Finite-Sample Analysis of Proximal Gradient TD Algorithms

Bo Liu
UMass Amherst
boliu@cs.umass.edu

Ji Liu
University of Rochester
jliu@cs.rochester.edu

Mohammad Ghavamzadeh
Adobe & INRIA Lille
Mohammad.ghavamzadeh@inria.fr

Sridhar Mahadevan
UMass Amherst
mahadeva@cs.umass.edu

Marek Petrik
IBM Research
marekpetrik@gmail.com

Abstract

In this paper, we show for the first time how gradient TD (GTD) reinforcement learning methods can be formally derived as true stochastic gradient algorithms, not with respect to their original objective functions as previously attempted, but rather using derived primal-dual saddle-point objective functions. We then 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. Two novel GTD algorithms are also proposed, namely projected GTD2 and GTD2-MP, which use proximal "mirror maps" to yield improved convergence guarantees and acceleration, respectively. 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

Obtaining a true stochastic gradient temporal difference method has been a longstanding goal of reinforcement learning (RL) [Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998], ever since it was discovered 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. Sutton *et al.* [2008, 2009] proposed the family of gradient-based temporal difference (GTD) algorithms which offer several interesting properties. A key property of this class of GTD algorithms is that they are asymptotically off-policy convergent, which was shown using stochastic approximation [Borkar, 2008]. This is quite important when we notice that 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 a rather ad-hoc splitting of terms. In this paper, we take a major step forward in resolving this problem by showing 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].

Since in real-world applications of RL, we have access to only a finite amount of data, finite-sample analysis of gradient TD algorithms is essential as it clearly shows the effect of the number of samples (and the parameters that play a role in the sampling budget of the algorithm) in their final performance. However, most of the work on finite-sample analysis in RL has been focused on batch RL (or approximate dynamic programming) algorithms (e.g., Kakade and Langford 2002; Munos and Szepesvári 2008; Antos *et al.* 2008; Lazaric *et al.* 2010a), especially those that are least squares TD (LSTD)-based (e.g., Lazaric *et al.* 2010b; Ghavamzadeh *et al.* 2010, 2011; Lazaric *et al.* 2012), and more importantly restricted to the on-policy setting. In this paper, we provide the finite-sample analysis of the GTD family of algorithms, a relatively novel class of gradient-based TD methods that are guaranteed to converge even in the off-policy setting, and for which, to the best of our knowledge, no finite-sample analysis has been reported. This analysis is challenging because **1)** the stochastic approximation methods that have been used to prove the asymptotic convergence of these algorithms do not address convergence rate analysis; **2)** as we explain in detail in Section 2.1, the techniques used for the analysis of the stochastic gradient methods cannot be applied here;

3) finally, the difficulty of finite-sample analysis in the off-policy setting.

The major contributions of this paper include the first finite-sample analysis of the class of gradient TD algorithms, as well as the design and analysis of several improved GTD methods that result from our novel approach of formulating gradient TD methods as true stochastic gradient algorithms w.r.t. a saddle-point objective function. We then use the techniques applied in the analysis of the stochastic gradient methods to propose a unified finite-sample analysis for the previously proposed as well as our novel gradient TD agorithms. Finally, given the results of our analysis, we study the GTD class of algorithms from several different perspectives, including acceleration in convergence, learning with biased importance sampling factors, etc.

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 $(\mathcal{S}, \mathcal{A}, P_{ss'}^a, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the sets of states and actions, the transition kernel $P_{ss'}^a$ 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 evaluate. 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

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

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 $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^n$, $s_i \sim \xi$, $a_i \sim \pi_b(\cdot|s_i)$, $s'_i \sim P(\cdot|s_i, a_i)$ 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

Before we present a finite-sample performance bound for GTD and GTD2, it would be helpful to give a brief overview of the existing literature on finite-sample analysis of the TD algorithms. The convergence rate of the TD algorithms mainly depends on (d, n, ν), where d is the size of the approximation space (the dimension of the feature vector), n is the number of samples, and ν is the smallest eigenvalue of the sample-based covariance matrix $\hat{C} = \hat{\Phi}^\top\hat{\Phi}$, i.e., $\nu = \lambda_{\min}(\hat{C})$.

