Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

Finite sample analysis of the GTD Policy Evaluation Algorithms in Markov Setting

Yue Wang

Editor: Kevin Murphy and Bernhard Schölkopf

Abstract

In reinforcement learning (RL), the key component is policy evaluation, which aims to estimate the value function (i.e., expected long-term accumulated reward starting from a state) of a given policy. With a good policy evaluation method, the RL algorithms can estimate the value functions of the given policy more accurately and find a better policy. When the state space is large or continuous, *Gradient-based Temporal Difference(GTD)* algorithms with linear function approximation to the value function are widely used. Considering that the collection of the evaluation data is very likely to be both time and reward consuming, to get a clear understanding of the finite sample performance of the GTD algorithms is very important to the efficiency of policy evaluation and the entire RL algorithms. Previous work converted GTD algorithms into a convex-concave saddle point problem and provided the finite sample analysis of the GTD algorithms with constant step size under the assumption that data are i.i.d. generated. However, as we know, in RL problems, the data are generated from Markov processes rather than i.i.d. and step size is set in different ways. In this paper, in the realistic Markov setting, we derive finite sample bounds both in expectation and with high probability for the general convex-concave saddle point problem, and hence for the GTD algorithms. Our bounds show that, in the Markov setting, (1) with variants of step size, GTD algorithms converge; (2) the convergence rate is determined by the step size, and related to the mixing time of the Markov process; (3) we explain that the experience reply trick is effective, since it can improve the mixing property of the Markov process. To the best of our knowledge, our analysis is the first to provide finite sample bounds for the GTD algorithms in Markov setting.

Keywords: Bayesian Networks, Mixture Models, Chow-Liu Trees

1. Introduction

Reinforcement Learning (RL) (Sutton and Barto (1998)) technologies are very powerful to learn how to interact with environments, and has variants of important applications, such as robotics, computer games and so on (Kober et al. (2013), Mnih et al. (2015), Silver et al. (2016), Bahdanau et al. (2016)).

In RL problem, an agent observes the current state, takes an action following a policy based on the current state, receives a reward from the environment, and the environment transits to the next state in Markov, and again repeats these steps. The goal of the RL algorithms is to find the optimal policy which leads to the maximum long-term reward. The value function of a fixed policy for a state is defined as the expected long-term accumulated reward the agent would receive by following the fixed policy starting from this state. Policy evaluation aims to accurately estimate the value of all states under a given policy, which is a key component in RL (Sutton and Barto (1998), Dann et al. (2014)). A better policy evaluation method will help us to improve the current policy and find the optimal policy.

When the state space is large or continuous, it is inefficient to represent the value function over all the states by a look-up table. A common approach is to extract features for states and use parameterized function over the feature space to approximate the value function. In applications, there are linear approximation and non-linear approximation (e.g. neural networks) to the value function. In this paper, we will focus on the linear approximation (Sutton et al. (2009a),Sutton et al. (2009b),Liu et al. (2015)). Leveraging the localization technique in Bhatnagar et al. (2009), the results can be generated into non-linear cases with extra efforts. We leave it as future work.

In policy evaluation with linear approximation, there were substantial work on the temporal-difference (TD) method, which uses the Bellman equation to update the value function during the learning process (Sutton (1988),Tsitsiklis et al. (1997)). Recently, Sutton et al. (2009a) Sutton et al. (2009b) have proposed *Gradient-based Temporal Difference (GTD)* algorithms which use gradient information of the error from the Bellman equation to update the value function. It is shown that, GTD algorithms have achieved the lower-bound of the storage and computation complexity, making them powerful to handle high dimensional big data. Therefore, now GTD algorithms are widely used in policy evaluation problem and the policy evaluation step in practical RL algorithms (Bhatnagar et al. (2009),Silver et al. (2014)).

However, we don't have sufficient theory to tell us about the finite sample performance of the GTD algorithms. To be specific, will the evaluation process converge? If yes, how many samples we need to get a target evaluation error? Will the step size in GTD algorithms influence the finite sample error? How to explain the effectiveness of the practical tricks, such as experience reply? Considering that the collection of the evaluation data is very likely to be both time and reward consuming, to get a clear understanding of the finite sample performance of the GTD algorithms is very important to the efficiency of policy evaluation and the entire RL algorithms.

Previous work (Liu et al. (2015)) converted the objective function of GTD algorithms into a convex-concave saddle problem and conduct the finite sample analysis for GTD with constant step size in the independent and identical distributed (i.i.d.) setting in which data are simply assumed to be generated by i.i.d.. However, in RL problem, the date are generated from an agent who interacts with the environment step by step, and the state will transit in Markov as introduced previously. As a result, the data are generated from a Markov process but not i.i.d.. In addition, the work did not study the decreasing step size , which are also commonly-used in many gradient based algorithms(Sutton et al. (2009a),Sutton et al. (2009b),Yu (2015)). Thus, the results from previous work cannot provide precise answers to the above questions for the finite sample performance of the GTD algorithms.

