Uncorrected Least-Squares Temporal Difference with Lambda-Return

Takayuki Osogami
IBM Research - Tokyo
osogami@jp.ibm.com

Abstract

Temporal difference, TD(λ), learning is a foundation of reinforcement learning and also of interest in its own right for the tasks of prediction. Recently, true online TD(λ) has been shown to closely approximate the "forward view" at every step, while conventional TD(λ) does this only at the end of an episode. We re-examine least-squares temporal difference, LSTD(λ), which has been derived from conventional TD(λ). We design Uncorrected LSTD(λ) in such a way that, when $\lambda = 1$, Uncorrected LSTD(1) is equivalent to the least-squares method for the linear regression of Monte Carlo (MC) return at every step, while conventional LSTD(1) has this equivalence only at the end of an episode, since the MC return is corrected to be unbiased. We prove that Uncorrected LSTD(λ) can have smaller variance than conventional LSTD(λ), and this allows Uncorrected LSTD(λ) to sometimes outperform conventional LSTD(λ) in practice. When $\lambda = 0$, however, Uncorrected LSTD(0) is not equivalent to LSTD. We thus also propose Mixed LSTD(λ), which matches conventional LSTD(λ) at $\lambda = 0$ and Uncorrected LSTD(λ) at $\lambda = 1$. In numerical experiments, we study how the three LSTD(λ)s behave under limited training data.

1 Introduction

A fundamental problem in reinforcement learning is in learning the value function, which maps a state to the expected cumulative reward (expected return) that can be obtained from that state with a policy under consideration (Sutton and Barto 2018). Effectiveness of reinforcement learning algorithms relies on the quality of the estimated value function. Learning the value function is also of interest in its own right for the purpose of prediction. As such, there has been a significant amount of work on learning the value function, where the existing methods may be classified in two ways.

The first classification is with respect to Monte Carlo (MC) or Temporal Difference (TD). An MC method estimates the value function directly on the basis of sampled sequences of immediate rewards (*i.e.*, MC returns). On the other hand, a TD method utilizes the relation that the return from a state is equal to the immediate reward from that state plus the return from the next state. There is also a family of methods, which is known as TD(λ), that mix MC and TD. Here, λ is a parameter between 0 and 1, where TD(0) reduces to TD, and TD(1) reduces to MC. A study shows that TD(1) tends to suffer from the high variance of MC returns, while TD(0) tends to suffer from the bias in the estimator of return; the optimal choice is often $0 < \lambda < 1$ (Sutton 1988).

The second classification is with respect to how the value function is updated. Two popular approaches are stochastic approximation and least-squares methods. In both approaches, the value function can be updated every time a new sample of data is obtained. With stochastic approximation, the value function is updated by the amount that is controlled by a step size (learning rate). On the other hand, a least-squares method recursively computes a matrix and a vector, which are used to compute the weights of the value function in a way that mean squared error is minimized. In the literature, a TD(λ) method with stochastic approximation is simply referred to as TD(λ) (Sutton 1988), and a least-squares TD(λ) method is referred to as LSTD(λ) (Boyan 2002). LSTD(λ) is sometimes called recursive LSTD(λ) to emphasize the recursive procedure, but we simply call LSTD(λ) in this paper.

Stochastic approximation has the advantage of small computational complexity per step (specifically, linear in the number of weights, while least-squares methods have quadratic complexity). On the other hand, least-squares methods tend to be sample efficient (Boyan 2002; Bradtke and Barto 1996; Xu, He, and Hu 2002), because they fully utilize available data (sufficient information is kept in the recursively computed matrix and vector)[1]. Also, the performance of TD(λ) is sensitive to the step size and the initial weights, while LSTD(λ) does not suffer from this sensitivity. In general, LSTD(λ) approximates the value function with a linear function, which we also assume throughout the paper.

Recently, van Seijen and Sutton; van Seijen et al. (2014; 2016) have proposed true online TD(λ) and showed that it has sound theoretical basis and empirically performs better than conventional TD(λ). This motivates us to re-examine conventional LSTD(λ) by Boyan (2002), which has been derived from conventional TD(λ) and shown to converge to the

[1] The inefficiency of stochastic approximation may be alleviated by the use of replay memory (Lin 1993)

weights which conventional TD(λ) converges to.

We derive Uncorrected LSTD(λ) in such a way that, when $\lambda = 1$, Uncorrected LSTD(1) is equivalent to "the Least-Squares method for the linear regression of Monte Carlo return" (LSMC) at every step. This differs from Boyan's LSTD(1), which corrects the MC return by "bootstrap" to make the resulting estimator unbiased. We prove that Uncorrected LSTD(λ), while it is only asymptotically unbiased, can have smaller variance than Boyan's. Our numerical experiments suggest that Uncorrected LSTD(λ) can indeed outperform Boyan's for some cases.

Uncorrected LSTD(λ) is derived from linear regression via the method of instrumental variables, similar to LSTD by Bradtke and Barto (1996). To shed light on the difference between Uncorrected and Boyan's LSTD(λ), we re-derive Boyan's LSTD(λ) from modified linear regression via instrumental variables. Our main contribution is in our analysis that illuminates how the variants of LSTD(λ) can be derived from the variants of linear regression as well as how Uncorrected LSTD(λ) can have small variance.