Antos *et al.* [2008] proved an error bound of $O(\frac{d\log d}{n^{1/4}})$ for LSTD in bounded spaces. Lazaric *et al.* [2010b] proposed a LSTD analysis in leaner spaces and obtained a tighter bound of $O(\sqrt{\frac{d\log d}{n\nu}})$ and later used it to derive a bound for the least-squares policy iteration (LSPI) algorithm [Lazaric *et al.*, 2012]. Tagorti and Scherrer [2014] recently proposed the first convergence analysis for LSTD(λ) and derived a bound of $\tilde{O}(d/\nu\sqrt{n})$. The analysis is a bit different than the one in Lazaric *et al.* [2010b] and the bound is weaker in terms of d and ν. Another recent result is by Prashanth *et al.* [2014] that use stochastic approximation to solve LSTD(0), where the resulting algorithm is exactly TD(0) with random sampling (samples are drawn i.i.d. and not from a trajectory), and report a Markov design bound (the bound is computed only at the states used by the algorithm) of $O(\sqrt{\frac{d}{n\nu}})$ for LSTD(0). All these results are for the on-policy setting, except the one by Antos *et al.* [2008] that also holds for the off-policy formulation. Another work in the off-policy setting is by Ávila Pires and Szepesvári [2012] that uses a bounding trick and improves the result of Antos *et al.* [2008] by a $\log d$ factor.

The line of research reported here has much in common with work on 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 work began originally with 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], 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 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 SGs in order 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.

A particular type of convex-concave saddle-point formulation is formally defined as

$$\min_\theta \max_y \big(L(\theta,y) = \langle b - A\theta, y\rangle + F(\theta) - K(y)\big), \quad (11)$$

where $F(\theta)$ is a convex function and $K(y)$ is a smooth convex function such that

$$K(y) - K(x) - \langle \nabla K(x), y-x\rangle \le \frac{L_K}{2}||x-y||^2. \quad (12)$$

Next we follow Juditsky *et al.* [2008]; Nemirovski *et al.* [2009]; Chen *et al.* [2013] and define the following error function for the saddle-point problem (11).

Definition 1. *The error function of the saddle-point problem* (11) *at each point* (θ', y') *is defined as*

$$\mathrm{Err}(\theta', y') = \max_y \; L(\theta', y) - \min_\theta \; L(\theta, y'). \quad (13)$$

In this paper, we consider the saddle-point problem (11) with $F(\theta) = 0$ and $K(y) = \frac{1}{2}||y||_M^2$, i.e.,

$$\min_\theta \max_y \Big(L(\theta,y) = \langle b - A\theta, y\rangle - \frac{1}{2}||y||_M^2\Big), \quad (14)$$

where A and b were defined by Eq. 3, and M is a positive definite matrix. It is easy to show that $K(y) = \frac{1}{2}||y||_M^2$ satisfies the condition in Eq. 12.

We first show in Proposition 1 that if (θ^*, y^*) is the saddle-point of problem (14), 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

$$y^*(\theta) = M^{-1}(b - A\theta). \quad (15)$$

The result follows by plugging y^* into (14) 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* (14) *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 (14) (ascending in y and descending in θ) may be written as

$$\begin{aligned} y_{t+1} &= y_t + \alpha_t (b - A\theta_t - My_t), \\ \theta_{t+1} &= \theta_t + \alpha_t A^\top y_t. \end{aligned} \quad (16)$$

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 (16) with their unbiased estimates $\hat{A}$, $\hat{b}$, and $\hat{C}$ from Eq. 4, which completes the proof. □

4 FINITE-SAMPLE ANALYSIS

In this section, we provide a finite-sample analysis for a revised version of the GTD/GTD2 algorithms. We first describe the revised GTD algorithms in Section 4.1 and then dedicate the rest of Section 4 to their sample analysis. Note that from now on we use the M matrix (and its unbiased estimate $\hat{M}_t$) to have a unified analysis for GTD and GTD2 algorithms. As described earlier, M is replaced by the identity matrix I in GTD and by the covariance matrix C (and its unbiased estimate $\hat{C}_t$) in GTD2.

4.1 THE REVISED GTD ALGORITHMS

The revised GTD algorithms that we analyze in this paper (see Algorithm 1) have three differences with the standard GTD algorithms of Eqs. 8 and 9 (and Eq. 16). **1)** We guarantee that the parameters θ and y remain bounded by projecting them onto bounded convex feasible sets Θ and Y defined in Assumption 2. In Algorithm 1, we denote by Π_Θ and Π_Y, the projection into sets Θ and Y, respectively. This is standard in stochastic approximation algorithms and has been used in off-policy TD(λ) [Yu, 2012] and actor-critic algorithms (e.g., Bhatnagar *et al.* 2009). **2)** after n iterations (n is the number of training samples in $\mathcal{D}$), the algorithms return the weighted (by the step size) average of the parameters at all the n iterations (see Eq. 18). **3)** The step-size α_t is selected as described in the proof of Proposition 3 in the supplementary material. Note that this fixed step size of $O(1/\sqrt{n})$ is required for the high-probability bound in Proposition 3 (see Nemirovski *et al.* 2009 for more details).