In this paper, we perform the finite sample analysis for the GTD algorithms in the more realistic Markov setting. To achieve the goal, first of all, same with Liu et al. (2015), we consider the stochastic gradient descent algorithms of the general convex-concave saddle point problems, which include the GTD algorithms. The quality of the solution is measured by the primal-dual gap in the convex-concave saddle point optimization problem (Liu et al. (2015) , Nemirovski et al. (2009)). The finite sample analysis for convex-concave optimization in Markov setting is challenging. On one hand, in Markov setting, the non-i.i.d. sampled gradients are no longer unbiased estimation of the gradients. Thus, the proof technique for i.i.d. setting cannot be applied. On the other hand, although for convex optimization problem, with the Markov gradients, SGD still has convergence guarantee, it is much more difficult to obtain the same results in the more complex convex-concave optimization problem.

To overcome the challenge, we design a novel decomposition ( ) of the error function. The intuition of the decomposition and key techniques are as follows: (1) Although samples are not i.i.d., for large enough τ, the sample at time $t+\tau$ is "nearly independent" of the sample at time t, and its distribution is "very close" to the stationary distribution. (2) We split the random variables in the objective related to $\mathbb{E}$ operator and the variables related to max operator into different terms in order to control them respectively. It is non-trivial, and we construct a sequence of auxiliary random variables to do so. (3) All constructions above need to be carefully considered the measurable issues in the Markov setting.

By using the above techniques, we prove a novel finite sample bound in expectation for the convex-concave saddle point problem. Based on the bound in expectation, we construct martingale difference sequences and apply Azuma's inequality to obtain the finite sample bound with high probability. Considering the GTD algorithms are specific convex-concave saddle point optimization methods, we finally obtained the finite sample bounds in expectation and with high probability for the GTD algorithms, in the realistic Markov setting for RL. To the best of our knowledge, our analysis is the first to provide finite sample bounds for the GTD algorithms in Markov setting.

We have the following discussions based on our finite sample bounds.
(1) GTD algorithms do converge, under a flexible condition on the step size, i.e. $\sum_{t=1}^T \alpha_t \to \infty, \frac{\sum_{t=1}^T \alpha_t^2}{\sum_{t=1}^T \alpha_t} < \infty$, as $T \to \infty$, where α_t is the step size. The popular step size settings all satisfy this condition.
(2) The convergence rate in expectation is $\mathcal{O}\left(\sqrt{(1+\tau(\eta))\frac{\sum_{t=1}^T \alpha_t^2}{\sum_{t=1}^T \alpha_t}}\right)$ and with probability $1-\delta$ is $\mathcal{O}\left(\sqrt{(1+\tau(\eta))\frac{\sum_{t=1}^T \alpha_t^2}{\sum_{t=1}^T \alpha_t} + \frac{\sqrt{\tau(\eta)\log(\frac{\tau(\eta)}{\delta})\sum_{t=1}^T \alpha_t^2}}{\sum_{t=1}^T \alpha_t}}\right)$, where $\tau(\eta)$ is the mixing time of the Markov process, and η is a constant. Different step size will lead to different convergence rates.
(3) The experience reply trick is effective, since it can improve the mixing property of the Markov process.

Finally, we conduct simulation experiments to verify our theoretical founding. All the conclusions from the analysis are consistent with our empirical observations.

2. Background and Notation

In this section, we briefly introduce the GTD algorithms and related works.

2.1 Gradient-based TD algorithms

Consider the reinforcement learning problem with Markov decision process(MDP) $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P = \{P_{s,s'}^a; s, s' \in \mathcal{S}, a \in \mathcal{A}\}$ is the transition matrix and $P_{s,s'}^a$ is the transition probability from state s to state s' after taking action a,

$R = \{R(s,a); s \in \mathcal{S}, a \in \mathcal{A}$ is the reward function and $R(s,a)$ is the reward received at state s if taking action a, and $0 < \gamma < 1$ is the discount factor. A policy function $\mu : \mathcal{A} \times \mathcal{S} \to [0,1]$ indicates the probability to take each action at each state. Value function for policy μ is defined as: $V^\mu(s) \triangleq E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)|s_0 = s, \mu\right]$.

In order to perform policy evaluation in a large state space, states are represented by a feature vector $\phi(s) \in \mathbb{R}^d$, and a linear function $\hat{v}(s) = \phi(s)^\top \theta$ is used to approximate the value function. The evaluation error is defined as $\|V(s) - \hat{v}(s)\|_{s\sim\pi}$, and π is a distribution over the state space $\mathcal{S}$. The evaluation error can be decomposed into approximation error and estimation error. In this paper, we will focus on the estimation error with linear function approximation.

As we know, the value function in RL satisfies the following Bellman equation: $V^\mu(s) = \mathbb{E}_{\mu,P}\left[R(s_t,a_t) + \gamma V^\mu(s_{t+1})|s_t = s\right] \triangleq T^\mu V^\mu(s)$, where T^μ is called Bellman operator for policy μ. Gradient-based TD (GTD) algorithms (including GTD and GTD2) proposed by Sutton et al. (2009a) and Sutton et al. (2009b) update the approximated value function by minimizing the objective function related to Bellman equation errors, i.e., the norm of the expected TD update (NEU) and mean-square projected Bellman error (MSPBE) respectively(Maei (2011),Liu et al. (2015)) ,

$$GTD: \quad J_{NEU}(\theta) = \|\Phi^\top K(T^\mu \hat{v} - \hat{v})\|^2 \tag{2.1}$$

$$GTD2: \quad J_{MSPBE}(\theta) = \|\hat{v} - \mathcal{P}T^\mu \hat{v}\| = \|\Phi^\top K(T^\mu \hat{v} - \hat{v})\|^2_{C^{-1}} \tag{2.2}$$

where K is a diagonal matrix whose elements are $\pi(s)$, $C = \mathbb{E}_\pi(\phi_i \phi_i^\top)$.