In the analysis, we also show that, when $\lambda = 0$, Boyan's LSTD(0) is equivalent to LSTD (Bradtke and Barto 1996), while Uncorrected LSTD(0) is not. This motivates us to propose Mixed LSTD(λ), which mixes the two LSTD(λ)s in a way that it matches LSTD at $\lambda = 0$ and LSMC at $\lambda = 1$. We prove that all of the three LSTD(λ)s converge to the same solution as the amount of training data tends to infinity, but they behave differently with limited training data. Our numerical experiments suggest that Boyan's LSTD(λ) is relatively more sensitive to the particular values of its hyperparameters than the other two LSTD(λ)s, and one may find well performing LSTD(λ) for a given domain from the family of Mixed LSTD(λ). Our secondary contribution thus includes Mixed LSTD(λ) and the empirical characterization of the three LSTD(λ)s under limited training data.

1.1 Related work

In the prior work on TD(λ) and LSTD(λ), it is standard to correct the MC return by bootstrap, because the correction reduces the bias, as has been discussed in Peng and Williams (1996) and Watkins (1989) (Chapter 7). The side effect of the correction on the variance has not been a major focus of the prior work, and uncorrected MC return has not been used to derive existing LSTD(λ)s. In the following, we discuss the work related to LSTD(λ).

LSTD(λ) has been extended or modified in several ways. These include incremental truncated LSTD (Gehring, Pan, and White 2016), iLSTD(λ) (Geramifard et al. 2007), forgetful LSTD(λ) (Vanseijen and Sutton 2015), generalized LSTD(λ) (Ueno et al. 2011), and off-policy LSTD(λ) (Mahmood, van Hasselt, and Sutton 2014). However, all follow Boyan's LSTD(λ), and one may consider the corresponding extensions or modifications to our LSTD(λ)s. LSTD(λ) is also studied by Xu, He, and Hu (2002), but their LSTD(λ) updates the weights in an essentially equivalent manner to Boyan's with a different computational procedure.

LSTD(λ) can be extended to deal with action-value functions, and the resulting method is referred to as LSTD-Q(λ). LSTD-Q(λ) can be used for policy evaluation in reinforcement learning, which has been studied as least-squares policy iteration, LSPI(λ) (Lagoudakis and Parr 2003; Szepesvári 2010). Our LSTD(λ)s can also be extended to LSTD-Q(λ) and LSPI(λ).

Least-squares policy evaluation (λ-LSPE) (Nedić and Bertsekas 2003) is conceptually related to LSTD(λ). However, unlike LSTD(λ), λ-LSPE first finds a least-squares solution of a subproblem and updates weights by the amount specified by a step size, similar to TD(λ).

2 Settings

The purpose of LSTD(λ) is to learn the value function $V(\cdot)$ for a Markov reward process, which is specified with a tuple $(\mathcal{S}, \mathbf{P}, R, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathbf{P}$ is a transition probability matrix, R is a reward function, and γ is a discount factor with $0 < \gamma < 1$. For $s, s' \in \mathcal{S}$, $\mathbf{P}_{s,s'}$ denotes the probability that the next state is s' when the current state is s, and $R(s)$ denotes the expected reward obtained at s. Throughout, we assume $\mathcal{S}$ to be a finite set.

The value function maps a state $s \in \mathcal{S}$ to the expected cumulative reward to be obtained from s:

$$V(s) = \sum_{m=0}^{\infty} \gamma^m \sum_{s' \in \mathcal{S}} (\mathbf{P}^m)_{s,s'} R(s'), \tag{1}$$

where $\mathbf{P}^m$ is the m-step transition probability matrix. Thus, $(\mathbf{P}^m)_{s,s'}$ is the probability that the state after m transitions is s' give that the current state is s. The reward after m steps is discounted by γ^m.

A fundamental property of the value function is given by the Bellman equation, which here we represent in an extended form and refer to it as an n-step Bellman equation:

$$V(s) = \sum_{m=0}^{n-1} \gamma^m \sum_{s' \in \mathcal{S}} (\mathbf{P}^m)_{s,s'} R(s') + \gamma^n \sum_{s' \in \mathcal{S}} (\mathbf{P}^n)_{s,s'} V(s') \tag{2}$$

for $n \geq 1$. The case with $n = 1$ reduces to the standard Bellman equation. On the right-hand side of (2), the first term represents the expected cumulative reward for the next n steps, and the second term represents the expected cumulative reward afterwards.

LSTD(λ) learns $V(\cdot)$ from training data, which we assume to be a series of states and rewards, $\{(s_t, r_{t+1})\}_t$. By convention, r_{t+1} denotes the reward obtained upon transitioning from s_t. More specifically, LSTD(λ) seeks to learn a linear function, $V_{\boldsymbol{\theta}}(\boldsymbol{\phi}(s)) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s)$, that best approximates $V(s)$, where $\boldsymbol{\phi}(s)$ is the feature vector of s, and $\boldsymbol{\theta}$ is the vector of weights (coefficients) of the linear function. A Markov reward process may be considered as a Markov decision process with a fixed policy. LSTD(λ) can thus be used for policy evaluation in the framework of policy iteration.

