Finite Sample Analysis of LSTD with Random Projections and Eligibility Traces

Haifang Li[1], **Yingce Xia**[2] and **Wensheng Zhang**[1]
[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Science and Technology of China, Hefei, Anhui, China
haifang.li@ia.ac.cn, yingce.xia@gmail.com, wensheng.zhang@ia.ac.cn

Abstract

Policy evaluation with linear function approximation is an important problem in reinforcement learning. When facing high-dimensional feature spaces, such a problem becomes extremely hard considering the computation efficiency and quality of approximations. We propose a new algorithm, LSTD(λ)-RP, which leverages random projection techniques and takes eligibility traces into consideration to tackle the above two challenges. We carry out theoretical analysis of LSTD(λ)-RP, and provide meaningful upper bounds of the estimation error, approximation error and total generalization error. These results demonstrate that LSTD(λ)-RP can benefit from random projection and eligibility traces strategies, and LSTD(λ)-RP can achieve better performances than prior LSTD-RP and LSTD(λ) algorithms.

1 Introduction

Policy evaluation, commonly referred to as value function approximation, is an important and central part in many reinforcement learning (RL) algorithms [Sutton and Barto, 1998], whose task is to estimate value functions for a fixed policy in a discounted Markov Decision Process (MDP) environment. The value function of each state specifies the accumulated reward an agent would receive in the future by following the fixed policy from that state. Value functions have been widely investigated in RL applications, and it can provide insightful and important information for the agent to obtain an optimal policy, such as important board configurations in Go [Silver *et al.*, 2007], failure probabilities of large telecommunication networks [Frank *et al.*, 2008], taxi-out times at large airports [Balakrishna *et al.*, 2010] and so on.

Despite the value functions can be approximated by different ways, the simplest form, linear approximations, are still widely adopted and studied due to their good generalization abilities, relatively efficient computation and solid theoretical guarantees [Sutton and Barto, 1998; Dann *et al.*, 2014; Geist and Scherrer, 2014; Liang *et al.*, 2016]. Temporal Difference (TD) learning is a common approach to this policy evaluation with linear function approximation problem [Sutton and Barto, 1998]. These typical TD algorithms can be divided into two categories: gradient based methods (e.g., GTD(λ) [Sutton *et al.*, 2009]) and least-square (LS) based methods (e.g., LSTD(λ) [Boyan, 2002]). A good survey on these algorithms can be found in [Maei, 2011; Dann, 2012; Geist and Pietquin, 2013; Dann *et al.*, 2014; Geist and Scherrer, 2014].

As the development of information technologies, high-dimensional data is widely seen in RL applications [Sutton, 1996; Tedrake *et al.*, 2004; Riedmiller and Gabel, 2007] , which brings serious challenges to design scalable and computationally efficient algorithms for the linear value function approximation problem. To address this practical issue, several approaches have been developed for efficient and effective value function approximation. Kolter and Ng (2009) and Farahmand and Szepesvari (2011) adopted l_1 or l_2 regularization techniques to control the complexity of the large function space and designed several l_1 and l_2 regularized RL algorithms. Gehring *et al.* (2016) studied this problem by using low-rank approximation via an incremental singular value function decomposition and proposed t-LSTD(λ). Pan *et al.* (2017b) derived ATD(λ) by combining the low-rank approximation and quasi-Newton gradient descent ideas.

Recently, Ghavamzadeh *et al.* (2010) and Pan *et al.* (2017a) investigated sketching (projecting) methods to reduce the dimensionality in order to make it feasible to employ Least-Squares Temporal Difference (briefly, LSTD) algorithms. Specifically, Ghavamzadeh *et al.* (2010) proposed an approach named LSTD-RP, which is based on random projections. They showed that LSTD-RP can benefit from the random projection strategy. The eligibility traces have already been proven to be important parameters to control the quality of approximation during the policy evaluation process, but Ghavamzadeh *et al.* (2010) did not take them into consideration. Pan *et al.* (2017a) empirically investigated the effective use of sketching methods including random projections, count sketch, combined sketch and hadamard sketch for value function approximation, but they did not provide any conclusion on finite sample analysis. However, finite sample analysis is important for these algorithms since it clearly demonstrates the effects of the number of samples, dimensionality of the function space and the other related parameters.

In this paper, we focus on exploring the utility of random projections and eligibility traces on LSTD algorithms to tackle the computation efficiency and quality of approximations challenges in the high-dimensional feature spaces setting. We also provide finite sample analysis to evaluate its performance. To the best of our knowledge, this is the first work that performs formal finite sample analysis of LSTD with random projections and eligibility traces. Our contributions can be summarized from the following two aspects:

Algorithm: By introducing random projections and eligibility traces, we propose a refined algorithm named *LSTD with Random Projections and Eligibility Traces* (denoted as LSTD(λ)-RP for short), where λ is the trace parameter of λ-return when considering eligibility traces. LSTD(λ)-RP algorithm consists of two steps: first, generate a low-dimensional linear feature space through random projections from the original high-dimensional feature space; then, apply LSTD(λ) to this generated low-dimensional feature space.

Theoretical Analysis: We perform theoretical analysis to evaluate the performance of LSTD(λ)-RP and provide its finite sample performance bounds, including the estimation error bound, approximation error bound and total error bound. The analysis of the prior works LSTD-RP and LSTD(λ) cannot directly apply to our setting, since (i) The analysis of LSTD-RP is based on a model of regression with Markov design, but it does not hold when we incorporate eligibility traces; (ii) Due to utilizing random projections, the analysis of LSTD(λ) cannot be directly used, especially the approximation error analysis. To tackle these challenges, we first prove the linear independence property can be preserved by random projections, which is important for our analysis. Second, we decompose the total error into two parts: estimation error and approximation error. Then we make analysis on any fixed random projection space, and bridge these error bounds between the fixed random projection space and any arbitrary random projection space by leveraging the norm and inner-product preservation properties of random projections, the relationship between the smallest eigenvalues of the Gram matrices in the original and randomly projected spaces and the Chernoff-Hoeffding inequality for stationary β-mixing sequence. What's more, our theoretical results show that

1) Compared to LSTD-RP, the parameter λ of eligibility traces illustrates a trade-off between the estimation error and approximation error for LSTD(λ)-RP. We could tune λ to select an optimal λ^* which could balance these two errors and obtain the smallest total error bound. Furthermore, for fixed sample n, optimal dimension of randomly projected space d^* in LSTD(λ)-RP is much smaller than that of LSTD-RP.
2) Compared to LSTD(λ), in addition to the computational gains which are the result of random projections, the estimation error of LSTD(λ)-RP is much smaller at the price of a controlled increase of the approximation error. LSTD(λ)-RP may have a better performance than LSTD(λ), whenever the additional term in the approximation error is smaller than the gain achieved in the estimation error.