Actually, the two objective functions in GTD and GTD2 can be unified as below

$$J(\theta) = \|b - A\theta\|^2_{M^{-1}}, \tag{2.3}$$

where $M = I$ in GTD, $M = C$, in GTD2, $A = \mathbb{E}_\pi[\rho(s,a)\phi(s)(\phi(s) - \gamma\phi(s'))^\top], b = \mathbb{E}_\pi[\rho(s,a)\phi(s)r], \rho(s,a) = \mu(a|s)/\mu_b(a|s)$ is the importance weighting factor.

Denote the set of training samples as $\mathcal{D}$. $\mathcal{D} = \{\xi_i = (s_i, a_i, r_i, s'_i)\}_{i=1}^n$, s_i sampled from a Markov chain rather than i.i.d from one distribution, $a_i \sim \pi_b(\cdot|s_i)$,$s'_i \sim P^a_{s,s'}(\cdot|s_i,a_i)$, $r_i = R(s_i,a_i,s'_i)$.

Since the underlying distribution is unknown, we use the data to estimate the value function by minimizing the empirical estimation error, i.e.,

$$\hat{J}(\theta) = \frac{1}{T}\sum_{i=1}^{T} \|\hat{b} - \hat{A}\theta\|^2_{\hat{M}^{-1}}$$

where $\hat{A}_i = \rho(s_i,a_i)\phi(s_i)(\phi(s_i) - \gamma\phi(s'_i))^\top, \hat{b}_i = \rho(s_i,a_i)\phi(s_i)r_i, \hat{C}_i = \phi(s_i)\phi(s_i)^\top$.

Liu et al. (2015) derived that the GTD algorithms to minimize (2.3) is equivalent to the stochastic gradient algorithms to solve the following convex-concave saddle point problem

$$\min_x \max_y \left(L(x,y) = \langle b - Ax, y\rangle - \frac{1}{2}\|y\|^2_M\right), \tag{2.4}$$

with x as the parameter θ in the value function ,y as the auxiliary variable used in GTD algorithms.

Algorithm 1 GTD Algorithms

1: **for** $t = 1, \dots, T$ **do**

2: Update parameters

$$y_{t+1} = P_{\mathcal{X}_y}\left(y_t + \alpha_t(\hat{b}_t - \hat{A}_t\theta_t - \hat{M}_t y_t)\right)$$
$$x_{t+1} = P_{\mathcal{X}_x(x_t + \alpha_t \hat{A}_t^\top y_t)}$$

3: **end for**

Output:

$$\tilde{x}_n = \frac{\sum_{t=1}^T \alpha_t x_t}{\sum_{t=1}^T \alpha_t} \qquad \tilde{y}_n = \frac{\sum_{t=1}^T \alpha_t y_t}{\sum_{t=1}^T \alpha_t} \tag{2.5}$$

Therefore, we consider the general convex-concave stochastic saddle point problem as below

$$\min_{x\in\mathcal{X}_x} \max_{y\in\mathcal{X}_y} \{\phi(x, y) = \mathbb{E}_\xi[\Phi(x, y, \xi)]\}, \tag{2.6}$$

where $\mathcal{X}_x \subset \mathbb{R}^n$ and $\mathcal{X}_y \subset \mathbb{R}^m$ are bounded closed convex sets, $\xi \in \Xi$ is random variable and its distribution is $\Pi(\xi)$, and the expected function $\phi(x, y)$ is convex in x and concave in y. Denote $z = (x, y) \in \mathcal{X}_x \times \mathcal{X}_y \triangleq \mathcal{X}$, the gradient of $\phi(z)$ as $g(z)$, and the gradient of $\Phi(z, \xi)$ as $G(z, \xi)$.

In the stochastic gradient algorithm, the model is updated as:

$$z_{t+1} = P_{\mathcal{X}}(z_t - \alpha_t(G(x_t, y_t, \xi_t))) \tag{2.7}$$

Here, $P_{\mathcal{X}}$ are projection operators. We use projection operator to keep parameters remain in bounded convex set. α_t is the step-size.

After T iterations, we get the model $\tilde{z}_1^T = \frac{\sum_{t=1}^T \alpha_t z_t}{\sum_{t=1}^T \alpha_t}$. The error of the model $\tilde{z}_1^T$ is measured by the primal-dual gap error

$$Err_\phi(\tilde{z}_1^T) = \max_{y\in\mathcal{X}_y} \phi(\tilde{x}_1^T, y) - \min_{x\in\mathcal{X}_x} \phi(x, \tilde{y}_1^T). \tag{2.8}$$

Liu et al. (2015) proved that the estimation error of the GTD algorithms can be upper bounded by their corresponding primal-dual gap error multiply a factor. Therefore, we are going to derive the finite sample primal-dual gap error bound for the convex-concave saddle point problem firstly, and then extend it to the finite sample estimation error bound for the GTD algorithms.