4.2 ASSUMPTIONS

In this section, we make several assumptions on the MDP and basis functions that are used in our finite-sample analysis of the revised GTD algorithms. These assumptions are

Algorithm 1 Revised GTD Algorithms

```
for $t = 1, \dots, n$ do
    Update parameters
        $y_{t+1} = \Pi_Y\Big(y_t + \alpha_t(\hat{b_t} - \hat{A}_t\theta_t - \hat{M}_t y_t)\Big)$
        $\theta_{t+1} = \Pi_\Theta\Big(\theta_t + \alpha_t \hat{A}_t^\top y_t\Big)$        (17)
end for
OUTPUT
    $\bar{\theta}_n := \frac{\sum_{t=1}^n \alpha_t\theta_t}{\sum_{t=1}^n \alpha_t}$ ,  $\bar{y}_n := \frac{\sum_{t=1}^n \alpha_t y_t}{\sum_{t=1}^n \alpha_t}$        (18)
```

quite standard and are similar to those made in the prior work on GTD algorithms [Sutton *et al.*, 2008, 2009; Maei, 2011] and those made in the analysis of SG algorithms [Nemirovski *et al.*, 2009].

Assumption 2. ***(Feasibility Sets)*** *We define the bounded closed convex sets $\Theta \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}^d$ as the feasible sets in Algorithm 1. We further assume that the saddle-point (θ^*, y^*) of the optimization problem* (14) *belongs to $\Theta \times Y$. We also define $D_\theta := \big[\max_{\theta\in\Theta} ||\theta||_2^2 - \min_{\theta\in\Theta} ||\theta||_2^2\big]^{1/2}$, $D_y := \big[\max_{y\in Y} ||y||_2^2 - \min_{y\in Y} ||y||_2^2\big]^{1/2}$, and $R = \max\big\{\max_{\theta\in\Theta} ||\theta||_2, \max_{y\in Y} ||y||_2\big\}$.*

Assumption 3. ***(Non-singularity)*** *We assume that the covariance matrix $C = \mathbb{E}[\phi_i\phi_i^\top]$ and matrix $A = \mathbb{E}\big[\rho_i\phi_i(\Delta\phi_i)^\top\big]$ are non-singular.*

Assumption 4. ***(Boundedness)*** *Assume the features $(\phi_i, \phi_i^{'})$ have uniformly bounded second moments. This together with the boundedness of features (by L) and importance weights (by $\rho_{\max}$) guarantees that the matrices A and C, and vector b are uniformly bounded.*

This assumption guarantees that for any $(\theta, y) \in \Theta \times Y$, the unbiased estimators of $b - A\theta - My$ and $A^\top y$, i.e.,

$$\mathbb{E}[\hat{b}_t - \hat{A}_t\theta - \hat{M}_t y] = b - A\theta - My,$$
$$\mathbb{E}[\hat{A}_t^\top y] = A^\top y, \tag{19}$$

all have bounded variance, i.e.,

$$\mathbb{E}\big[||\hat{b}_t - \hat{A}_t\theta - \hat{M}_t y - (b - A\theta - My)||^2\big] \le \sigma_1^2,$$
$$\mathbb{E}\big[||\hat{A}_t^\top y - A^\top y||^2\big] \le \sigma_2^2, \tag{20}$$

where σ_1 and σ_2 are non-negative constants. We further define

$$\sigma^2 = \sigma_1^2 + \sigma_2^2. \tag{21}$$

Assumption 4 also gives us the following "light-tail" assumption. There exist constants $M_{*,\theta}$ and $M_{*,y}$ such that

$$\mathbb{E}[\exp\{\frac{||\hat{b}_t - \hat{A}_t\theta - \hat{M}_t y||^2}{M_{*,\theta}^2}\}] \le \exp\{1\},$$
$$\mathbb{E}[\exp\{\frac{||\hat{A}_t^\top y||^2}{M_{*,y}^2}\}] \le \exp\{1\}. \tag{22}$$

This "light-tail" assumption is equivalent to the assumption in Eq. 3.16 in Nemirovski *et al.* [2009] and is necessary for the high-probability bound of Proposition 3. We will show how to compute $M_{*,\theta}, M_{*,y}$ in the Appendix.