These results demonstrate that LSTD(λ)-RP can benefit from eligibility traces and random projections strategies in computation efficiency and approximation quality, and can be superior to LSTD-RP and LSTD(λ) algorithms.

2 Background

In this section, first we introduce some notations and preliminaries. Then we make a brief review of LSTD(λ) and LSTD-RP algorithms.

Now we introduce some notations for the following paper. Let $|\cdot|$ denote the size of a set and $\|\cdot\|_2$ denote the L_2 norm for vectors. Let $\mathcal{X}$ be a measurable space. Denote $\mathcal{S}(\mathcal{X})$ the set of probability measure over $\mathcal{X}$, and denote the set of measurable functions defined on $\mathcal{X}$ and bounded by $L \in \mathbb{R}^+$ as $\mathcal{B}(\mathcal{X}, L)$. For a measure $\mu \in \mathcal{S}(\mathcal{X})$, the μ-weighted L_2 norm of a measurable function f is defined as $\|f\|_\mu = \sqrt{\sum_{x\in\mathcal{X}} f(x)^2\mu(x)}$. The operator norm for matrix W is defined as $\|W\|_\mu = \sup_{w\neq 0} \frac{\|Ww\|_\mu}{\|w\|_\mu}$.

2.1 Value Functions

Reinforcement learning (RL) is an approach to find optimal policies in sequential decision-making problems, in which the RL agent interacts with a stochastic environment formalized by a discounted *Markov Decision Process* (MDP) [Puterman, 2014]. An MDP is described as a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{P}^a_{xx'}, \mathcal{R}, \gamma)$, where state space $\mathcal{X}$ is finite [1], action space $\mathcal{A}$ is finite, $\mathcal{P}^a_{xx'}$ is the transition probability from state x to the next state x' when taking action a, $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function, which is uniformly bound by $R_{\max}$, and $\gamma \in (0, 1)$ is the discount factor. A deterministic policy[2] $\pi : \mathcal{X} \to \mathcal{A}$ is a mapping from state space to action space, which is an action selection policy. Given the policy π, the MDP $\mathcal{M}$ can be reduced to a Markov chain $\mathcal{M}^\pi = (\mathcal{X}, P^\pi, r^\pi, \gamma)$, with transition probability $P^\pi(\cdot|x) = P(\cdot|x, \pi(x))$ and reward $r^\pi(x) = \mathcal{R}(x, \pi(x))$.

In this paper, we are interested in policy evaluation, which can be used to find optimal policies or select actions. It involves computing the state-value function of a given policy which assigns to each state a measure of long-term performance following the given policy. Mathematically, given a policy π, for any state $x \in \mathcal{X}$, the value function of state x is defined as follows:

$$V^\pi(x) = \mathbb{E}_\pi[\textstyle\sum_{t=0}^\infty \gamma^t r(X_t)|X_0 = x],$$

where $\mathbb{E}_\pi$ denotes the expectation over random samples which are generated by following policy π. Let V^π denote a vector constructed by stacking the values of $V^\pi(1), ..., V^\pi(|\mathcal{X}|)$ on top of each other. Then, we can see that V^π is the unique fixed point of the Bellman operator T^π:

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

where R^π is the expected reward vector under policy π. Equation (1) is called Bellman Equation, which is the basis of temporal difference learning approaches. In the reminder

[1] For simplicity, we assume the state space is finite. However, the results in this paper can be generalized into other more general state spaces.

[2] Without loss of generality, here we only consider the deterministic policy. The extension to stochastic policy setting is straightforward.

of this paper, we omit the policy superscripts for ease of reference in unambiguous cases, since we are interested in on-policy learning in this work.

When the size of state space $|\mathcal{X}|$ is very large or even infinite, one may consider to approximate the state-value function by a linear function approximation, which is widely used in RL [Sutton and Barto, 1998; Dann *et al.*, 2014]. We define a linear function space $\mathcal{F}$, which is spanned by the basis functions $\phi_i : \mathcal{X} \to \mathbb{R}, i \in [D](D \ll |\mathcal{X}|)$[3], i.e., $\mathcal{F} = \{f_\alpha | f_\alpha(\cdot) = \phi(\cdot)^T\alpha, \alpha \in \mathbb{R}^D\}$, where $\phi(\cdot) = (\phi_1(\cdot), ..., \phi_D(\cdot))^T$ is the feature vector. We assume $\phi_i \in \mathcal{B}(\mathcal{X}, L), i \in [D]$ for some finite positive constant L. For any function $f_\alpha \in \mathcal{F}$, let $m(f_\alpha) := \|\alpha\|_2 \sup_{x\in\mathcal{X}} \|\phi(x)\|_2$. Furthermore, we generate a d-dimensional ($d < D$) random space $\mathcal{G}$ from $\mathcal{F}$ through random projections H, where $H \in \mathbb{R}^{d\times D}$ be a random matrix whose each element is drawn independently and identically distributed (i.i.d.) from Gaussion distribution $\mathcal{N}(0, 1/d)$[4]. For any $j \in [d]$, denote the randomly projected feature vector $\psi(\cdot) = (\psi_1(\cdot), ..., \psi_i(\cdot))^T$, where $\psi(\cdot) = H\phi(\cdot)$. Thus, $\mathcal{G} = \{g_\beta | g_\beta(\cdot) = \psi(\cdot)^T\beta, \beta \in \mathbb{R}^d\}$. Define $\Phi = (\phi(x))_{x\in\mathcal{X}} = (\phi_1, \ldots, \phi_D)$ of dimension $|\mathcal{X}| \times D$ and $\Psi = (\psi(x))_{x\in\mathcal{X}} = (\psi_1, \ldots, \psi_D)$ of dimension $|\mathcal{X}| \times d$ to be the original and randomly projected feature matrix respectively.