Details of GTD algorithms used to optimize (2.4) are placed in **Algorithm 1**(Liu et al. (2015)).

2.2 Related work

The TD algorithms for policy evaluation can be divided into two categories: gradient based methods and least-square(LS) based methods(Dann et al. (2014)). Since LS based algorithms need $\mathcal{O}(d^2)$ storage and computational complexity while GTD algorithms are both of $\mathcal{O}(d)$ complexity, gradient based algorithms are more commonly used when the feature dimension is large. Thus, in this paper, we focus on GTD algorithms.

Sutton et al. (2009a) proposed the gradient-based temporal difference (GTD) algorithm for off-policy policy evaluation problem with linear function approximation. Sutton et al. (2009b) proposed GTD2 algorithm which shows a faster convergence in practice. Liu et al. (2015) connected GTD algorithms to a convex-concave saddle point problem and derive a finite sample bound in both on-policy and off-policy cases for constant step size in i.i.d. setting.

In the realistic Markov setting, although the finite sample bounds for LS-based algorithms have been proved Lazaric et al. (2012) Tagorti and Scherrer (2015) LSTD(λ), to the best of our knowledge, there is no previous finite sample analysis work for GTD algorithms.

3. Main Theory

In this section, we will present our main results. In Theorem 2 and Theorem 3, we present our error bound for the general convex-concave saddle point problem in expectation and with high probability respectively; in Theorem 5, we provide the finite sample bounds for GTD algorithms in both on-policy and off-policy cases. We only give proof sketches in the main paper due to space limitation. Please check the complete proofs in the supplementary materials.

Our results are derived based on the following common assumptions(Nemirovski (2004), Duchi et al. (2012), Liu et al. (2015)). Please note that, assumption 4 in RL can guarantee the Lipschitz and smooth properties in assumption 5-6 (Please see Propsition 4).

Assumption 1 (Bounded parameter space) *We assume there are finite $D < \infty$ such that*

$$\|z - z'\| \leq D \quad for\ z, z' \in \mathcal{X}.$$

Assumption 2 (Step size) *Let $\{\alpha_t\}$ denote step size sequence which is non-increasing and for $\forall t \quad 0 < \alpha_t \leq \alpha_0 \leq \infty$.*

Assumption 3 (Problem solvable) *The matrix A and C are non-singular.*

Assumption 4 (Bounded data) *The max norm of features are bounded by L, rewards are bounded by R_{max} and importance weights are bounded by ρ_{max}.*

Assumption 5 (Lipschitz) *For Π-almost every ξ , the function $\Phi(x, y, \xi)$ is Lipschitz for both x and y, that is there exists three constant $0 < L_{1x} < \infty$,$0 < L_{1y} < \infty$, $0 < L_1 < \infty$ such that:*

$$|\Phi(x', y, \xi) - \Phi(x, y, \xi)| \leq L_{1x}\|x - x'\| \quad for \quad \forall x, x' \in \mathcal{X}_x$$

$$|\Phi(x, y, \xi) - \Phi(x, y', \xi)| \leq L_{1y}\|y - y'\| \quad for \quad \forall y, y' \in \mathcal{X}_y$$

and let $L_1 \triangleq \sqrt{2}\sqrt{L_{1x}^2 + L_{1y}^2}$, *we have*

$$|\Phi(z,\xi) - \Phi(z',\xi)| \leq L_1\|z - z'\| \quad for \quad \forall z, z' \in \mathcal{X}_x \times \mathcal{X}_y.$$

Assumption 6 (Smooth) *For* Π*-almost every* ξ *, the partial gradient function of* $\Phi(x, y, \xi)$ *is Lipschitz, that is there exists three constant* $0 < L_{2x} < \infty$,$0 < L_{2y} < \infty$, $0 < L_2 < \infty$ *such that:*

$$\|G_x(x', y, \xi) - G_x(x, y, \xi)\| \leq L_{2x}\|x - x'\| \quad for \quad \forall x, x' \in \mathcal{X}_x, \forall y \in \mathcal{X}_y$$

$$\|G_y(x, y, \xi) - G_y(x, y', \xi)\| \leq L_{2y}\|y - y'\| \quad for \quad \forall y, y' \in \mathcal{X}_y, \forall x \in \mathcal{X}_x$$

Then, let $L_2 \triangleq \sqrt{2}\sqrt{L_{2x}^2 + L_{2y}^2}$, *we have*

$$\|G(z,\xi) - G(z',\xi)\| \leq L_2\|z - z'\| \quad for \quad \forall z, z' \in \mathcal{X}_x \times \mathcal{X}_y.$$

In the Markov setting, we also make a common assumption on the mixing time. Following the notation of Duchi et al. (2012), we denote the conditional probability distribution $P(\xi_t \in A|\mathcal{F}_s)$ as $P_{[s]}^t(A)$ and the corresponding probability density as $p_{[s]}^t$. Similarly, we denote the stationary distribution of the data generating stochastic process as Π and its density as π.