3 LSTD(λ)

LSTD(λ) can be derived in a way that that the n-step Bellman equations (2) are approximately satisfied with $V_\theta(\cdot)$ *for a range of* n. Depending on how the n-step Bellman equations are weighted, we arrive at variations of

(a) Boyan's LSTD(λ)

$\mathbf{A}^{-1} \leftarrow \frac{1}{\alpha}\mathbf{I}$; $\mathbf{b} \leftarrow \mathbf{0}$;
$\mathbf{z}_0 \leftarrow \boldsymbol{\phi}_0$;
for $t = 0, 1, \ldots$ **do**
  $\boldsymbol{\delta} \leftarrow \boldsymbol{\phi}_t - \gamma\,\boldsymbol{\phi}_{t+1}$;
  $\mathbf{A}^{-1} \leftarrow (\mathbf{A} + \mathbf{z}_t\boldsymbol{\delta}^\top)^{-1}$;
  $\mathbf{b} \leftarrow \mathbf{b} + r_{t+1}\,\mathbf{z}_t$;
  $\mathbf{z}_{t+1} \leftarrow \lambda\,\gamma\,\mathbf{z}_t + \boldsymbol{\phi}_{t+1}$;
  $\boldsymbol{\theta} = \mathbf{A}^{-1}\,\mathbf{b}$;
end

(b) Uncorrected LSTD(λ)

$\mathbf{A}^{-1} \leftarrow \frac{1}{\alpha}\mathbf{I}$; $\mathbf{b} \leftarrow \mathbf{0}$;
$\mathbf{z}_{-1} \leftarrow \mathbf{0}$; $\mathbf{z}_0 \leftarrow \boldsymbol{\phi}_0$;
for $t = 0, 1, \ldots$ **do**
  $\boldsymbol{\delta} \leftarrow \mathbf{z}_t - \gamma\,\mathbf{z}_{t-1}$;
  $\mathbf{A}^{-1} \leftarrow (\mathbf{A} + \boldsymbol{\delta}\,\boldsymbol{\phi}_t^\top)^{-1}$;
  $\mathbf{b} \leftarrow \mathbf{b} + r_{t+1}\,\mathbf{z}_t$;
  $\mathbf{z}_{t+1} \leftarrow \lambda\,\gamma\,\mathbf{z}_t + \boldsymbol{\phi}_{t+1}$;
  $\boldsymbol{\theta} = \mathbf{A}^{-1}\,\mathbf{b}$;
end

(c) Mixed LSTD(λ)

$\mathbf{A}^{-1} \leftarrow \frac{1}{\alpha}\mathbf{I}$; $\mathbf{b} \leftarrow \mathbf{0}$;
$\mathbf{z}_{-1} \leftarrow \mathbf{0}$; $\mathbf{z}_0 \leftarrow \boldsymbol{\phi}_0$;
for $t = 0, 1, \ldots$ **do**
  $\boldsymbol{\delta} \leftarrow \lambda\,(\mathbf{z}_t - \gamma\,\mathbf{z}_{t-1})$;
  $\mathbf{A}^{-1} \leftarrow (\mathbf{A} + \boldsymbol{\delta}\,\boldsymbol{\phi}_t^\top)^{-1}$;
  $\boldsymbol{\delta} \leftarrow (1-\lambda)(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})$;
  $\mathbf{A}^{-1} \leftarrow (\mathbf{A} + \mathbf{z}_t\,\boldsymbol{\delta}^\top)^{-1}$;
  $\mathbf{b} \leftarrow \mathbf{b} + r_{t+1}\,\mathbf{z}_t$;
  $\mathbf{z}_{t+1} \leftarrow \lambda\,\gamma\,\mathbf{z}_t + \boldsymbol{\phi}_{t+1}$;
  $\boldsymbol{\theta} = \mathbf{A}^{-1}\,\mathbf{b}$;
end

Algorithm 1: LSTD(λ) algorithms, where the step of the form $\mathbf{A}^{-1} \leftarrow (\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1}$ first computes $(\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1}$ from $\mathbf{A}^{-1}$ by the use of the Sherman-Morrison lemma and then replaces the old $\mathbf{A}^{-1}$ with $(\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1}$.

LSTD(λ). Although our derivation is rather involved, Uncorrected LSTD(λ), which we propose in this paper, consists of simple calculations and can be directly compared against Boyan's. So, we start by discussing Uncorrected LSTD(λ) shown in Algorithm 1(b) with comparison to Boyan's (Algorithm 1(a)). Algorithm 1(c) mixes the two LSTD(λ)s and will be discussed in Section 3.5.