2.2 LSTD(λ)

Least-Squares Temporal Difference (LSTD) is a traditional and important approach for policy evaluation in RL, which was first introduced by [Bradtke and Barto, 1996], and later was extended to include the eligibility traces by [Boyan, 1999; Boyan, 2002] referred to as LSTD(λ).

The essence of LSTD(λ) is to estimate the fixed point of the projected multi-step Bellman equation, that is,

$$\begin{aligned} &V = \Pi_{\mathcal{F}} T^\lambda V, \\ &\text{where } V = \Phi\theta, \text{ and } \Pi_{\mathcal{F}} = \Phi(\Phi^T D_\mu \Phi)^{-1}\Phi^T D_\mu, \end{aligned} \tag{2}$$

where μ is the steady-state probabilities of the Markov chain $\mathcal{M}^\pi$ induced by policy π, D_μ denotes the diagonal matrix with diagonal elements being μ, $\Pi_{\mathcal{F}}$ is the orthogonal projection operator into the linear function space $\mathcal{F}$, and T^λ is a multi-step Bellman operator parameterized by $\lambda \in [0, 1]$, and it is defined as follows:

$$T^\lambda = (1-\lambda)\textstyle\sum_{i=0}^{\infty} \lambda^i T^{i+1}.$$

When $\lambda = 0$, we have $T^\lambda = T$, and it becomes LSTD.

Given one sampled trajectory $\{X_t\}_{t=1}^n$ generated by the Markov chain $\mathcal{M}^\pi$ under policy π, the LSTD(λ) algorithm returns $\hat{V}_{\text{LSTD}(\lambda)} = \Phi\tilde{\theta}$, with $\tilde{\theta} = \tilde{A}^{-1}\tilde{b}$, where

$$\begin{aligned} &\tilde{A} = \frac{1}{n-1}\sum_{i=1}^{n-1} \tilde{z}_i(\phi(X_i) - \gamma\phi(X_{i+1}))^T, \\ \text{and} \quad &\tilde{b} = \frac{1}{n-1}\sum_{i=1}^{n-1} \tilde{z}_i r(X_i), \end{aligned} \tag{3}$$

where $\tilde{z}_i = \sum_{k=1}^{i} (\lambda\gamma)^{i-k}\phi(X_k)$ is called the eligibility trace, and $\lambda \in [0, 1]$ is the trace parameter for the λ-return.

2.3 LSTD-RP

Compared to gradient based temporal difference (TD) learning algorithms, LSTD(λ) has data sample efficiency and parameter insensitivity advantages, but it is less computationally efficient. LSTD(λ) requires $O(D^3)$ computation per time step or still requires $O(D^2)$ by using the Sherman-Morrison formula to make incremental update. This expensive computation cost makes LSTD(λ) impractical for the high-dimensional feature spaces scenarios in RL. Recently, Least-Squares TD with Random Projections algorithm (briefly denoted as LSTD-RP) was proposed to deal with the high-dimensional data setting [Ghavamzadeh *et al.*, 2010].

The basic idea of LSTD-RP is to learn the value function of a given policy from a low-dimensional linear space $\mathcal{G}$ which is generated through random projections from a high-dimensional space $\mathcal{F}$. Their theoretical results show that the total computation complexity of LSTD-RP is $O(d^3 + ndD)$, which is dramatically less than the computation cost in the high dimensional space $\mathcal{F}$ (i.e., $O(D^3 + nD^2)$). In addition to these practical computational gains, Ghavamzadeh *et al.* (2010) demonstrate that LSTD-RP can provide an efficient and effective approximation for value functions, since LSTD-RP reduces the estimation error at the price of the increase in the approximation error which is controlled.

However, LSTD-RP does not take the eligibility traces into consideration, which are important parameters in RL. First, the use of these traces can significantly speed up learning by controlling the trade off between bias and variance [Att *et al.*, 2000; Sutton *et al.*, 2014]. Second, the parameter λ of these traces is also known to control the quality of approximation [Tsitsiklis *et al.*, 1997]. In the remainder of this paper, we present a generalization of LSTD-RP to deal with the $\lambda > 0$ scenario (i.e., LSTD(λ)-RP (see Section 3)). What's more, we also give its theoretical guarantee in Section 4.

3 Algorithm

In this section, we first consider the Bellman equation with random projections (see Equation (4)), and explore the existence and uniqueness properties of its solution, which is the goal of our newly proposed algorithm to estimate. Then we present the *LSTD with Random Projections and Eligibility Traces* algorithm (briefly denoted as LSTD(λ)-RP) as shown in Algorithm 1, and discuss its computational cost.

3.1 Bellman Equation with Random Projections

To begin with, we make the following assumption throughout the paper as [Tsitsiklis *et al.*, 1997; Tagorti and Scherrer, 2015].

Assumption 1. *The feature matrix Φ has full column rank; that is, the original high-dimensional feature vectors $(\phi_j)_{j\in\{1,\ldots,D\}}$ are linearly independent.*

From the following lemma, we can get that the linear independence property can be preserved by random projections. Due to the space restrictions, we leave its detailed proof into Appendix B in the full version of this work [Li *et al.*, 2018].

[3] $[D] = \{1, ..., D\}$.

[4] It is also can be some sub-Gaussian distributions. Without loss of generality, here we only consider Gaussian distribution for simplicity.