4.3 FINITE-SAMPLE PERFORMANCE BOUNDS

The finite-sample performance bounds that we derive for the GTD algorithms in this section are for the case that the training set $\mathcal{D}$ has been generated as discussed in Section 2. We further discriminate between the on-policy ($\pi = \pi_b$) and off-policy ($\pi \neq \pi_b$) scenarios. The sampling scheme used to generate $\mathcal{D}$, in which the first state of each tuple, s_i, is an i.i.d. sample from a distribution ξ, also considered in the original GTD and and GTD2 papers, for the analysis of these algorithms, and not in the experiments [Sutton *et al.*, 2008, 2009]. Another scenario that can motivate this sampling scheme is when we are given a set of high-dimensional data generated either in an on-policy or off-policy manner, and d is so large that the value function of the target policy cannot be computed using a least-squares method (that involves matrix inversion), and iterative techniques similar to GTD/GTD2 are required.

We first derive a high-probability bound on the error function of the saddle-point problem (14) at the GTD solution $(\bar{\theta}_n, \bar{y}_n)$. Before stating this result in Proposition 3, we report the following lemma that is used in its proof.

Lemma 2. *The induced ℓ_2-norm of matrix A and the ℓ_2-norm of vector b are bounded by*

$$||A||_2 \le (1+\gamma)\rho_{\max}L^2 d, \qquad ||b||_2 \le \rho_{\max}L\mathrm{R}_{\max}. \tag{23}$$

Proof. See the supplementary material. □

Proposition 3. *Let $(\bar{\theta}_n, \bar{y}_n)$ be the output of the GTD algorithm after n iterations (see Eq. 18). Then, with probability at least $1-\delta$, we have*

$$\mathrm{Err}(\bar{\theta}_n, \bar{y}_n) \le \sqrt{\frac{5}{n}}(8 + 2\log\frac{2}{\delta})R^2 \tag{24}$$
$$\times\Bigg(\rho_{\max}L\Big(2(1+\gamma)Ld + \frac{R_{\max}}{R}\Big) + \tau + \frac{\sigma}{R}\Bigg),$$

where $\mathrm{Err}(\bar{\theta}_n, \bar{y}_n)$ is the error function of the saddle-point problem (14) *defined by Eq. 13, R defined in Assumption 2, σ is from Eq. 21, and $\tau = \sigma_{\max}(M)$ is the largest singular value of M, which means $\tau = 1$ for GTD and $\tau = \sigma_{\max}(C)$ for GTD2.*

Proof. See the supplementary material. □

Theorem 1. *Let $\bar{\theta}_n$ be the output of the GTD algorithm after n iterations (see Eq. 18). Then, with probability at least $1-\delta$, we have*

$$\frac{1}{2}||A\bar{\theta}_n - b||_\xi^2 \le \tau\xi_{\max}\,\mathrm{Err}(\bar{\theta}_n, \bar{y}_n). \tag{25}$$

Proof. From Proposition 1, for any θ, we have

$$\max_y \; L(\theta, y) = \frac{1}{2}||A\theta - b||^2_{M^{-1}}.$$

Given Assumption 3, the system of linear equations $A\theta = b$ has a solution θ^*, i.e., the (off-policy) fixed-point θ^* exists, and thus, we may write

$$\begin{aligned}\min_\theta \; \max_y \; L(\theta, y) &= \min_\theta \; \frac{1}{2}||A\theta - b||^2_{M^{-1}} \\ &= \frac{1}{2}||A\theta^* - b||^2_{M^{-1}} = 0.\end{aligned}$$

In this case, we also have[2]

$$\begin{aligned}\min_\theta \; L(\theta, y) \leq \max_y \; \min_\theta \; L(\theta, y) &\leq \min_\theta \; \max_y \; L(\theta, y) \\ &= \frac{1}{2}||A\theta^* - b||^2_{M^{-1}} = 0. \qquad (26)\end{aligned}$$

From Eq. 26, for any $(\theta, y) \in \Theta \times Y$ including $(\bar{\theta}_n, \bar{y}_n)$, we may write

$$\begin{aligned}\text{Err}(\bar{\theta}_n, \bar{y}_n) &= \max_y \; L(\bar{\theta}_n, y) - \min_\theta \; L(\theta, \bar{y}_n) \qquad (27) \\ &\geq \max_y \; L(\bar{\theta}_n, y) = \frac{1}{2}||A\bar{\theta}_n - b||^2_{M^{-1}}.\end{aligned}$$

Since $||A\bar{\theta}_n - b||^2_\xi \leq \tau\xi_{\max} \; ||A\bar{\theta}_n - b||^2_{M^{-1}}$, where τ is the largest singular value of M, we have

$$\frac{1}{2}||A\bar{\theta}_n - b||^2_\xi \leq \frac{\tau\xi_{\max}}{2}||A\bar{\theta}_n - b||^2_{M^{-1}} \leq \tau\xi_{\max} \; \text{Err}(\bar{\theta}_n, \bar{y}_n). \qquad (28)$$

The proof follows by combining Eqs. 28 and Proposition 3. It completes the proof. □

With the results of Proposition 3 and Theorem 1, we are now ready to derive finite-sample bounds on the performance of GTD/GTD2 in both on-policy and off-policy settings.