At each time step t, both of the two LSTD(λ)s update the weights, $\boldsymbol{\theta}$, to the solution of a system of linear equations, $\boldsymbol{\theta} = \mathbf{A}^{-1}\,\mathbf{b}$, where $\mathbf{A}^{-1}$ and $\mathbf{b}$ are recursively computed. Note that $(\mathbf{A} + \mathbf{u}\,\mathbf{v}^\top)^{-1}$ can be computed from $\mathbf{A}^{-1}$ via the Sherman-Morrison lemma (rank-one update). The two LSTD(λ)s differ only by two lines. While Uncorrected LSTD(λ) updates $\mathbf{A}$ by $(\mathbf{z}_t - \gamma\,\mathbf{z}_{t-1})\,\boldsymbol{\phi}_t^\top$, Boyan's updates $\mathbf{A}$ by $\mathbf{z}_t\,(\boldsymbol{\phi}_t - \gamma\,\boldsymbol{\phi}_{t+1})^\top$, where $\boldsymbol{\phi}_t \equiv \boldsymbol{\phi}(s_t)$, and $\mathbf{z}_t$ is called an eligibility trace. Both of the two LSTD(λ)s run in $O(k^2)$ time and space, where k is the dimension of $\boldsymbol{\phi}_t$.

Here, $\alpha > 0$ is set sufficiently large to ensure that $\mathbf{A}^{-1}$ exists and can be updated in a numerically stable manner. Alternatively, one may also update $\mathbf{A}$, while $\mathbf{A}^{-1}$ does not exists, and compute $\mathbf{A}^{-1}$ from $\mathbf{A}$ when it becomes invertible. Once $\mathbf{A}^{-1}$ is obtained, $\mathbf{A}^{-1}$ can be updated as in Algorithm 1. These are standard techniques in recursive least-squares methods.

In the rest of this section, we derive the three LSTD(λ)s in Algorithm 1 and discuss where the difference stems from. The performance of these LSTD(λ)s will be evaluated empirically in Section 4.

3.1 Deriving Uncorrected LSTD(λ)

First, we derive Uncorrected LSTD(λ) by following the approach taken in Bradtke and Barto (1996), who have derived LSTD. Throughout, we assume $0 \leq \lambda \leq 1$.

At (time) step T, one can consider n-step Bellman equations for the state s_t visited at each step $t < T$. We take the weighted sum of those n-step Bellman equations. Specifically, for $1 \leq n < T - t$, the n-step Bellman equation (Eq. (2)) is weighted by $(1-\lambda)\,\lambda^{n-1}$. We also add the MC return (Eq. (1)) with weight λ^{T-t-1}. The resulting weighted sum of the $T - t$ equations is given by

$$
\begin{aligned}
V(s_t) = &\sum_{m=0}^{T-t-1} (\lambda\,\gamma)^m \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\,R(s) \\
&+ (1-\lambda)\,\gamma \sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1} \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\,V(s) \\
&+ (\lambda\,\gamma)^{T-t-1} \sum_{m=T-t}^{\infty} \gamma^{m+1-(T-t)} \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\,R(s).
\end{aligned} \tag{3}
$$

Because our regressor $V_{\boldsymbol{\theta}}(\cdot)$ is linear, one may find its weights, $\boldsymbol{\theta}$, via linear regression, *if* $\mathbf{P}$ *is known*:

$$
\left\{ \boldsymbol{\phi}_t - (1-\lambda)\,\gamma \sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1} \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\,\boldsymbol{\phi}(s) \longmapsto \sum_{m=0}^{T-t-1} (\lambda\,\gamma)^m\,r_{t+1+m} \right\}_{t=0}^{T-1}, \tag{4}
$$

where we use $\{\mathbf{x}_t \mapsto y_t\}_{t=0}^{T-1}$ to denote the linear regression where input variables are $\mathbf{x}_t$ and the corresponding target variable is y_t for $t = 0, \ldots, T-1$. The target variable in (4) involves the reward, r_{t+1+m}, observed at time $t+1+m$. The reward after step T (*i.e.*, r_t for $t > T$) has not been observed at step T and is not included in (4). The target variable thus has the observation noise of

$$
\begin{aligned}
&\sum_{m=0}^{T-t-1} (\lambda\gamma)^m \left(\sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s} R(s) - r_{t+1+m} \right) \\
&+ (\lambda\gamma)^{T-t-1} \sum_{m=T-t}^{\infty} \gamma^{m+1-(T-t)} \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s} R(s),
\end{aligned} \tag{5}
$$

which we will revisit when we discuss convergence of Uncorrected LSTD(λ).