Lemma 1. *Let Assumption 1 hold. Then the randomly projected low-dimensional feature vectors $(\psi_j)_{j\in\{1,\dots,d\}}$ are linearly independent a.e.. Accordingly, $\Psi^T D_\mu \Psi$ is invertible a.e..*[5]

Let $\Pi_{\mathcal{G}}$ denote the orthogonal projection onto the randomly projected low-dimensional feature space $\mathcal{G}$ with respect to the μ-weighted L_2 norm. According to Lemma 1, we obtain the projection $\Pi_{\mathcal{G}}$ has the following closed form

$$\Pi_{\mathcal{G}} = \Psi(\Psi^T D_\mu \Psi)^{-1}\Psi^T D_\mu.$$

Then the projected multi-step *Bellman equation with random projections* becomes

$$\begin{aligned} V = \Pi_{\mathcal{G}} T^\lambda V,\ \lambda \in [0,1], \\ \text{where } T^\lambda = (1-\lambda)\sum\nolimits_{i=0}^{\infty} \lambda^i T^{i+1}. \end{aligned} \tag{4}$$

Note that when $\lambda = 0$, we have $T^\lambda = T$.

According to the Banach fixed point theorem, in order to guarantee the existence and uniqueness of the fixed point of Bellman equation with random projections (see Equation (4)), we only need to demonstrate the contraction property of operator $\Pi_{\mathcal{G}} T^\lambda$. By simple derivations, we can demonstrate that the contraction property of $\Pi_{\mathcal{G}} T^\lambda$ holds as shown in the following Lemma 2, and we leave its detailed proof into Appendix C of the full paper [Li *et al.*, 2018].

Lemma 2. *Let Assumption 1 hold. Then the projection operator $\Pi_{\mathcal{G}}$ is non-expansive w.r.t. μ-weighted quadratic norm, and the operator $\Pi_{\mathcal{G}} T^\lambda$ is a ($\frac{\gamma(1-\lambda)}{1-\gamma\lambda}$-)contraction.*

Denote the unique solution of the Bellman equation with random projections (see Equation (4)) as $V_{\text{LSTD}(\lambda)\text{-RP}}$. In this work, we focus exclusively on the linear function approximation problem. Therefore, there exists $\theta \in \mathbb{R}^d$ such that

$$V_{\text{LSTD}(\lambda)\text{-RP}} = \Psi\theta = \Pi_{\mathcal{G}} T^\lambda \Psi\theta. \tag{5}$$

Just as the derivations of LSTD(λ) algorithm [Tsitsiklis *et al.*, 1997; Sutton and Barto, 1998; Boyan, 2002], we can obtain that θ is a solution of the linear equation

$$\begin{aligned} A\theta = b, \\ \text{where } A = \Psi^T D_\mu (I-\gamma P)(I-\lambda\gamma P)^{-1}\Psi, \\ \text{and } b = \Psi^T D_\mu (I-\gamma\lambda P)^{-1} r. \end{aligned} \tag{6}$$

Furthermore, by Lemma 1, we can prove that A is invertible. Thus, $V_{\text{LSTD}(\lambda)\text{-RP}} = \Psi A^{-1} b$ is well defined.

3.2 LSTD(λ)-RP Algorithm

Now we present our proposed algorithm LSTD(λ)-RP in Algorithm 1, which aims to estimate the solution of Bellman equation with random projections (see Equation (6)) by using one sample trajectory $\{X_t\}_{t=1}^n$ generated by the Markov chain $\mathcal{M}^\pi$. Then we discuss its computational advantage compared to LSTD(λ) and LSTD-RP.

LSTD(λ)-RP algorithm is a generalization of LSTD-RP. It uses eligibility traces to handle the $\lambda > 0$ case. Line 8 updates the eligibility traces z, and lines 9-12 incrementally update A and b as described in Equation (8), which have some differences from that in LSTD-RP algorithm due to eligibility traces. If the parameter λ is set to zero, then the LSTD(λ)-RP algorithm becomes the original LSTD-RP algorithm. What's more, if the random projection matrix H is identity matrix, then LSTD(λ)-RP becomes LSTD(λ).

Algorithm 1: LSTD(λ)-RP Algorithm

```
1  Input: The original high-dimensional feature vector
       $\phi : \mathcal{X} \to \mathbb{R}^D$; discount factor $\gamma \in [0,1)$; eligibility
       trace parameter $\lambda \in [0,1]$; the sample trajectory
       $\{X_t, r_t\}_{t=1}^n$, where $X_t$ and $r_t$ are the observed state
       and reward received at time $t$ respectively;
2  Output: $\hat{\theta} := \hat{A}^{-1}\hat{b}$ or $\hat{\theta} := \hat{A}^{\dagger}\hat{b}$, where $\hat{A}^{\dagger}$ denote the
       Moore-Penrose pseudo-inverse of matrix $\hat{A}$;
3  Initialize: $\hat{A} \leftarrow 0, \hat{b} \leftarrow 0, z \leftarrow 0, t \leftarrow 0$;
4  Generate random projection matrix $H \in \mathbb{R}^{d\times D}$ whose
       elements are drawn i.i.d. from $\mathcal{N}(0, 1/d)$;
5  for $t = 0, 1, \dots, n$ do
6      $t \leftarrow t + 1$;
7      The randomly projected low-dimensional feature
           vector $\psi(X_t) = H\phi(X_t)$;
8      $z \leftarrow \lambda\gamma z + \psi(X_t)$;
9      $\Delta\hat{A} \leftarrow z(\psi(X_t) - \psi(X_{t+1}))^T$;
10     $\Delta\hat{b} \leftarrow z r_t$;
11     $\hat{A} \leftarrow \hat{A} + \frac{1}{t}[\Delta\hat{A} - \hat{A}]$;
12     $\hat{b} \leftarrow \hat{b} + \frac{1}{t}[\Delta\hat{b} - \hat{b}]$;
```

From Algorithm 1, we obtain that the LSTD(λ)-RP algorithm returns

$$\hat{V}_{\text{LSTD}(\lambda)\text{-RP}} = \Psi\hat{\theta}, \tag{7}$$

with $\hat{\theta} = \hat{A}^{-1}\hat{b}$,[6] where

$$\begin{aligned} \hat{A} &= \frac{1}{n-1}\sum\nolimits_{i=1}^{n-1} z_i(\psi(X_i) - \gamma\psi(X_{i+1}))^T, \\ \hat{b} &= \frac{1}{n-1}\sum\nolimits_{i=1}^{n-1} z_i r(X_i), \text{ and } z_i = \sum\nolimits_{k=1}^{i} (\lambda\gamma)^{i-k}\psi(X_k). \end{aligned} \tag{8}$$

Here z_i is referred to as *randomly projected eligibility trace*.