Definition 1 *The mixing time* $\tau(P_{[t]}, \eta)$ *of the sampling distribution* P *conditioned on the* $\sigma-$*field of the initial* t *sample* $\mathcal{F}_t = \sigma(\xi_1, \ldots, \xi_t)$ *is defined as:* $\tau(P_{[t]}, \eta) \triangleq \inf \Big\{ \Delta : t \in \mathbb{N}, \int |p_{[t]}^{t+\Delta}(\xi) - \pi(\xi)| d(\xi) \leq$ *where* $p_{[t]}^{t+\Delta}$ *is the conditional probability density at time* $t + \Delta$, *given* $\mathcal{F}_t$.

Assumption 7 (Mixing time) *The mixing times of the stochastic process* $\{\xi_t\}$ *are uniform. i.e., there exists uniform mixing times* $\tau(P, \eta) \leq \infty$ *such that, with probability* 1, *we have* $\tau(P_{[s]}, \eta) \leq \tau(P, \eta)$ *for all* $\eta > 0$ *and* $s \in \mathbb{N}$.

Intuitively, the mixing time upper bound $\tau(P, \eta)$ measure the speed of stochastic process converge to its stationary distribution. We will simply denote $\tau(P, \eta)$ by $\tau(\eta)$. Please note that, any time-homogeneous Markov chain with finite state-space and any uniformly ergodic Markov chains with general state space satisfy the above assumption(Meyn and Tweedie (2012)).

3.1 Expectation error bound for convex-concave saddle point problem

Theorem 2 *Suppose Assumption 1,2,5,6 hold, then for the gradient algorithm optimizing the convex-concave saddle point problem in (2.6),* $\forall \eta > 0$ *such that* $\tau(\eta) \leq T/2$, *we have*

$$\mathbb{E}_{\mathcal{D}}[Err_\phi(\tilde{z}_1^T)] \leq \frac{1}{\sum_{t=1}^T \alpha_t}\left[A + B\sum_{t=1}^T \alpha_t^2 + C\tau(\eta)\sum_{t=1}^T \alpha_t^2 + F\eta\sum_{t=1}^T \alpha_t + H\tau(\eta)\right], \forall \eta > 0,$$

$$where: \; A = D^2 \qquad B = \frac{5}{2}L_1^2 \qquad C = 6L_1^2 + 2L_1L_2D \qquad F = 2L_1D \qquad H = 6L_1D\alpha_0$$

Remark: (1) With $T \to \infty$, the error bound approaches 0 in order $O(\frac{\sum_{t=1}^{T}\alpha_t^2}{\sum_{t=1}^{T}\alpha_t})$. (2) The mixing time $\tau(\eta)$ will influence the convergence rate. If the Markov process has better mixing property with smaller $\tau(\eta)$, the algorithm converge faster. (3) If the data are i.i.d. generated (i.e., the mixing time $\tau(\eta) = 0, \forall \eta$) and the step size is set to the constant $\frac{c}{\sqrt{T}}$, our error bound will reduce to $\mathbb{E}[Err_\phi(\tilde{z}_1^T)] \leq \frac{1}{\sum_{t=1}^{T}\alpha_t}\left[A + B\sum_{t=1}^{T}\alpha_t^2\right] = \mathcal{O}(\frac{L_1^2}{\sqrt{T}})$, which is identical to previous work with constant step size in i.i.d. setting (Liu et al. (2015),Nemirovski et al. (2009)).

Proof By the definition of the error function in (2.8) and the property that $\phi(x, y)$ is convex for x and concave for y, the expected error can be bounded as below

$$\mathbb{E}\left[Err_\phi(\tilde{z}_1^T)\right] \leq \mathbb{E}\left[\max_z \frac{1}{\sum_{t=1}^{T}\alpha_t}\sum_{t=1}^{T}\alpha_t\left[(z_t - z)^\top g(z_t)\right]\right].$$

Denote $\delta_t \triangleq g(z_t) - G(z_t, \xi_t)$, $\delta_t' \triangleq g(z_t) - G(z_t, \xi_{t+\tau})$, $\delta_t'' \triangleq G(z_t, \xi_{t+\tau}) - G(z_t, \xi_t)$. Constructing $\{v_t\}_{t\geq 1}$ which is measurable with respect to $\mathcal{F}_{t-1}$,$v_{t+1} = P_{\mathcal{X}}\big(v_t - \alpha_t(g(z_t) - G(z_t, \xi_t))\big)$. We have the following key decomposition to the right hand side in the above inequality, the initiation and the explanation for such decomposition is placed in supplementary materials. For $\forall \tau \geq 0$:

$$\begin{aligned}\mathbb{E}\left[\max_z \sum_{t=1}^{T}\alpha_t\left[(z_t - z)^\top g(z_t)\right]\right] = \mathbb{E}\max_z\Bigg[\sum_{t=1}^{T-\tau}\alpha_t\Big[&\underbrace{(z_t - z)^\top G(z_t, \xi_t)}_{(a)} + \underbrace{(z_t - v_t)^\top \delta_t'}_{(b)} \\ &+ \underbrace{(z_t - v_t)^\top \delta_t''}_{(c)} + \underbrace{(v_t - z)^\top \delta_t}_{(d)}\Big] + \underbrace{\sum_{t=T-\tau+1}^{T}\alpha_t\left[(z_t - z)^\top g(z_t)\right]\Bigg]}_{(e)}.\end{aligned} \tag{3.1}$$