Because $\mathbf{P}$ is unknown, the input variables for the linear regression (4) cannot be directly observed. Instead, one can observe a sample path, $\{(s_t, r_{t+1})\}_{t=0}^{T-1}$. Then one may use $\boldsymbol{\phi}_t - (1-\lambda)\,\gamma \sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1}\,\boldsymbol{\phi}_{t+m}$ as input variables, which however involve the observation noise of

$$-(1-\lambda)\,\gamma \sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1} \left(\sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\, \boldsymbol{\phi}(s) - \boldsymbol{\phi}_{t+m} \right). \tag{6}$$

With this observation noise in input variables, the standard least-squares solution would be biased, which however can be corrected with the method of instrumental variables (Young 2011). Following Bradtke and Barto (1996), we use $\boldsymbol{\phi}_t$ as instrumental variables. Then the least-squares solution of the regression (4) is given by the solution of $\frac{1}{T}\mathbf{A}_T^{\mathrm{Unc}}\,\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$, where

$$\mathbf{A}_T^{\mathrm{Unc}} \equiv \sum_{t=0}^{T-1} \boldsymbol{\phi}_t \left(\boldsymbol{\phi}_t - (1-\lambda)\,\gamma \sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1}\, \boldsymbol{\phi}_{t+m} \right)^\top \tag{7}$$

$$\mathbf{b}_T \equiv \sum_{t=0}^{T-1} \boldsymbol{\phi}_t \sum_{m=0}^{T-t-1} (\lambda\,\gamma)^m\, r_{t+1+m}. \tag{8}$$

We compute $\mathbf{A}_T^{\mathrm{Unc}}$ and $\mathbf{b}_T$ recursively, as follows:

Theorem 1. *We can compute* (7)-(8) *recursively as follows:* $\mathbf{A}_{T+1}^{\mathrm{Unc}} = \mathbf{A}_T^{\mathrm{Unc}} + (\mathbf{z}_T - \gamma\,\mathbf{z}_{T-1})\,\boldsymbol{\phi}_T^\top$ *and* $\mathbf{b}_{T+1} = \mathbf{b}_T + \mathbf{z}_T\, r_{T+1}$ *for* $T > 0$*, starting from* $\mathbf{A}_0^{\mathrm{Unc}} = \mathbf{O}$ *and* $\mathbf{b}_0 = \mathbf{0}$*, where the eligibility trace* $\mathbf{z}_T \equiv \sum_{t=0}^{T} (\lambda\,\gamma)^{T-t}\,\boldsymbol{\phi}_t$ *can be computed recursively as* $\mathbf{z}_T = \lambda\,\gamma\,\mathbf{z}_{T-1} + \boldsymbol{\phi}_T$ *for* $T > 0$*.*

Proof. The theorem can be proved in a straightforward manner from the definition of the eligibility trace. We provide a complete proof in the supplementary material (Osogami 2019). □

One may also recursively compute $(\mathbf{A}_T^{\mathrm{Unc}})^{-1}$ via the Sherman-Morrison lemma, and this gives Algorithm 1(b).

Now, consider the special cases of Uncorrected LSTD(λ). When $\lambda = 1$, at each step T, Uncorrected LSTD(1) is equivalent to LSMC, which finds the least-squares solution of the linear regression of Monte Carlo return up to T (*i.e.*, $\{\boldsymbol{\phi}_t \mapsto \sum_{m=0}^{T-t-1} \gamma^m\, r_{t+1+m}\}_{t=0}^{T-1}$). Specifically, when $\lambda = 1$, we have that $\mathbf{A}_T^{\mathrm{Unc}}$ and $\mathbf{b}_T$ reduce to

$$\mathbf{A}_T^{\mathrm{Unc}} = \sum_{t=0}^{T-1} \boldsymbol{\phi}_t\,\boldsymbol{\phi}_t^\top \text{ and } \mathbf{b}_T = \sum_{t=0}^{T-1} \boldsymbol{\phi}_t \sum_{m=0}^{T-t-1} \gamma^m r_{t+1+m}. \tag{9}$$

When $\lambda = 1$, input variables have no observation noise, and the instrumental variables, $\boldsymbol{\phi}_t$, are equivalent to the input variables. Thus, Uncorrected LSTD(1) recursively finds standard least-squares solutions.

When $\lambda = 0$, Uncorrected LSTD(0) is slightly different from LSTD (Bradtke and Barto 1996), which recursively computes the solution of $\frac{1}{T}\mathbf{A}_T^{\mathrm{LSTD}}\,\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T^{\mathrm{LSTD}}$, where

$$\mathbf{A}_T^{\mathrm{LSTD}} \equiv \sum_{t=0}^{T-1} \boldsymbol{\phi}_t\,(\boldsymbol{\phi}_t - \gamma\,\boldsymbol{\phi}_{t+1})^\top \text{ and } \mathbf{b}_T^{\mathrm{LSTD}} \equiv \sum_{t=0}^{T-1} \boldsymbol{\phi}_t\, r_{t+1}. \tag{10}$$

When $\lambda = 0$, we have $\mathbf{b}_T = \mathbf{b}_T^{\mathrm{LSTD}}$, but $\mathbf{A}_T^{\mathrm{Unc}}$ differs from $\mathbf{A}_T^{\mathrm{LSTD}}$ by $\mathbf{A}_T^{\mathrm{Unc}} - \mathbf{A}_T^{\mathrm{LSTD}} = \gamma\,\boldsymbol{\phi}_{T-1}\,\boldsymbol{\phi}_T^\top$, which, however, becomes negligible ($\frac{1}{T}|\mathbf{A}_T^{\mathrm{Unc}} - \mathbf{A}_T^{\mathrm{LSTD}}| \to 0$) as $T \to \infty$.

