphi_i\}_{i=1}^d$, i.e., $\varphi_i : \mathcal{S} \to \mathbb{R}$ and $\max_i ||\varphi_i||_\infty \leq L$. We denote by $\phi(\cdot) = \big(\varphi_1(\cdot), \ldots, \varphi_d(\cdot)\big)^\top$ the feature vector and by $\mathcal{F}$ the linear function space spanned by the basis functions $\{\varphi_i\}_{i=1}^d$, i.e., $\mathcal{F} = \big\{f_\theta \mid \theta \in \mathbb{R}^d \text{ and } f_\theta(\cdot) = \phi(\cdot)^\top\theta\big\}$. We may write the approximation of V in $\mathcal{F}$ in the vector form as $\hat{v} = \Phi\theta$, where Φ is the $|\mathcal{S}| \times d$ feature matrix. When only n training samples of the form $\mathcal{D} = \big\{\big(s_i, a_i, r_i = r(s_i, a_i), s_i'\big)\big\}_{i=1}^n$, $s_i \sim \xi$, $a_i \sim \pi_b(\cdot|s_i)$, $s_i' \sim P(\cdot|s_i, a_i)$, are available ($\xi$ is a distribution over the state space $\mathcal{S}$), we may write the *empirical Bellman operator* $\hat{T}$ for a function in $\mathcal{F}$ as

$$\hat{T}(\hat{\Phi}\theta) = \hat{R} + \gamma\hat{\Phi}'\theta, \tag{2}$$

where $\hat{\Phi}$ (resp. $\hat{\Phi}'$) is the empirical feature matrix of size $n \times d$, whose i-th row is the feature vector $\phi(s_i)^\top$ (resp. $\phi(s_i')^\top$), and $\hat{R} \in \mathbb{R}^n$ is the reward vector, whose i-th element is r_i. We denote by $\delta_i(\theta) = r_i + \gamma\phi_i'^\top\theta - \phi_i^\top\theta$, the TD error for the i-th sample (s_i, r_i, s_i') and define $\Delta\phi_i = \phi_i - \gamma\phi_i'$. Finally, we define the matrices A and C, and the vector b as

$$A := \mathbb{E}\big[\rho_i\phi_i(\Delta\phi_i)^\top\big], \quad b := \mathbb{E}\left[\rho_i\phi_i r_i\right], \quad C := \mathbb{E}[\phi_i\phi_i^\top], \tag{3}$$

where the expectations are w.r.t. ξ and P^{π_b}. We also denote by Ξ, the diagonal matrix whose elements are $\xi(s)$, and $\xi_{\max} := \max_s \xi(s)$. For each sample i in the training set $\mathcal{D}$, we can calculate an unbiased estimate of A, b, and C as follows:

$$\hat{A}_i := \rho_i\phi_i\Delta\phi_i^\top, \quad \hat{b}_i := \rho_i r_i\phi_i, \quad \hat{C}_i := \phi_i\phi_i^\top. \tag{4}$$

2.1 GRADIENT-BASED TD ALGORITHMS

The class of gradient-based TD (GTD) algorithms were proposed by Sutton *et al.* [2008, 2009]. These algorithms target two objective functions: the *norm of the expected TD update* (NEU) and the *mean-square projected Bellman error* (MSPBE), defined as (see e.g., Maei 2011)[1]

$$\text{NEU}(\theta) = ||\Phi^\top\Xi(T\hat{v} - \hat{v})||^2, \tag{5}$$

$$\text{MSPBE}(\theta) = ||\hat{v} - \Pi T\hat{v}||_\xi^2 = ||\Phi^\top\Xi(T\hat{v} - \hat{v})||_{C^{-1}}^2, \tag{6}$$

where $C = \mathbb{E}[\phi_i\phi_i^\top] = \Phi^\top\Xi\Phi$ is the covariance matrix defined in Eq. 3 and is assumed to be non-singular, and $\Pi = \Phi(\Phi^\top\Xi\Phi)^{-1}\Phi^\top\Xi$ is the orthogonal projection operator into the function space $\mathcal{F}$, i.e., for any bounded function g, $\Pi g = \arg\min_{f\in\mathcal{F}} ||g - f||_\xi$. From (5) and (6), it is clear that NEU and MSPBE are square unweighted and weighted by C^{-1}, ℓ_2-norms of the quantity $\Phi^\top\Xi(T\hat{v} - \hat{v})$, respectively, and thus, the two objective functions can be unified as

$$J(\theta) = ||\Phi^\top\Xi(T\hat{v} - \hat{v})||_{M^{-1}}^2 = ||\mathbb{E}[\rho_i\delta_i(\theta)\phi_i]||_{M^{-1}}^2, \tag{7}$$

[1]It is important to note that T in (5) and (6) is T^π, the Bellman operator of the target policy π.

with M equals to the identity matrix I for NEU and to the covariance matrix C for MSPBE. The second equality in (7) holds because of the following lemma from Section 4.2 in Maei [2011].