For term(a), we split $G(z_t, \xi_t)$ into three terms by the definition of $\mathcal{L}_2$-norm and the iteration formula of z_t, and then we bound its summation by $\sum_{t=1}^{T-\tau}\left(\|\alpha_t G(z_t, \xi_t)\|^2 + \|z_t - z\|^2 - \|z_{t+1} - z\|^2\right)$. Actually, in the summation, the last two terms will be eliminated except for their first and the last terms. Swap the max and $\sum$ operators and use the Lipschitz Assumption 5, the first term can be bounded. For term (b), since $(z_t - v_t)$ is not related to max operator and it is measurable with respect to $\mathcal{F}_{t-1}$, we can bound Term (b) through the definition of mixing time. Term (c) includes the sum of $G(z_t, \xi_{t+\tau}) - G(z_t, \xi_t)$, which is might be large in Markov setting. We reformulate it into the sum of $G(z_{t-\tau}, \xi_t) - G(z_t, \xi_t)$ and use the smooth Assumption 6 to bound it. Term (d) is similar to term (a) except that $g(z_t) - G(z_t, \xi_t)$ is the gradient that used to update v_t. We can bound it similarly with term (a). Term(e) is a constant that does not change much with $T \to \infty$, and we can bound it directly through upper bound of each of its own terms. Finally, we combine all the upper bounds to each term, use the mixing time Assumption 7 to choose $\tau = \tau(\eta)$ and obtain the error bound in Theorem 1. ■

3.2 High probability error bound for convex-concave saddle point problem

Theorem 3 *Suppose Assumption 1,2,5,6 hold. For $\forall\delta > 0$, and $\forall\eta > 0$ such that $\tau(\eta) \leq T/2$, with probability at least $1-\delta$, we have*

$$Err_\phi(\tilde{z}_1^T) \leq \frac{1}{\sum_{t=1}^{T} \alpha_t}\left[A + B\sum_{t=1}^{T}\alpha_t^2 + C\tau(\eta)\sum_{t=1}^{T}\alpha_t^2 + F\eta\sum_{t=1}^{T}\alpha_t + H\tau(\eta) \right.$$
$$\left. + 8DL_1\sqrt{2\tau(\eta)\log\frac{\tau(\eta)}{\delta}\left(\sum_{t=1}^{T}\alpha_t^2 + \tau(\eta)\alpha_0\right)}\right]$$

Remark: The above high probability error bound is similar to the expectation error bound in Theorem 2 except for the last term. This is because we have to take the deviation of the data around its expectation into account in the high probability bound.
Proof We start from the key decomposition (3.1), and bound each term without expectation this time. We can easily bound each term as previously except for Term (b). We decompose Term(b) into a martingale part and an expectation part.By constructing a martingale difference sequence and using the Azuma's inequality together with the Assumption 7, we can bound Term (b) and finally obtain the high probability error bound. ■

3.3 Finite Sample Bounds for GTD Algorithms

As a specific convex-concave saddle point problem, the error bounds in Theorem 1&2 can also provide the error bounds for GTD with the following specifications for the Lipschitz constants.

Proposition 4 *Suppose Assumption 1-4 hold, then the objective function in GTD algorithms is Lipschitz and smooth with the following coefficients:*

$$L_1 \leq \sqrt{2}(2D(1+\gamma)\rho_{max}L^2d + \rho_{max}LR_{max} + \lambda_M)$$
$$L_2 \leq \sqrt{2}(2(1+\gamma)\rho_{max}L^2d + \lambda_M)$$

where λ_M is the largest singular value of M.

Theorem 5 *Suppose assumptions 1-4 hold, then we have the following finite sample bounds for the error $\|V - \tilde{v}_1^T\|_\pi$ in GTD algorithms: In on-policy case, the bound in expectation is $\mathcal{O}\left(\frac{L\sqrt{L^4d^3\lambda_M\pi_{max}(1+\tau(\eta))\pi_{max}o_1(T)}}{\nu_C}\right)$ and with probability $1-\delta$ is $\mathcal{O}\left(\frac{\sqrt{L^4d^2\lambda_M\pi_{max}}}{\nu_C}\left(\sqrt{(1+\tau(\eta))L^2do_1(T) + \sqrt{\tau}\right.\right.$; In off-policy case, the bound in expectation is $\mathcal{O}\left(\frac{L^2d\sqrt{2\lambda_C\lambda_M\pi_{max}(1+\tau(\eta))o_1(T)}}{\nu_{(A^TM^{-1}A)}}\right)$ and with probability $1-\delta$ is $\mathcal{O}\left(\frac{\sqrt{2\lambda_C\lambda_M\pi_{max}}}{\nu_{(A^TM^{-1}A)}}\left(\sqrt{L^4d^2(1+\tau(\eta))o_1(T) + \sqrt{\tau(\eta)\log(\frac{\tau(\eta)}{\delta})}o_2(T)}\right)\right)$,* *where* $\nu_C, \nu_{(A^TM^{-1}A)}$ *is the smallest eigenvalue of the* C *and* $A^TM^{-1}A$ *respectively ,* λ_C *is the largest singular value of* C*,* $o_1(T) = (\frac{\sum_{t=1}^T \alpha_t^2}{\sum_{t=1}^T \alpha_t}), o_2(T) = (\frac{\sqrt{\sum_{t=1}^T \alpha_t^2}}{\sum_{t=1}^T \alpha_t})$.