3.2 Convergence of Uncorrected LSTD(λ)

Bradtke and Barto (1996) have shown that the weights given by LSTD converge to the true values. For the convergence, the key property that need to be verified is that instrumental variables are uncorrelated with observation noise in input variables and in target variables. Although we use the same instrumental variables as Bradtke and Barto (1996), our observation noise is different from theirs.

First, consider the observation noise in input variables (input noise). Ignoring a constant factor in (6), our input noise has the form $\zeta \equiv \sum_{m=0}^{T-t-1} (\lambda\,\gamma)^{m-1}\,\zeta_m$, where ζ_m is the input noise at step $t+m$. It has been shown in Bradtke and Barto (1996) that ζ_0 is uncorrelated with $\boldsymbol{\phi}_t$. By analogous reasons (more specifically, by considering an m-step transition as a single-step transition, $\mathbf{P} \leftarrow \mathbf{P}^m$), for each $m > 0$, ζ_m is uncorrelated with $\boldsymbol{\phi}_t$. Therefore, $\boldsymbol{\phi}_t$ is uncorrelated with our input noise.

It can be seen in (5) that our observation noise in target variables has the form $\sum_{m=0}^{T-t-1} (\lambda\,\gamma)^m \eta_m + \eta_{T-t}$, where η_m is the observation noise of the reward at step $t+1+m$ for $0 \le m < T-t$, and η_{T-t} is the remaining observation noise. It has been shown in Bradtke and Barto (1996) that η_0 is uncorrelated with $\boldsymbol{\phi}_t$. By analogous reasons, η_m is uncorrelated with $\boldsymbol{\phi}_t$ for each $0 < m < T-t$.

On the other hand, η_{T-t} can be correlated with $\boldsymbol{\phi}_t$. However, unlike η_m for $m < T-t$, η_{T-t} is deterministic given s_t[2]. Thus, if we consider a hypothetical situation where η_{T-t} is observed and included in the target variable, $\boldsymbol{\phi}_t$ is uncorrelated with the observation noise in target variables.

The following lemma suggests that Uncorrected LSTD(λ) gives the weights that are equivalent to those found in this hypothetical situation:

Lemma 1. *Let* $\boldsymbol{\theta}_T$ *be the solution of* $\frac{1}{T}\mathbf{A}_T^{\mathrm{Unc}}\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$ *and* $\boldsymbol{\theta}_T^\star$ *the solution of* $\frac{1}{T}\mathbf{A}_T^{\mathrm{Unc}}\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T^\star$*, where* $\mathbf{A}_T^{\mathrm{Unc}}$ *and* $\mathbf{b}_T$ *are given by* (7)-(8)*, and*

$$\mathbf{b}_T^\star \equiv \sum_{t=0}^{T-1} \boldsymbol{\phi}_t \Bigg(\sum_{m=1}^{T-t-1} (\lambda\,\gamma)^{m-1}\,\boldsymbol{\phi}_{t+m} + \lambda^{T-t-1} \sum_{m=T-t}^{\infty} \gamma^{m+1-(T-t)} \sum_{s\in\mathcal{S}} (\mathbf{P}^m)_{s_t,s}\, R(s) \Bigg). \tag{11}$$

If $\frac{1}{T}\mathbf{A}_T^{\mathrm{Unc}}$ *converges to an invertible matrix as* $T \to \infty$*, then* $|\boldsymbol{\theta}_T - \boldsymbol{\theta}_T^\star| \to \mathbf{0}$ *as* $T \to \infty$*.*

Proof. Here, we only provide a proof sketch, but a complete proof can be found in the supplementary material (Osogami 2019).

[2]Both of η_{T-t} and ϕ_t depend on s_t, so they are random and correlated before time t. After t, the randomness associated with s_t is resolved, and the value of η_{T-t} is determined (has no randomness), because s_t was the only randomness in η_{T-t}.

By the continuity of matrix inverse, we can show

$$\lim_{T\to\infty}(\boldsymbol{\theta}_T - \boldsymbol{\theta}^\star_T) = \left(\lim_{T\to\infty}\frac{1}{T}\mathbf{A}^{\mathrm{Unc}}_T\right)^{-1}\lim_{T\to\infty}\frac{1}{T}(\mathbf{b}_T - \mathbf{b}^\star_T). \quad (12)$$

It thus suffices to show $\frac{1}{T}|\mathbf{b}^\star_T - \mathbf{b}_T| \to 0$ as $T\to\infty$. Now, because the state space $\mathcal{S}$ is finite, there exists $c<\infty$ such that $R(s)\le c$ and $|\boldsymbol{\phi}(s)|\le c$ elementwise for any $s\in\mathcal{S}$. Then we can show the following elementwise inequality:

$$\frac{1}{T}|\mathbf{b}^\star_T - \mathbf{b}_T| \le c^2\,\frac{\gamma}{1-\gamma}\,\frac{1}{T}\,\frac{1-(\lambda\gamma)^T}{1-\lambda\gamma}, \quad (13)$$

which tends to 0 as $T\to\infty$. □

Note that $\mathbf{b}^\star_T$ involves η_{T-t} that is observed in the hypothetical situation. We discuss the invertibility of $\frac{1}{T}\mathbf{A}^{\mathrm{Unc}}_T$ in the following.