4.3.1 On-Policy Performance Bound

In this section, we consider the on-policy setting in which the behavior and target policies are equal, i.e., $\pi_b = \pi$, and the sampling distribution ξ is the stationary distribution of the target policy π (and the behavior policy π_b). We use Lemma 3 to derive our on-policy bound. The proof of this lemma can be found in Geist *et al.* [2012].

Lemma 3. *For any parameter vector θ and corresponding $\hat{v} = \Phi\theta$, the following equality holds*

$$V - \hat{v} = (I - \gamma\Pi P)^{-1}\left[(V - \Pi V) + \Phi C^{-1}(b - A\theta)\right]. \quad (29)$$

Using Lemma 3, we derive the following performance bound for GTD/GTD2 in the on-policy setting.

Proposition 4. *Let V be the value of the target policy and $\bar{v}_n = \Phi\bar{\theta}_n$, where $\bar{\theta}_n$ defined by* (18)*, be the value function returned by on-policy GTD/GTD2. Then, with probability at least $1 - \delta$, we have*

$$||V - \bar{v}_n||_\xi \leq \frac{1}{1-\gamma}\left(||V - \Pi V||_\xi + \frac{L}{\nu}\sqrt{2d\tau\xi_{\max}\text{Err}(\bar{\theta}_n, \bar{y}_n)}\right) \qquad (30)$$

where $\mathrm{Err}(\bar{\theta}_n, \bar{y}_n)$ is upper-bounded by Eq. 24 in Proposition 3, with $\rho_{\max} = 1$ (on-policy setting).

Proof. See the supplementary material. □

Remark: It is important to note that Proposition 4 shows that the error in the performance of the GTD/GTD2 algorithm in the on-policy setting is of $O\left(\frac{L^2 d\sqrt{\tau\xi_{\max}\log\frac{1}{\delta}}}{n^{1/4}\nu}\right)$. Also note that the term $\frac{\tau}{\nu}$ in the GTD2 bound is the conditioning number of the covariance matrix C.

4.3.2 Off-Policy Performance Bound

In this section, we consider the off-policy setting in which the behavior and target policies are different, i.e., $\pi_b \neq \pi$, and the sampling distribution ξ is the stationary distribution of the behavior policy π_b. We assume that off-policy fixed-point solution exists, i.e., there exists a θ^* satisfying $A\theta^* = b$. Note that this is a direct consequence of Assumption 3 in which we assumed that the matrix A in the off-policy setting is non-singular. We use Lemma 4 to derive our off-policy bound. The proof of this lemma can be found in Kolter [2011]. Note that $\kappa(\bar{D})$ in his proof is equal to $\sqrt{\rho_{\max}}$ in our paper.

Lemma 4. *If Ξ satisfies the following linear matrix inequality*

$$\begin{bmatrix} \Phi^\top\Xi\Phi & \Phi^\top\Xi P\Phi \\ \Phi^\top P^\top\Xi\Phi & \Phi^\top\Xi\Phi \end{bmatrix} \succeq 0 \qquad (31)$$

and let θ^* be the solution to $A\theta^* = b$, then we have

$$||V - \Phi\theta^*||_\xi \leq \frac{1 + \gamma\sqrt{\rho_{\max}}}{1-\gamma}||V - \Pi V||_\xi. \qquad (32)$$

Note that the condition on Ξ in Eq. 31 guarantees that the behavior and target policies are not too far away from each other. Using Lemma 4, we derive the following performance bound for GTD/GTD2 in the off-policy setting.