The difference between LSTD(λ)-RP algorithm and the prior LSTD-RP algorithm lies in the fact that LSTD(λ)-RP incorporates the eligibility traces. From Algorithm 1, we know that the computational cost of eligibility traces is $O(nd)$. Based on the analysis of the computational complexity of LSTD-RP algorithm, we obtain that the total computational complexity of LSTD(λ)-RP is $O(d^3 + ndD)(D \gg d)$. This reveals that the computation cost of LSTD(λ)-RP algorithm is much less than that of LSTD(λ) algorithm, which is $O(D^3 + nD^2)$ [Ghavamzadeh *et al.*, 2010].

To evaluate the performance of LSTD(λ)-RP algorithm, we consider the gap between the value function learned by LSTD(λ)-RP algorithm $\hat{V}_{\text{LSTD}(\lambda)\text{-RP}}$ and the true value function V, i.e., $\|\hat{V}_{\text{LSTD}(\lambda)\text{-RP}} - V\|_\mu$. We refer to this gap as

[5] Notice that here the randomness is w.r.t. the random projection rather than the random sample. In the following paper, without loss of generality, we can assume $(\psi_j)_{j\in\{1,\dots,d\}}$ are linearly independent and $\Psi^T D_\mu \Psi$ is invertible.

[6] We will see that $\hat{A}^{-1}$ exists with high probability for a sufficiently large sample size n in Theorem 3.

the *total error* of the LSTD(λ)-RP algorithm. According to the triangle inequality, we can decompose the total error into two parts: *estimation error* $\|\hat{V}_{\text{LSTD}(\lambda)\text{-RP}} - V_{\text{LSTD}(\lambda)\text{-RP}}\|_\mu$ and *approximation error* $\|V_{\text{LSTD}(\lambda)\text{-RP}} - V\|_\mu$. We will illustrate how to derive meaningful upper bounds for these three errors of LSTD(λ)-RP in the following section.

4 Theoretical Analysis

In this section, we conduct theoretical analysis for LSTD(λ)-RP. First, we examine the sample size needed to ensure the uniqueness of the sample-based LSTD(λ)-RP solution, that is, we explore sufficient conditions to guarantee the invertibility of $\hat{A}$ with high probability, which can be used in the analysis of estimation error bound. Second, we make finite sample analysis of LSTD(λ)-RP including discussing how to derive meaningful upper bounds for the estimation error $\|\hat{V}_{\text{LSTD}(\lambda)\text{-RP}} - V_{\text{LSTD}(\lambda)\text{-RP}}\|_\mu$, the approximation error $\|V_{\text{LSTD}(\lambda)\text{-RP}} - V\|_\mu$ and the total error $\|\hat{V}_{\text{LSTD}(\lambda)\text{-RP}} - V\|_\mu$.

To perform such finite sample analysis, we also need to make a common assumption on the Markov chain process $(X_t)_{t\geq 1}$ that has some β-mixing properties as shown in Assumption 2 [Mohri and Rostamizadeh, 2010; Tagorti and Scherrer, 2015]. Under this assumption, we can make full use of the concentration inequality for β-mixing sequences during the process of finite sample analysis.

Assumption 2. *$(X_t)_{t\geq 1}$ is a stationary exponential β-mixing sequence, that is, there exist some constant parameters $\beta_0 > 0$, $\beta_1 > 0$, and $\kappa > 0$ such that $\beta(m) \leq \beta_0 \exp(-\beta_1 m^\kappa)$.*

4.1 Uniqueness of the Sample-Based Solution

In this subsection, we explore how sufficiently large the number of observations n needed to guarantee the invertibility of $\hat{A}$ with high probability as shown in Theorem 3, which indicates the uniqueness of sample-based LSTD(λ)-RP solution. Due to the space limitations, we leave the detailed proof into Appendix D of the full paper [Li *et al.*, 2018].

Theorem 3. *Let Assumptions 1 and 2 hold, and $X_1 \sim \mu$. For any $\delta \in (0,1), \gamma \in (0,1)$, and $\lambda \in [0,1]$, let $n_0(\delta)$ be the smallest integer such that*

$$\frac{2dL^2}{(1-\gamma)\nu_F \eta(d,D,\delta/2)}\Bigg[\frac{2\xi(d,n,\delta/4)}{\sqrt{n-1}}\sqrt{(1+m_n^\lambda)I(n-1,\frac{\delta}{2})} + \frac{2\xi(d,n,\delta/4)}{n-1}m_n^\lambda + \frac{1}{(1-\lambda\gamma)(n-1)}\Bigg] < 1, \tag{9}$$

where

$$m_n^\lambda = \begin{cases} \lceil \frac{\log(n-1)}{\log\frac{1}{\lambda\gamma}} \rceil & \lambda \in (0,1] \\ 0 & \lambda = 0 \end{cases}, \xi(n,d,\delta) = 1 + \sqrt{\frac{8}{d}\log\frac{n}{\delta}},$$

$$\eta(d,D,\delta) = \big(1 - \sqrt{d/D} - \sqrt{2\log(2/\delta)/D}\big)^2,$$

$$I(n,\delta) = 32\Lambda(n,\delta)\max\{\Lambda(n,\delta)/\beta_1, 1\}^{\frac{1}{\kappa}},$$

$$\Lambda(n,\delta) = \log(8n^2/\delta) + \log(\max\{4e^2, n\beta_0\}),$$

and ν_F is the smallest eigenvalue of the Gram matrix $F = \Phi^T D_\mu \Phi$. Then when $D > d + 2\sqrt{2d\log(4/\delta)} + 2\log(4/\delta)$, with probability at least $1-\delta$ (the randomness w.r.t. the random sample and the random projection), we have, for all $n \geq n_0(\delta)$, $\hat{A}$ is invertible.