We would like to make the following discussions for Theorem 5 .

The GTD algorithms do converge in the realistic Markov setting. As shown in Theorem 5, the bound in expectation is $\mathcal{O}\left(\sqrt{(1+\tau(\eta))o_1(T)}\right)$ and with probability $1-\delta$ is $\mathcal{O}\left(\sqrt{(1+\tau(\eta))o_1(T) + \sqrt{\tau(\eta)\log(\frac{\tau(\eta)}{\delta})}o_2(T)}\right)$. If the step size α_t makes $o_1(T) \to 0$ and $o_2(T) \to 0$, as $T \to \infty$, the GTD algorithms will converge. Additionally, in high probability bound, if $\sum_{t=1}^T \alpha_t^2 > 1$, then $o_1(T)$ dominates the order, if $\sum_{t=1}^T \alpha_t^2 < 1$, $o_2(T)$ dominates the order.

The setup of the step size can be flexible. Our finite sample bounds for GTD algorithms converge to 0 if the step size satisfies $\sum_{t=1}^T \alpha_t \to \infty, \frac{\sum_{t=1}^T \alpha_t^2}{\sum_{t=1}^T \alpha_t} < \infty$, as $T \to \infty$. This condition on step size is much weaker than the constant step size in previous work Liu et al. (2015) , and the common-used step size $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}}), \alpha_t = \mathcal{O}(\frac{1}{t}), \alpha_t = c = \mathcal{O}(\frac{1}{\sqrt{T}})$ all satisfy the condition. To be specific, for $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}})$, the convergence rate is $\mathcal{O}(\frac{\ln(T)}{\sqrt{T}})$; for $\alpha_t = \mathcal{O}(\frac{1}{t})$, the convergence rate is $\mathcal{O}(\frac{1}{\ln(T)})$, for the constant step size, the optimal setup is $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{T}})$ considering the trade off between $o_1(T)$ and $o_2(T)$, and the convergence rate is $\mathcal{O}(\frac{1}{\sqrt{T}})$.

The mixing time matters. If the data are generated from a Markov process with smaller mixing time, the error bound will be smaller, and we just need fewer samples to achieve a fixed estimation error. This finding can explain why the experience reply trick (Lin (1993)) works. With experience reply, we store the agent's experiences (or data samples) at each step, and randomly sample one from the pool of stored samples to update the policy function. By Theorem 1.19 - 1.23 of Durrett (2016), it can be proved that, for arbitrary $\eta > 0$, there exists t_0, such that $\forall t > t_0 \max_i |\frac{N_t(i)}{t} - \pi(i)| \leq \eta$. That is to say, when the size of the stored samples is larger than t_0, the mixing time of the new data process with experience replay is 0. Thus, the experience reply trick improves the mixing property of the data process, and hence improves the convergence rate.

Other factors that influence the finite sample bound: (1) With the increasing of the feature norm L, the finite sample bound increase. This is consistent with the empirical finding by Dann et al. (2014) that the normalization of features is crucial for the estimation quality of GTD algorithms. (2) With the increasing of the feature dimension d, the bound increase. Intuitively, we need more samples for a linear approximation in a higher dimension feature space.

4. Experiments

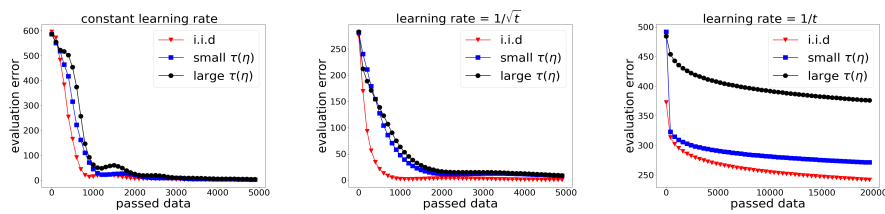
In this section, we report our simulation results to validate our theoretical findings. We consider the general convex-concave saddle problem,

$$\min_x \max_y \left(L(x,y) = \langle b - Ax, y \rangle + \frac{1}{2}\|x\|^2 - \frac{1}{2}\|y\|^2 \right) \tag{4.1}$$