The following theorem specifies what weights Uncorrected LSTD(λ) converges to as $T\to\infty$.

Theorem 2. *Let $\boldsymbol{\theta}_T$ be the weights given by Uncorrected LSTD(λ) at step T. Suppose that the Markov chain of state transition is ergodic*[3]*. Let $\boldsymbol{\pi}$ be the vector of steady state probability at each $s\in\mathcal{S}$. Let $\mathbf{r}$ be the vector of expected immediate reward from each $s\in\mathcal{S}$. Let $\boldsymbol{\Phi}$ be the matrix whose rows are $\boldsymbol{\phi}(s)$ for $s\in\mathcal{S}$. Then, as $T\to\infty$, $\boldsymbol{\theta}_t$ converges to the solution of $\bar{\mathbf{A}}\,\boldsymbol{\theta} = \bar{\mathbf{b}}$ almost surely, where $\bar{\mathbf{A}} \equiv \boldsymbol{\Phi}^\top \mathrm{Diag}(\boldsymbol{\pi})\,(\mathbf{I}-\gamma\,\mathbf{P})\,(\mathbf{I}-\lambda\,\gamma\,\mathbf{P})^{-1}\,\boldsymbol{\Phi}$ and $\bar{\mathbf{b}} \equiv \boldsymbol{\Phi}^\top \mathrm{Diag}(\boldsymbol{\pi})\,(\mathbf{I}-\lambda\,\gamma\,\mathbf{P})^{-1}\,\mathbf{r}$. Here, $\mathrm{Diag}(\boldsymbol{\pi})$ is the diagonal matrix whose diagonal elements are $\boldsymbol{\pi}$. We assume $0\le\lambda\le 1$ and $0\le\gamma<1$.*

Proof. Here, we only provide a proof sketch, but a complete proof can be found in the supplementary material (Osogami 2019).

At each step T, Uncorrected LSTD(λ) gives the weight vector $\boldsymbol{\theta}_T$ that is the solution of $\frac{1}{T}\mathbf{A}^{\mathrm{Unc}}_T\,\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$. Therefore, it suffices to show $\frac{1}{T}\mathbf{A}^{\mathrm{Unc}}_T \to \bar{\mathbf{A}}$ and $\frac{1}{T}\mathbf{b}_T \to \bar{\mathbf{b}}$ as $T\to 0$. We can prove these almost sure convergence by relating the time average to the ensemble average (almost surely) via the pointwise ergodic theorem, where the key assumption that we exploit is the ergodicity of the Markov chain of state transition. □

In the theorem, $(\mathbf{I}-\gamma\,\mathbf{P})$ and $(\mathbf{I}-\lambda\,\gamma\,\mathbf{P})$ have full rank, because $\mathbf{P}$ is a stochastic matrix[4]. Also, $\boldsymbol{\pi}>0$ elementwise by ergodicity. Hence, $\bar{\mathbf{A}}$ is invertible, as long as the feature vectors are chosen in a way that $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ has full rank.

3.3 Where does Boyan's LSTD(λ) come from?

We have formally derived Uncorrected LSTD(λ) via instrumental variables. Then Uncorrected LSTD(λ) is shown to reduce to LSMC at $\lambda=1$ and is asymptotically equivalent to LSTD at $\lambda=0$. We have also established asymptotic convergence analogous to what has been shown for LSTD (Bradtke and Barto 1996). These results suggest that Uncorrected LSTD(λ) is a natural extension of LSTD.

However, Boyan's LSTD(λ) is also a natural extension of LSTD. In this section, we derive Boyan's LSTD(λ) via the method of instrumental variables, which is quite different from Boyan's derivation from TD(λ) (Boyan 2002). Our analysis will illuminate how the two LSTD(λ)s differ and shed new lights on LSTD(λ)s.

Now, at (time) step T, we take the weighted sum of the n-step Bellman equations with $1\le n\le T-t$ for the state s_t visited at step $t<T$. Here, n-step Bellman equation is weighted by $(1-\lambda)\lambda^{n-1}$ for $1\le n<T-t$ and by λ^{T-t-1} for $n=T-t$. The resulting weighted sum is given by

$$\begin{aligned} V(s_t) = &\sum_{m=0}^{T-t-1}(\lambda\,\gamma)^m\sum_{s\in\mathcal{S}}(\mathbf{P}^m)_{s_t,s}\,R(s)\\ &+(1-\lambda)\,\gamma\sum_{m=1}^{T-t-1}(\lambda\,\gamma)^{m-1}\sum_{s\in\mathcal{S}}(\mathbf{P}^m)_{s_t,s}\,V(s)\\ &+\gamma\,(\lambda\,\gamma)^{T-t-1}\sum_{s\in\mathcal{S}}(\mathbf{P}^{T-t})_{s_t,s}\,V(s). \end{aligned} \quad (14)$$

Recall that, when we have derived Uncorrected LSTD(λ), we have added the MC return (Eq. (1)) with weight λ^{T-t-1}, instead of the $(T-t)$-step Bellman equation, which is added in (14) with weight λ^{T-t-1}. This difference is reflected in the last term of (14).