Lemma 1. *Let $\mathcal{D} = \big\{\big(s_i, a_i, r_i, s_i'\big)\big\}_{i=1}^n$, $s_i \sim \xi$, $a_i \sim \pi_b(\cdot|s_i)$, $s_i' \sim P(\cdot|s_i, a_i)$ be a training set generated by the behavior policy π_b and T be the Bellman operator of the target policy π. Then, we have*

$$\Phi^\top\Xi(T\hat{v} - \hat{v}) = \mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i\big] = b - A\theta.$$

Motivated by minimizing the NEU and MSPBE objective functions using the stochastic gradient methods, the GTD and GTD2 algorithms were proposed with the following update rules:

$$\textbf{GTD:} \quad \begin{aligned} y_{t+1} &= y_t + \alpha_t\big(\rho_t\delta_t(\theta_t)\phi_t - y_t\big), \\ \theta_{t+1} &= \theta_t + \alpha_t\rho_t\Delta\phi_t(y_t^\top\phi_t), \end{aligned} \tag{8}$$

$$\textbf{GTD2:} \quad \begin{aligned} y_{t+1} &= y_t + \alpha_t\big(\rho_t\delta_t(\theta_t) - \phi_t^\top y_t\big)\phi_t, \\ \theta_{t+1} &= \theta_t + \alpha_t\rho_t\Delta\phi_t(y_t^\top\phi_t). \end{aligned} \tag{9}$$

However, it has been shown that the above update rules do not update the value function parameter θ in the gradient direction of NEU and MSPBE, and thus, NEU and MSPBE are not the true objective functions of the GTD and GTD2 algorithms [Szepesvári, 2010]. Consider the NEU objective function in (5). Taking its gradient w.r.t. θ, we obtain

$$\begin{aligned} -\frac{1}{2}\nabla\text{NEU}(\theta) &= -\big(\nabla\mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i^\top\big]\big)\mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i\big] \\ &= -\big(\mathbb{E}\big[\rho_i\nabla\delta_i(\theta)\phi_i^\top\big]\big)\mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i\big] \\ &= \mathbb{E}\big[\rho_i\Delta\phi_i\phi_i^\top\big]\mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i\big]. \end{aligned} \tag{10}$$

If the gradient can be written as a single expectation, then it is straightforward to use a stochastic gradient method. However, we have a product of two expectations in (10), and unfortunately, due to the correlation between them, the sample product (with a single sample) won't be an unbiased estimate of the gradient. To tackle this, the GTD algorithm uses an auxiliary variable y_t to estimate $\mathbb{E}\big[\rho_i\delta_i(\theta)\phi_i\big]$, and thus, the overall algorithm is no longer a true stochastic gradient method w.r.t. NEU. It can be easily shown that the same problem exists for GTD2 w.r.t. the MSPBE objective function. This prevents us from using the standard convergence analysis techniques of stochastic gradient descent methods to obtain a finite-sample performance bound for the GTD and GTD2 algorithms.

It should be also noted that in the original publications of GTD/GTD2 algorithms [Sutton *et al.*, 2008, 2009], the authors discussed handling the off-policy scenario using both importance and rejected sampling. In rejected sampling that was mainly used in Sutton *et al.* [2008, 2009], a sample (s_i, a_i, r_i, s_i') is rejected and the parameter θ does not update for this sample, if $\pi(a_i|s_i) = 0$. This sampling strategy is not efficient since a lot of samples will be discarded if π_b and π are very different.

2.2 Related Work

The line of research reported here began with the development of a broad framework called proximal reinforcement learning [Mahadevan *et al.*, 2014], which explores first-order reinforcement learning algorithms using *mirror maps* [Bubeck, 2014; Juditsky *et al.*, 2008] to construct primal-dual spaces. This framework led to a dual space formulation of first-order sparse TD learning [Mahadevan and Liu, 2012]. A saddle point formulation for off-policy TD learning was initially explored in Liu *et al.* [2012] and later in Valcarcel Macua *et al.* [2015], where the objective function is the norm of the approximation residual of a linear inverse problem [Ávila Pires and Szepesvári, 2012]. A sparse off-policy GTD2 algorithm with regularized dual averaging is introduced by Qin and Li [2014]. These studies provide different approaches to formulating the problem, first as a variational inequality problem [Juditsky *et al.*, 2008; Mahadevan *et al.*, 2014] or as a linear inverse problem [Liu *et al.*, 2012], or as a quadratic objective function (MSPBE) using two-time-scale solvers [Qin and Li, 2014]. In this paper, we are going to explore the true nature of the GTD algorithms as stochastic gradient algorithm w.r.t the convex-concave saddle-point formulations of NEU and MSPBE.