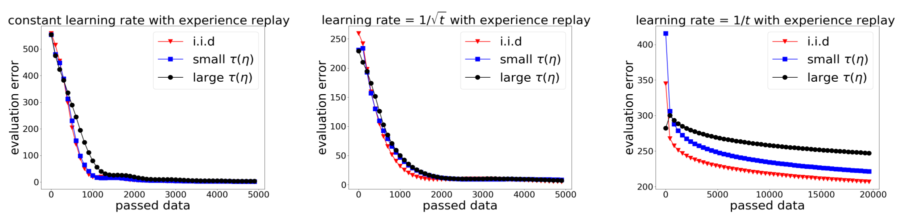
where A is a $n \times n$ matrix, b is a $n \times 1$ vector, Here we set $n = 10$. We conduct three experiment and set the step size to $\alpha_t = c = 0.001$, $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}}) = \frac{0.015}{\sqrt{t}}$ and $\alpha_t = \mathcal{O}(\frac{1}{t}) = \frac{0.03}{t}$ respectively. In each experiment we sample the data $\hat{A}, \hat{b}$ three ways : sample from two Markov chains with different mixing time but share the same stationary distribution or sample from stationary distribution i.i.d. directly. We sample $\hat{A}$ and $\hat{b}$ from Markov chain by using MCMC Metropolis-Hastings algorithms. Specifically, notice that the mixing time of a Markov chain is positive correlation with the second largest eigenvalue of its transition probability matrix (Levin et al. (2009)), we firstly conduct two transition probability matrix with different second largest eigenvalues(both with 1001 state and the second largest eigenvalue are 0.634 and 0.31 respectively), then using Metropolis-Hastings algorithms construct two Markov chain with same stationary distribution.

We run the gradient algorithm for the objective in (4.1) based on the simulation data, without and with experience reply trick. The primal-dual gap error curves are plotted in Figure 1.

We have the following observations. (1) The error curves converge in Markov setting with all the three setups of the step size. (2) The error curves with the data generated from the process which has small mixing time converge faster. The error curve for i.i.d. generated data converge fastest. (3) The error curve for different step size convergence at different rate. (4) With experience replay trick, the error curves in the Markov settings converge faster than previously. All these observations are consistent with our theoretical findings.



(a) $\alpha_t = c$ (b) $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}})$ (c) $\alpha_t = \mathcal{O}(\frac{1}{t})$



(d) $\alpha_t = c$ with trick (e) $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}})$ with trick (f) $\alpha_t = \mathcal{O}(\frac{1}{t})$ with trick

Figure 1: Experimental Results

5. Conclusion

In this paper, in the more realistic Markov setting, we proved the finite sample bound for the convex-concave saddle problems in expectation and with high probability. Then, we obtain the finite sample bound for GTD algorithms both in on-policy and off-policy, considering that the GTD algorithms are specific convex-concave saddle point problems. Our finite sample bounds provide important theoretical guarantee to the GTD algorithms, and also insights to improve them, including how to setup the step size and we need to improve the mixing property of the data like experience replay. In the future, we will study the finite sample bounds for policy evaluation with nonlinear function approximation.

Appendix A.

In this appendix we prove the following theorem from Section 6.2:

Theorem *Let u, v, w be discrete variables such that v, w do not co-occur with u (i.e., $u \neq 0 \Rightarrow v = w = 0$ in a given dataset $\mathcal{D}$). Let N_{v0}, N_{w0} be the number of data points for which $v = 0, w = 0$ respectively, and let I_{uv}, I_{uw} be the respective empirical mutual information values based on the sample $\mathcal{D}$. Then*

$$N_{v0} \; > \; N_{w0} \;\; \Rightarrow \;\; I_{uv} \; \leq \; I_{uw}$$

with equality only if u is identically 0. ■

Proof. We use the notation:

$$P_v(i) \; = \; \frac{N_v^i}{N}, \quad i \neq 0; \quad P_{v0} \; \equiv \; P_v(0) \; = \; 1 - \sum_{i \neq 0} P_v(i).$$

These values represent the (empirical) probabilities of v taking value $i \neq 0$ and 0 respectively. Entropies will be denoted by H. We aim to show that $\frac{\partial I_{uv}}{\partial P_{v0}} < 0$....

Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

Shalabh Bhatnagar, Doina Precup, David Silver, Richard S Sutton, Hamid R Maei, and Csaba Szepesvári. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems*, pages 1204–1212, 2009.

Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: a survey and comparison. *Journal of Machine Learning Research*, 15(1): 809–883, 2014.

John C Duchi, Alekh Agarwal, Mikael Johansson, and Michael I Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

Richard Durrett. Poisson processes. In *Essentials of Stochastic Processes*, pages 95–124. Springer, 2016.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13(Oct):3041–3074, 2012.

David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.

Long-Ji Lin. *Reinforcement learning for robots using neural networks*. PhD thesis, Fujitsu Laboratories Ltd, 1993.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *UAI*, pages 504–513. Citeseer, 2015.

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

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 387–395, 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

Richard S Sutton, Hamid R Maei, and Csaba Szepesvári. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, pages 1609–1616, 2009a.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009b.

Manel Tagorti and Bruno Scherrer. On the rate of convergence and error bounds for lstd (*lambda*). In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1521–1529, 2015.

John N Tsitsiklis, Benjamin Van Roy, et al. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997.

H Yu. On convergence of emphatic temporal-difference learning. In *Proceedings of The 28th Conference on Learning Theory*, pages 1724–1751, 2015.