Proposition 5. *Let V be the value of the target policy and $\bar{v}_n = \Phi\bar{\theta}_n$, where $\bar{\theta}_n$ is defined by* (18)*, be the value function returned by off-policy GTD/GTD2. Also let the sampling distribution Ξ satisfies the condition in Eq. 31. Then,*

[2] We may write the second inequality as an equality for our saddle-point problem defined by Eq. 14.

with probability at least $1-\delta$*, we have*

$$\begin{aligned}||V-\bar{v}_n||_\xi \le{}& \frac{1+\gamma\sqrt{\rho_{\max}}}{1-\gamma}||V-\Pi V||_\xi \qquad (33)\\ &+\sqrt{\frac{2\tau_C\tau\xi_{\max}}{\sigma_{\min}(A^\top M^{-1}A)}\mathrm{Err}(\bar{\theta}_n,\bar{y}_n)},\end{aligned}$$

where $\tau_C=\sigma_{\max}(C)$*.*

Proof. See the supplementary material. □

5 ACCELERATED ALGORITHM

As discussed at the beginning of Section 3, 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, as we will show in Section 4, 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:[3]

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

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 2, and the experimental comparison study between GTD2 and GTD2-MP is reported in Section 7.

6 FURTHER ANALYSIS

6.1 ACCELERATION 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. 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 as in above analysis. As can be seen from the above analysis, the convergence rate of the GTD algorithms family is

$$(\mathbf{GTD/GTD2}):\quad O\left(\frac{\tau+||A||_2+\sigma}{\sqrt{n}}\right) \qquad (35)$$

Algorithm 2 GTD2-MP

```
for $t = 1, \dots, n$ do
    Update parameters
        $\delta_t = r_t - \theta_t^\top\Delta\phi_t$
        $y_t^m = y_t + \alpha_t(\rho_t\delta_t - \phi_t^\top y_t)\phi_t$
        $\theta_t^m = \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t)$
        $\delta_t^m = r_t - (\theta_t^m)^\top\Delta\phi_t$
        $y_{t+1} = y_t + \alpha_t(\rho_t\delta_t^m - \phi_t^\top y_t^m)\phi_t$
        $\theta_{t+1} = \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t^m)$
end for
OUTPUT
    $\bar{\theta}_n := \frac{\sum_{t=1}^n\alpha_t\theta_t}{\sum_{t=1}^n\alpha_t} \quad , \quad \bar{y}_n := \frac{\sum_{t=1}^n\alpha_t y_t}{\sum_{t=1}^n\alpha_t} \qquad (34)$
```

In this section, we raise an 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 (14) is

$$(\mathbf{Optimal}):\quad O\left(\frac{\tau}{n^2}+\frac{||A||_2}{n}+\frac{\sigma}{\sqrt{n}}\right) \qquad (36)$$

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 accelerate the convergence rate of our gradient TD method to:

$$(\mathbf{SMP}):\quad O\left(\frac{\tau+||A||_2}{n}+\frac{\sigma}{\sqrt{n}}\right), \qquad (37)$$

and stochastic accelerated primal-dual (SAPD) method [Chen *et al.*, 2013] which can reach the optimal convergence rate in (36). 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.

6.2 LEARNING WITH BIASED ρ_t

The importance weight factor ρ_t is lower bounded by 0, but yet may have an arbitrarily large upper bound. In real applications, the importance weight factor ρ_t may not be estimated exactly, i.e., the estimation $\hat{\rho}_t$ is a biased estimation of the true ρ_t. To this end, the stochastic gradient we obtained is not unbiased gradient of $L(\theta,y)$ anymore. This falls into a broad category of learning with inexact stochastic gradient, or termed as stochastic gradient methods with an inexact oracle [Devolder, 2011]. Given the inexact stochastic gradient, the convergence rate and performance bound become much worse than the results with

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

exact stochastic gradient. Based on the analysis by Juditsky *et al.* [2008], we have the error bound for inexact estimation of ρ_t.

Proposition 6. *Let $\bar{\theta}_n$ be defined as above. Assume at the t-th iteration, $\hat{\rho}_t$ is the estimation of the importance weight factor ρ_t with bounded bias such that $\mathbb{E}[\hat{\rho}_t - \rho_t] \le \epsilon$. The convergence rates of GTD/GTD2 algorithms with iterative averaging is as follows, i.e.,*

$$||A\bar{\theta}_n - b||^2_{M^{-1}} \le O\left(\frac{\tau + ||A||_2 + \sigma}{\sqrt{n}}\right) + O(\epsilon) \quad (38)$$

This implies that the inexact estimation of ρ_t may cause disastrous estimation error, which implies that an exact estimation of ρ_t is very important.