3 Algorithm Analysis

3.1 Saddle-Point Formulation of GTD Algorithms

In this section, we show how the GTD and GTD2 algorithms can be formulated as true stochastic gradient (SG) algorithms by writing their respective objective functions, NEU and MSPBE, in the form of a convex-concave saddle-point. As discussed earlier, this new formulation of GTD and GTD2 as true SG methods allows us to use the convergence analysis techniques for SG methods to derive finite-sample performance bounds for these RL algorithms. Moreover, it allows us to use more efficient algorithms that have been recently developed to solve SG problems, such as *stochastic Mirror-Prox* (SMP) [Juditsky *et al.*, 2008], to derive more efficient versions of GTD and GTD2.

In this paper, we consider the saddle-point problem formulation as follows,

$$\min_{\theta}\max_{y}\Big(L(\theta,y)=\langle b-A\theta,y\rangle-\frac{1}{2}||y||_M^2\Big), \qquad (11)$$

where A and b were defined by Eq. 3, and M is a positive definite matrix.

We first show in Proposition 1 that if (θ^*, y^*) is the saddle-point of problem (11), then θ^* will be the optimum of NEU and MSPBE defined in Eq. 7. We then prove in Proposition 2 that GTD and GTD2 in fact find this saddle-point.

Proposition 1. *For any fixed θ, we have $\frac{1}{2}J(\theta) = \max_y L(\theta, y)$, where $J(\theta)$ is defined by Eq. 7.*

Proof. Since $L(\theta, y)$ is an unconstrained quadratic program w.r.t. y, the optimal $y^*(\theta) = \arg\max_y L(\theta, y)$ can be analytically computed as

$$y^*(\theta) = M^{-1}(b - A\theta). \qquad (12)$$

The result follows by substituting y^* into (11) and using the definition of $J(\theta)$ in Eq. 7 and Lemma 1. □

Proposition 2. *GTD and GTD2 are true stochastic gradient algorithms w.r.t. the objective function $L(\theta, y)$ of the saddle-point problem* (11) *with $M = I$ and $M = C = \Phi^\top \Xi \Phi$ (the covariance matrix), respectively.*

Proof. It is easy to see that the gradient updates of the saddle-point problem (11) (ascending in y and descending in θ) may be written as

$$\begin{aligned} y_{t+1} &= y_t + \alpha_t\left(b - A\theta_t - My_t\right), \qquad (13)\\ \theta_{t+1} &= \theta_t + \alpha_t A^\top y_t. \end{aligned}$$

We denote $\hat{M} := 1$ (resp. $\hat{M} := \hat{C}$) for GTD (resp. GTD2). We may obtain the update rules of GTD and GTD2 by replacing A, b, and C in (13) with their unbiased estimates $\hat{A}$, $\hat{b}$, and $\hat{C}$ from Eq. 4, which completes the proof. □

3.2 Accelerated Algorithm

As discussed at the beginning of Section 3.1, this saddle-point formulation not only gives us the opportunity to use the techniques for the analysis of SG methods to derive finite-sample performance bounds for the GTD algorithms, but also it allows us to use the powerful algorithms that have been recently developed to solve the SG problems and derive more efficient versions of GTD and GTD2. Stochastic Mirror-Prox (SMP) [Juditsky *et al.*, 2008] is an "almost dimension-free" non-Euclidean extra-gradient method that deals with both smooth and non-smooth stochastic optimization problems (see Juditsky and Nemirovski 2011 and Bubeck 2014 for more details). Using SMP, we propose a new version of GTD/GTD2, called GTD-MP/GTD2-MP, with the following update formula:[2]

$$y_t^m = y_t + \alpha_t(\hat{b}_t - \hat{A}_t\theta_t - \hat{M}_t y_t), \qquad \theta_t^m = \theta_t + \alpha_t\hat{A}_t^\top y_t,$$

$$y_{t+1} = y_t + \alpha_t(\hat{b}_t - \hat{A}_t\theta_t^m - \hat{M}_t y_t^m),\ \theta_{t+1} = \theta_t + \alpha_t\hat{A}_t^\top y_t^m.$$

After T iterations, these algorithms return $\bar{\theta}_T := \frac{\sum_{t=1}^T \alpha_t\theta_t}{\sum_{t=1}^T \alpha_t}$ and $\bar{y}_T := \frac{\sum_{t=1}^T \alpha_t y_t}{\sum_{t=1}^T \alpha_t}$. The details of the algorithm is shown in Algorithm 1, and the experimental comparison study between GTD2 and GTD2-MP is reported in Section 4.