From Theorem 3, we can draw the following conclusions:

1) The number of observations needed to guarantee the uniqueness of the sample-based LSTD(λ)-RP solution is of order $\tilde{O}(d^2)$, and it is much smaller than that of LSTD(λ), which is of order $\tilde{O}(D^2)(D \gg d)$ (Theorem 1 in [Tagorti and Scherrer, 2015]).
2) In our analysis, setting $\lambda = 0$, we can see that our result has some differences from LSTD-RP (Lemma 3 in [Ghavamzadeh *et al.*, 2010]), since we consider the invertibility of the matrix $\hat{A}$, while they consider the empirical Gram matrix $\frac{1}{n}\Psi^T\Psi$.

Remark 1: According to Assumption 1, we know that $\nu_F > 0$. For all $\delta \in (0,1)$ and fixed d, $n_0(\delta)$ exists since the left hand side of Equation (9) tends to 0 when n tends to infinity.

4.2 Estimation Error Bound

In this subsection, we upper bound the estimation error of LSTD(λ)-RP as shown in Theorem 4. For its proof, first, bound the estimation error on one fixed randomly projected space $\mathcal{G}$. Then, by utilizing properties of random projections, the relationship between the smallest eigenvalues of the Gram matrices in $\mathcal{F}$ and $\mathcal{G}$ and the conditional expectation properties, bridge the error bounds between the fixed space and any arbitrary random projection space. Due to space limitations, we leave its detailed proof into Appendix E [Li *et al.*, 2018].

Theorem 4. *Let Assumptions 1 and 2 hold, and let $X_1 \sim \mu$. For any $\delta \in (0,1)$, $\gamma \in (0,1)$, and $\lambda \in [0,1]$, when $D > d + 2\sqrt{2d\log(4/\delta)} + 2\log(4/\delta)$ and $d \geq 15\log(4n/\delta)$, with probability $1-\delta$ (the randomness w.r.t. the random sample and the random projection), for all $n \geq n_0(\delta)$, the estimation error $\|V_{LSTD(\lambda)\text{-}RP} - \hat{V}_{LSTD(\lambda)\text{-}RP}\|_\mu$ is upper bounded as follows:*

$$\|V_{LSTD(\lambda)\text{-}RP} - \hat{V}_{LSTD(\lambda)\text{-}RP}\|_\mu \leq h(n,d,\delta) + \frac{4V_{\max}dL^2\xi(n,d,\delta/4)}{\sqrt{n-1}(1-\gamma)\nu_F\eta(d,D,\delta/2)}\sqrt{(m_n^\lambda+1)I(n-1,\delta/4)}, \tag{10}$$

with $h(n,d,\delta) = \tilde{O}(\frac{d}{n}\log\frac{1}{\delta})$, where $\nu_F(>0)$ is the smallest eigenvalue of the Gram matrix $\Phi^T D_u \Phi$, $V_{\max} = \frac{R_{\max}}{1-\gamma}$, $\xi(n,d,\delta)$, $\eta(d,D,\delta)$, m_n^λ, $I(n,\delta)$, and $n_0(\delta)$ are defined as in Theorem 3.

From Theorem 4, we have by setting $\lambda = 0$ in Equation (10), the estimation error bound of LSTD(λ)-RP becomes of order $\tilde{O}(d/\sqrt{n})$, and it is consistent with that of LSTD-RP (Theorem 2 in [Ghavamzadeh *et al.*, 2010]).

4.3 Approximation Error Bound

Now we upper bound the approximation error of LSTD(λ)-RP which is shown in Theorem 5. As to its proof, we first analyze the approximation error on any fixed random projected space $\mathcal{G}$. Then, we make a bridge of approximation error bound between the fixed random projection space and any arbitrary random projection space by leveraging the definition of projection and the inner-product preservation property of random projections and the Chernoff-Hoeffding inequality for stationary β-mixing sequence. Due to space limitations, we leave detailed proof into Appendix F of [Li *et al.*, 2018].

Theorem 5. *Let Assumptions 1 and 2 hold. Let* $X_1 \sim \mu$. *For any* $\delta \in (0,1), \gamma \in (0,1)$, *and* $\lambda \in [0,1]$, *when* $d \geq 15 \log(8n/\delta)$, *with probability at least* $1-\delta$ *(w.r.t. the random projection), the approximation error of LSTD(λ)-RP algorithm* $\|V - V_{\textit{LSTD}(\lambda)\textit{-RP}}\|_\mu$ *can be upper bounded as below,*

$$\begin{aligned}\|V - V_{\textit{LSTD}(\lambda)\textit{-RP}}\|_\mu \leq \frac{1-\lambda\gamma}{1-\gamma}\big[\|V - \Pi_{\mathcal{F}}V\|_\mu \\ +\sqrt{(8/d)\log(8n/\delta)}(1+\frac{2\sqrt{\Upsilon(n,\delta/2)}}{\sqrt{n}})m(\Pi_{\mathcal{F}}V)\big],\end{aligned} \tag{11}$$

where $\Upsilon(n,\delta) = (\log \frac{4+n\beta_0}{\delta})^{1+\frac{1}{\kappa}} \beta_1^{-\frac{1}{\kappa}}$.