6.3 FINITE-SAMPLE ANALYSIS OF ONLINE LEARNING

Another more challenging scenario is online learning scenario, where the samples are interactively generated by the environment, or by an interactive agent. The difficulty lies in that the sample distribution does not follow i.i.d sampling condition anymore, but follows an underlying Markov chain $\mathcal{M}$. If the Markov chain $\mathcal{M}$'s mixing time is small enough, i.e., the sample distribution reduces to the stationary distribution of π_b very fast, our analysis still applies. However, it is usually the case that the underlying Markov chain's mixing time τ_{mix} is not small enough. The analysis result can be obtained by extending the result of recent work [Duchi *et al.*, 2012] from strongly convex loss functions to saddle-point problems, which is non-trivial and is thus left for future work.

6.4 DISCUSSION OF TDC ALGORITHM

Now we discuss the limitation of our analysis with regard to the temporal difference with correction (TDC) algorithm [Sutton *et al.*, 2009]. Interestingly, the TDC algorithm seems not to have an explicit saddle-point representation, since it incorporates the information of the optimal $y_t^*(\theta_t)$ into the update of θ_t, a quasi-stationary condition which is commonly used in two-time-scale stochastic approximation approaches. An intuitive answer to the advantage of TDC over GTD2 is that the TDC update of θ_t can be considered as incorporating the prior knowledge into the update rule: for a stationary θ_t, if the optimal $y_t^*(\theta_t)$ has a closed-form solution or is easy to compute, then incorporating this $y_t^*(\theta_t)$ into the update law tends to accelerate the algorithm's convergence performance. For the GTD2 update, note that there is a sum of two terms where y_t appears, which are $\rho_t(\phi_t - \gamma\phi_t')(y_t^T\phi_t) = \rho_t\phi_t(y_t^T\phi_t) - \gamma\rho_t\phi_t'(y_t^T\phi_t)$. Replacing y_t in the first term with $y_t^*(\theta_t) = \mathbb{E}[\phi_t\phi_t^T]^{-1}\mathbb{E}[\rho_t\delta_t(\theta_t)\phi_t]$, we have the TDC update rule. Note that in contrast to GTD/GTD2, TDC is a two-time scale algorithm; Also, note that TDC does not minimize *any* objective functions and the convergence of TDC requires more restrictions than GTD2 as shown by Sutton *et al.* [2009].

7 EMPIRICAL EVALUATION

In this section, we compare the previous GTD2 method with our proposed GTD2-MP method using various domains with regard to their value function approximation performance capability. 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.

7.1 BAIRD DOMAIN

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.

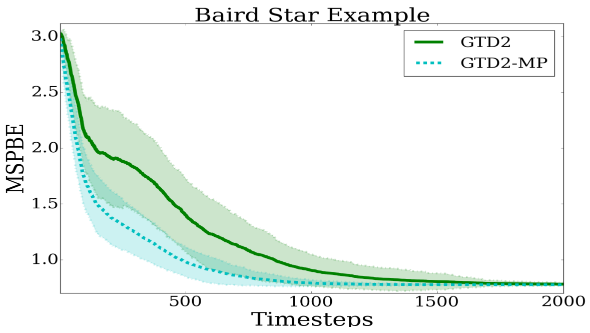

Figure 1: Off-Policy Convergence Comparison

7.2 50-STATE CHAIN DOMAIN

The 50 state chain [Lagoudakis and Parr, 2003] is a standard MDP domain. There are 50 discrete states $\{s_i\}_{i=1}^{50}$ and two actions moving the agent left $s_i \to s_{\max(i-1,1)}$ and right $s_i \to s_{\min(i+1,50)}$. The actions succeed with probability 0.9; failed actions move the agent in the opposite direction. The discount factor is $\gamma = 0.9$. The agent receives a reward of $+1$ when in states s_{10} and s_{41}. All other states have a reward of 0. In this experiment, we compare the performance of the value approximation w.r.t different set of stepsizes $\alpha = 0.0001, 0.001, 0.01, 0.1, 0.2, \cdots, 0.9$ using the BEBF basis [Parr *et al.*, 2007], and Figure 2 shows the value function approximation result, where the cyan curve

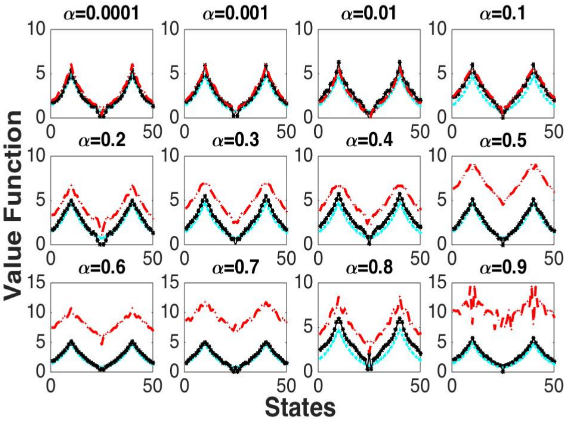

Figure 2: Chain Domain

is the true value function, the red dashed curve is the GTD result,and the black curve is the GTD2-MP result. From the figure, one can see that GTD2-MP is much more robust with stepsize choice than the GTD2 algorithm.