3.3 Finite-Sample Analysis

In this section, we are going to discuss the convergence rate of the accelerated algorithms using off-the-shelf accelerated solvers for saddle-point problems. Due to space limitations, many details are relegated to a longer paper [Liu *et al.*, 2015], where both error bounds and performance bounds are provided, which shows that the value function approximation bound of the GTD algorithms family is $O\left(\frac{d}{n^{1/4}}\right)$. For simplicity, we will discuss the error bound of $\frac{1}{2}||A\theta - b||^2_{M^{-1}}$, and the corresponding error bound of $\frac{1}{2}||A\theta - b||^2_\xi$ and $\|V - \bar{v}_n\|_\xi$ can be likewise derived, We show the convergence rate of the GTD algorithms family is

$$(\mathbf{GTD/GTD2}): \quad O\left(\frac{\tau + ||A||_2 + \sigma}{\sqrt{n}}\right) \qquad (15)$$

[2] For simplicity, we only describe mirror-prox GTD methods where the mirror map is identity, which can also be viewed as extra-gradient (EG) GTD methods. Mahadevan *et al.* [2014] gives a more detailed discussion of a broad range of mirror maps in RL.

Algorithm 1 GTD2-MP

1: **for** $t = 1, \ldots, n$ **do**
2: Update parameters

$$\delta_t = r_t - \theta_t^\top \Delta\phi_t$$
$$y_t^m = y_t + \alpha_t(\rho_t\delta_t - \phi_t^\top y_t)\phi_t$$
$$\theta_t^m = \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t)$$
$$\delta_t^m = r_t - (\theta_t^m)^\top\Delta\phi_t$$
$$y_{t+1} = y_t + \alpha_t(\rho_t\delta_t^m - \phi_t^\top y_t^m)\phi_t$$
$$\theta_{t+1} = \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t^m)$$

3: **end for**
4: OUTPUT

$$\bar{\theta}_n := \frac{\sum_{t=1}^n \alpha_t\theta_t}{\sum_{t=1}^n \alpha_t} \quad , \quad \bar{y}_n := \frac{\sum_{t=1}^n \alpha_t y_t}{\sum_{t=1}^n \alpha_t} \tag{14}$$

We also discuss the interesting question: what is the "optimal" GTD algorithm? To answer this question, we review the convex-concave formulation of GTD2. According to convex programming complexity theory [Juditsky *et al.*, 2008], the un-improvable convergence rate of stochastic saddle-point problem (11) is

$$(\textbf{Optimal}): \quad O\left(\frac{\tau}{n^2} + \frac{||A||_2}{n} + \frac{\sigma}{\sqrt{n}}\right) \tag{16}$$

There are many readily available stochastic saddle-point solvers, such as stochastic Mirror-Prox (SMP) [Juditsky *et al.*, 2008] algorithm, which leads to our proposed GTD2-MP algorithm. SMP is able to improve the convergence rate of our gradient TD method to:

$$(\textbf{GTD2} - \textbf{MP}): \quad O\left(\frac{\tau + ||A||_2}{n} + \frac{\sigma}{\sqrt{n}}\right), \tag{17}$$

Another example of an optimal solver is the stochastic accelerated primal-dual (SAPD) method [Chen *et al.*, 2013] which can reach the optimal convergence rate in (16). Due to space limitations, we are unable to present a more complete description, and refer interested readers to Juditsky *et al.* [2008]; Chen *et al.* [2013] for more details.

4 Empirical Evaluation

In this section, we compare the previous GTD2 method with our proposed GTD2-MP method using various domains. It should be mentioned that since the major focus of this paper is on policy evaluation, the comparative study focuses on value function approximation and thus comparisons on control learning performance is not reported in this paper.

The Baird example [Baird, 1995] is a well-known example to test the performance of off-policy convergent algorithms. Constant stepsize $\alpha = 0.005$ for GTD2 and $\alpha = 0.004$ for GTD2-MP, which are chosen via comparison studies as in [Dann *et al.*, 2014]. Figure 1 shows the MSPBE curve of GTD2, GTD2-MP of 8000 steps averaged over 200 runs. We can see that GTD2-MP has a significant improvement over the GTD2 algorithm wherein both the MSPBE and the variance are substantially reduced.