From Theorem 5, we know that by setting $\lambda = 0$, the right hand of Equation (11) becomes $\frac{1}{1-\gamma}\big[\|V - \Pi_{\mathcal{F}}V\|_\mu + O\big(\sqrt{(1/d)\log(n/\delta)}m(\Pi_{\mathcal{F}}V)\big)\big]$, while for LSTD-RP (Theorem 2 in [Ghavamzadeh *et al.*, 2010]) it is $\frac{4\sqrt{2}}{\sqrt{1-\gamma^2}}\big[\|V - \Pi_{\mathcal{F}}V\|_\mu + O\big(\sqrt{(1/d)\log(n/\delta)}m(\Pi_{\mathcal{F}}V)\big)\big]$. Notice that they are just different from the coefficients. Furthermore, due to eligibility traces which can control the quality of approximation, we could tune λ to make approximation error of LSTD(λ)-RP smaller than that of LSTD-RP, since the coefficient in Equation (11) is $\frac{1-\lambda\gamma}{1-\gamma}$, while it is $\frac{4\sqrt{2}}{\sqrt{1-\gamma^2}}$ in LSTD-RP.

Remark 2: The coefficient $\frac{1-\lambda\gamma}{1-\gamma}$ in the approximation can be improved by $\frac{1-\lambda\gamma}{\sqrt{(1-\gamma)(1+\gamma-2\lambda\gamma)}}$ [Tsitsiklis *et al.*, 1997].

4.4 Total Error Bound

Combining Theorem 4 and Theorem 5, and by leveraging the triangle inequality, we can obtain the total error bound for LSTD(λ)-RP as shown in the following corollary.

Corollary 6. *Let Assumptions 1 and 2 hold. Let* $X_1 \sim \mu$. *For any* $\delta \in (0,1), \gamma \in (0,1)$, *and* $\lambda \in [0,1]$, *when* $D > d + 2\sqrt{2d\log(8/\delta)} + 2\log(8/\delta)$ *and* $d \geq 15\log(16n/\delta)$, *with probability (the randomness w.r.t. the random sample and the random projection) at least* $1-\delta$, *for all* $n \geq n_0(\delta)$, *the total error* $\|V - \hat{V}_{\textit{LSTD}(\lambda)\textit{-RP}}\|_\mu$ *can be upper bounded by:*

$$\begin{aligned}&\frac{4V_{\max}dL^2\xi(n,d,\delta/8)}{\sqrt{n-1}(1-\gamma)\nu_F\eta(d,D,\delta/4)}\sqrt{(m_n^\lambda+1)I(n-1,\delta/8)} \\ &+\frac{1-\lambda\gamma}{1-\gamma}\big[\|V - \Pi_{\mathcal{F}}V\|_\mu + \sqrt{(8/d)\log(16n/\delta)}(1+ \\ &(2/\sqrt{n})\sqrt{\Upsilon(n,\delta/4)})m(\Pi_{\mathcal{F}}V)\big] + h(n,d,\delta)\end{aligned} \tag{12}$$

with $h(n,d,\delta) = \tilde{O}(\frac{d}{n}\log\frac{1}{\delta})$, *where* $\nu_F(>0)$ *is the smallest eigenvalue of the Gram matrix* $\Phi^T D_u \Phi$, $V_{\max} = \frac{R_{\max}}{1-\gamma}$, $\xi(n,d,\delta)$, $\eta(d,D,\delta)$, m_n^λ, $I(n,\delta)$, $n_0(\delta)$ *are defined as in Theorem 3 and* $\Upsilon(n,\delta)$ *is defined as in Theorem 5.*

By setting $\lambda = 0$, the total error bound of LSTD(λ)-RP is consistent with that of LSTD-RP except for some differences in coefficients. These differences lie in the analysis of LSTD-RP based on a model of regression with Markov design.

Although our results consistent with LSTD-RP when setting $\lambda = 0$ except for some coefficients, our results have some advantages over LSTD-RP and LSTD(λ). Now we have some discussions. From Theorem 4, Theorem 5 and Corollary 6, we can obtain that

1) Compared to LSTD(λ), the estimation error of LSTD(λ)-RP is of order $\tilde{O}(d/\sqrt{n})$, which is much smaller than that of LSTD(λ) (i.e., $\tilde{O}(D/\sqrt{n})$ (Theorem 1 in [Tagorti and Scherrer, 2015])), since random projections can make the complexity of the projected space $\mathcal{G}$ is smaller than that of the original high-dimensional space $\mathcal{F}$. Furthermore, the approximation error of LSTD(λ)-RP increases by at most $O(\sqrt{(1/d)\log(n/\delta)}m(\Pi_{\mathcal{F}}V))$, which decreases w.r.t d. This shows that in addition to the computational gains, the estimation error of LSTD(λ)-RP is much smaller at the cost of a increase of the approximation error which can be fortunately controlled. Therefore, LSTD(λ)-RP may have a better performance than LSTD(λ), whenever the additional term in the approximation error is smaller than the gain achieved in the estimation error.
2) Compared to LSTD-RP, λ illustrates a trade-off between the estimation error and approximation error for LSTD(λ)-RP, since eligibility traces can control the trade off between the approximation bias and variance during the learning process. When λ increases, the estimation error would increase, while the approximation error would decrease. Thus, we could select an optimal λ^* to balance these two errors and obtain the smallest total error.
3) Compared to LSTD-RP, we can select an optimal $d^*_{\text{LSTD}(\lambda)\text{-RP}} = \tilde{O}(n\log n)^{\frac{1}{3}}$ to obtain the smallest total error, and make a balance between the estimation error and the approximation error of LSTD(λ)-RP, which is much smaller than that of LSTD-RP ($d^*_{\text{LSTD-RP}} = \tilde{O}(n\log n)^{\frac{1}{2}}$) due to the effect of eligibility traces.

These conclusions demonstrate that random projections and eligibility traces can improve the approximation quality and computation efficiency. Therefore, LSTD(λ)-RP can provide an efficient and effective approximation for value functions and can be superior to LSTD-RP and LSTD(λ).

Remark 3: Some discussions about the role of factor $m(\Pi_{\mathcal{F}}V)$ in the error bounds can be found in [Maillard and Munos, 2009] and [Ghavamzadeh *et al.*, 2010].

Remark 4: Our analysis can be simply generalized to the emphatic LSTD algorithm (ELSTD)[Yu, 2015] with random projections and eligibility traces.