7.3 ENERGY MANAGEMENT DOMAIN

In this experiment we compare the performance of the algorithms on 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 3, in the initial transit state, GTD2-MP performs much better than GTD2 at the transient state. Then after reaching the steady state, as can be seen from Table 1, we can see that GTD2-MP reaches better steady state solution than the GTD algorithm. Based on the above empirical results and many other experiments we have conducted in other domains, we can conclude that GTD2-MP usually performs much better than the "vanilla" GTD2 algorithm.

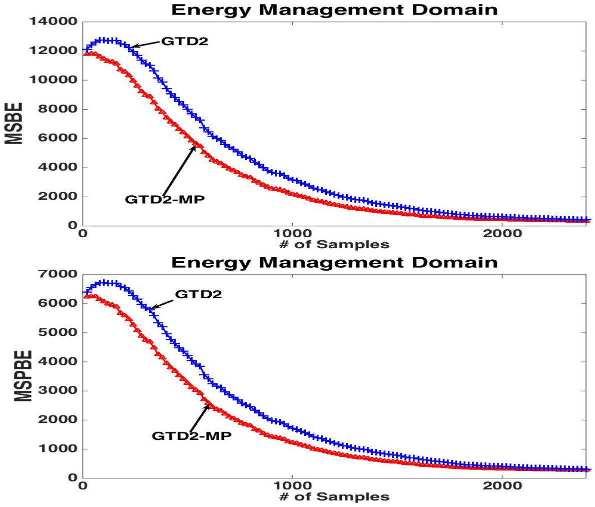

Figure 3: Energy Management Example

Algorithm	MSPBE	MSBE
GTD2	176.4	228.7
GTD2-MP	138.6	191.4

Table 1: Steady State Performance Comparison

8 SUMMARY

In this paper, we showed how gradient TD methods can be shown to be true stochastic gradient methods with respect to a saddle-point primal-dual objective function, which paved the way for the finite-sample analysis of off-policy convergent gradient-based temporal difference learning algorithms such as GTD and GTD2. Both error bound and performance bound are provided, which shows that the value function approximation bound of the GTD algorithms family is $O\left(\frac{d}{n^{1/4}}\right)$. Further, two revised algorithms, namely the projected GTD2 algorithm and the accelerated GTD2-MP algorithm, are proposed. There are many interesting directions for future research. Our framework can be easily used to design regularized sparse gradient off-policy TD methods. One interesting direction is to investigate the convergence rate and performance bound for the TDC algorithm, which lacks a saddle-point formulation. The other is to explore tighter value function approximation bounds for off-policy learning.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-1216467. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

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

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.

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

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.

O. Devolder. Stochastic first order methods in smooth convex optimization. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics, 2011.

J. Duchi, A. Agarwal, M. Johansson, and M. Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

M. Geist, B. Scherrer, A. Lazaric, and M. Ghavamzadeh. A Dantzig Selector approach to temporal difference learning. In *International Conference on Machine Learning*, pages 1399–1406, 2012.

M. Ghavamzadeh, A. Lazaric, O. Maillard, and R. Munos. LSTD with Random Projections. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 721–729, 2010.

M. Ghavamzadeh, A. Lazaric, R. Munos, and M. Hoffman. Finite-sample analysis of Lasso-TD. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1177–1184, 2011.

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.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

Z. Kolter. The fixed points of off-policy TD. In *Advances in Neural Information Processing Systems 24*, pages 2169–2177, 2011.

M. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Analysis of a classification-based policy iteration algorithm. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 607–614, 2010.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Finite-sample analysis of LSTD. In *Proceedings of 27th International Conference on Machine Learning*, pages 615–622, 2010.

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

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

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.

R. Munos and Cs. Szepesvári. Finite time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value function approximation. In *Proceedings of the International Conference on Machine Learning*, pages 737–744, 2007.

LA Prashanth, N. Korda, and R. Munos. Fast LSTD using stochastic approximation: Finite time analysis and application to traffic control. In *Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer, 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.

M. Tagorti and B. Scherrer. Rate of convergence and error bounds for LSTD (λ). *arXiv:1405.3229*, 2014.

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