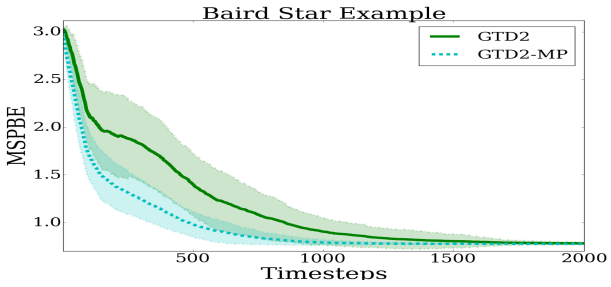

Figure 1: Off-Policy Convergence Comparison

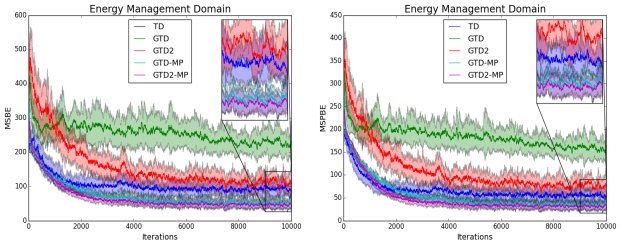

Figure 2: Energy Management Example

4.1 Energy Management Domain

In this experiment we compare the performance of the proposed algorithms using an energy management domain. The decision maker must decide how much energy to purchase or sell subject to stochastic prices. This problem is relevant in the context of utilities as well as in settings such as hybrid vehicles. The prices are generated from a Markov chain process. The amount of available storage is limited and it also degrades with use. The degradation process is based on the physical properties of lithium-ion batteries and discourages fully charging or discharging the battery. The energy arbitrage problem is closely related to the broad class of inventory management problems, with the storage level corresponding to the inventory. However, there are no known results describing the structure of optimal threshold policies in energy storage.

Note that since for this off-policy evaluation problem, the formulated $A\theta = b$ does not have a solution, and thus the optimal MSPBE(θ^*) (resp. MSBE(θ^*)) do not reduce to 0. The result is averaged over 200 runs, and $\alpha = 0.001$ for both GTD2 and GTD2-MP is chosen via comparison studies for each algorithm. As can be seen from FIgure 2, in the initial transit state, GTD2-MP performs much better than GTD2 at the transient state. Based on the above empirical results and many other experiments we have conducted, we conclude that GTD2-MP usually performs much better than the "vanilla" GTD2 algorithm.

5 Summary

The proximal gradient TD framework yields a principled approach to designing off-policy convergent gradient TD algorithms. We proposed and analyzed a new family of gradient TD algorithms with a nearly optimal convergence rate.

References

B. Ávila Pires and C. Szepesvári. Statistical linear estimation with penalized estimators: an application to reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1535–1542, 2012.

L. C. Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning*, pages 30–37, 1995.

H. H Bauschke and P. L Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.

V. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

S. Bubeck. Theory of convex optimization for machine learning. *arXiv:1405.4980*, 2014.

Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *arXiv:1309.5548*, 2013.

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

A. Juditsky and A. Nemirovski. *Optimization for Machine Learning*. MIT Press, 2011.

A. Juditsky, A. Nemirovskii, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *arXiv:0809.0815*, 2008.

B. Liu, S. Mahadevan, and J. Liu. Regularized off-policy TD-learning. In *Advances in Neural Information Processing Systems 25*, pages 845–853, 2012.

B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Proc. The 31st Conf. Uncertainty in Artificial Intelligence, Amsterdam, Netherlands*, 2015.

H. Maei. *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta, 2011.

S. Mahadevan and B. Liu. Sparse Q-learning with Mirror Descent. In *Proceedings of the Conference on Uncertainty in AI*, 2012.

S. Mahadevan, B. Liu, P. Thomas, W. Dabney, S. Giguere, N. Jacek, I. Gemp, and J. Liu. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv:1405.6757*, 2014.

Z. Qin and W. Li. Sparse Reinforcement Learning via Convex Optimization. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

R. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

R. Sutton, C. Szepesvári, and H. Maei. A convergent o(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Neural Information Processing Systems*, pages 1609–1616, 2008.

R. Sutton, H. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning*, pages 993–1000, 2009.

C. Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.

S. Valcarcel Macua, J. Chen, S. Zazo, and A. H Sayed. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2015.