5 Conclusion and Future Work

In this paper, we propose a new algorithm LSTD(λ)-RP, which leverages random projection techniques and takes eligibility traces into consideration to tackle the computation efficiency and quality of approximations challenges in the high-dimensional feature space scenario. We also make theoretical analysis for LSTD(λ)-RP.

For the future work, there are still many important and interesting directions: (1) the convergence analysis of the off-policy learning with random projections is worth studying; (2) the comparison of LSTD(λ)-RP to l_1 and l_2 regularized approaches asks for further investigation. (3) the role of $m(\Pi_{\mathcal{F}}V)$ in the error bounds is in need of discussion.

Acknowledgments

This work is partially supported by the National Key Research and Development Program of China (No. 2017YFC0803704), the National Natural Science Foundation of China (Grant No. 61772525, Grant No. 61772524, Grant No. 61702517 and Grant No. 61402480) and the Beijing Natural Science Foundation (Grant No. 4182067 and Grant No. 4172063).

References

[Att *et al.*, 2000] Michael Kearns Att, Michael Kearns, and Satinder Singh. Bias-variance error bounds for temporal difference updates. In *In Proceedings of the 13th Annual Conference on Computational Learning Theory*. Citeseer, 2000.

[Balakrishna *et al.*, 2010] Poornima Balakrishna, Rajesh Ganesan, and Lance Sherry. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures. *Transportation Research Part C: Emerging Technologies*, 18(6):950–962, 2010.

[Boyan, 1999] Justin A Boyan. Least-squares temporal difference learning. In *ICML*, pages 49–56. Morgan Kaufmann Publishers Inc., 1999.

[Boyan, 2002] Justin A Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.

[Bradtke and Barto, 1996] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3):33–57, 1996.

[Dann *et al.*, 2014] Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: a survey and comparison. *JMLR*, 15(1):809–883, 2014.

[Dann, 2012] Christoph Dann. Algorithms for fast gradient temporal difference learning. In *Autonomous Learning Systems Seminar, TU Darmstadt*, 2012.

[Farahmand and Szepesvari, 2011] Amir-massoud Farahmand and Csaba Szepesvari. *Regularization in reinforcement learning*. University of Alberta, 2011.

[Frank *et al.*, 2008] Jordan Frank, Shie Mannor, and Doina Precup. Reinforcement learning in the presence of rare events. In *ICML*, pages 336–343. ACM, 2008.

[Gehring *et al.*, 2016] Clement Gehring, Yangchen Pan, and Martha White. Incremental truncated lstd. In *IJCAI*, pages 1505–1511. AAAI Press, 2016.

[Geist and Pietquin, 2013] Matthieu Geist and Olivier Pietquin. Algorithmic survey of parametric value function approximation. *TNNLS*, 24(6):845–867, 2013.

[Geist and Scherrer, 2014] Matthieu Geist and Bruno Scherrer. Off-policy learning with eligibility traces: A survey. *JMLR*, 15(1):289–333, 2014.

[Ghavamzadeh *et al.*, 2010] Mohammad Ghavamzadeh, Alessandro Lazaric, Odalric Maillard, and Rémi Munos. Lstd with random projections. In *NIPS*, pages 721–729, 2010.

[Kolter and Ng, 2009] J Zico Kolter and Andrew Y Ng. Regularization and feature selection in least-squares temporal difference learning. In *ICML*, pages 521–528. ACM, 2009.

[Li *et al.*, 2018] Haifang Li, Yingce Xia, and Wensheng Zhang. Finite sample analysis of lstd with random projections and eligibility traces. *arXiv preprint arXiv:1805.10005*, 2018.

[Liang *et al.*, 2016] Yitao Liang, Marlos C Machado, Erik Talvitie, and Michael Bowling. State of the art control of atari games using shallow reinforcement learning. In *AAMAS*, pages 485–493. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[Maei, 2011] Hamid Reza Maei. Gradient temporal-difference learning algorithms. 2011.

[Maillard and Munos, 2009] Odalric Maillard and Rémi Munos. Compressed least-squares regression. In *NIPS*, pages 1213–1221, 2009.

[Mohri and Rostamizadeh, 2010] Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary φ-mixing and β-mixing processes. *JMLR*, 11(Feb):789–814, 2010.

[Pan *et al.*, 2017a] Yangchen Pan, Erfan Sadeqi Azer, and Martha White. Effective sketching methods for value function approximation. *arXiv preprint arXiv:1708.01298*, 2017.

[Pan *et al.*, 2017b] Yangchen Pan, Adam M White, and Martha White. Accelerated gradient temporal difference learning. In *AAAI*, pages 2464–2470, 2017.

[Puterman, 2014] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Riedmiller and Gabel, 2007] Martin Riedmiller and Thomas Gabel. On experiences in a complex and competitive gaming domain: Reinforcement learning meets robocup. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 17–23. IEEE, 2007.

[Silver *et al.*, 2007] D Silver, R Sutton, and M Müller. Reinforcement learning of local shape in the game of go. In *IJCAI*, pages 1053–1058, 2007.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[Sutton *et al.*, 2009] 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 *NIPS*, pages 1609–1616, 2009.

[Sutton *et al.*, 2014] Rich Sutton, Ashique Rupam Mahmood, Doina Precup, and Hado Hasselt. A new q (lambda) with interim forward view and monte carlo equivalence. In *ICML*, pages 568–576, 2014.

[Sutton, 1996] Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *NIPS*, pages 1038–1044, 1996.

[Tagorti and Scherrer, 2015] Manel Tagorti and Bruno Scherrer. On the rate of convergence and error bounds for lstd (λ). In *ICML*, pages 1521–1529, 2015.

[Tedrake *et al.*, 2004] Russ Tedrake, Teresa Weirui Zhang, and H Sebastian Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *In Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 2849–2854. IEEE, 2004.

[Tsitsiklis *et al.*, 1997] 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.

[Yu, 2015] Huizhen Yu. On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory*, pages 1724–1751, 2015.