From (14), we arrive at the linear regression $\{\mathbf{x}_t \mapsto y_t\}_{t=0}^{T-1}$, where

$$\begin{aligned} \mathbf{x}_t \equiv\ &\boldsymbol{\phi}_t - (1-\lambda)\gamma\sum_{m=1}^{T-t-1}(\lambda\gamma)^{m-1}\sum_{s\in\mathcal{S}}(\mathbf{P}^m)_{s_t,s}\boldsymbol{\phi}(s)\\ &-\gamma(\lambda\gamma)^{T-t-1}\sum_{s\in\mathcal{S}}(\mathbf{P}^{T-t})_{s_t,s}\boldsymbol{\phi}(s) \end{aligned} \quad (15)$$

$$y_t \equiv \sum_{m=0}^{T-t-1}(\lambda\,\gamma)^m\,r_{t+1+m}. \quad (16)$$

Via the instrumental variables $\boldsymbol{\phi}_t$, the least-squares solution of this linear regression is given by the solution of $\frac{1}{T}\mathbf{A}^{\mathrm{Boy}}_T\,\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$, where $\mathbf{b}_T$ is given by (8), and

$$\begin{aligned} \mathbf{A}^{\mathrm{Boy}}_T \equiv \sum_{t=0}^{T-1}\boldsymbol{\phi}_t\Bigg(&\boldsymbol{\phi}_t - (1-\lambda)\,\gamma\sum_{m=1}^{T-t-1}(\lambda\,\gamma)^{m-1}\,\boldsymbol{\phi}_{t+m}\\ &-\gamma\,(\lambda\,\gamma)^{T-t-1}\,\boldsymbol{\phi}_T\Bigg)^\top. \end{aligned} \quad (17)$$

The following lemma implies that the solution of $\frac{1}{T}\mathbf{A}^{\mathrm{Boy}}_T\,\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$ is what is given by Boyan's LSTD(λ) at step T:

Lemma 2. *Let $\mathbf{z}_k$ be as defined in Theorem 1. Then we can write $\mathbf{A}^{\mathrm{Boy}}_T = \sum_{k=0}^{T-1}\mathbf{z}_k\,(\boldsymbol{\phi}_k - \gamma\,\boldsymbol{\phi}_{k+1})^\top$.*

Proof. The lemma can be proved in a way similar to Theorem 1. See a full proof in the supplementary material (Osogami 2019). □

[3] aperiodic and irreducible, as we assume a finite state space

[4] In Sutton, Mahmood, and White (2016), analogous matrices are shown to be positive definite.

The expression of $\mathbf{A}_T^{\mathrm{Boy}}$ in Lemma 2 is equivalent to Equation (2) in Boyan (2002). One can also show that $\mathbf{A}_T^{\mathrm{Boy}}$ and its inverse can be computed recursively (Boyan 2002; Xu, He, and Hu 2002), and this leads to Algorithm 1(a).

It is straightforward to verify that Uncorrected LSTD(λ) becomes asymptotically equivalent to Boyan's LSTD(λ) as $T \to \infty$. Specifically, the theorem analogous to Theorem 2 holds for Boyan's LSTD(λ).

We now study special cases. Because $\mathbf{z}_T = \boldsymbol{\phi}_T$ when $\lambda = 0$, Boyan's LSTD(0) is equivalent to LSTD, as discussed in Boyan (2002). When $\lambda = 1$, we have from (17) that $\mathbf{A}_T^{\mathrm{Boy}} = \sum_{t=0}^{T-1} \boldsymbol{\phi}_t \left(\boldsymbol{\phi}_t - \gamma^{T-t}\boldsymbol{\phi}_T\right)$, which is different from $\mathbf{A}_T^{\mathrm{Unc}}$ in (9) unless $\boldsymbol{\phi}_T = \mathbf{0}$. Hence, at each step T, Boyan's LSTD(1) is different from LSMC. In Boyan (2002), Boyan's LSTD(1) has been shown to be equivalent to LSMC at the end of an episode, and this is indeed the case, because one should set $\boldsymbol{\phi}_T = \mathbf{0}$ if the episode ends at T.

3.4 Bias-variance tradeoff

In this section, we discuss the quality of the estimators given by Uncorrected and Boyan's LSTD(λ). Specifically, we show that, in a special case, the estimator given by Uncorrected LSTD(λ) is biased but has smaller variance than that given by Boyan's LSTD(λ), which is unbiased.

Proposition 1. *Consider a stateless Markov reward process, where i.i.d. reward with a finite second moment is obtained at each step. Let μ and σ^2 respectively denote the mean and variance of the reward. Let θ_T^{Unc} and θ_T^{Boy}, respectively, be the estimator of the discounted cumulative reward given by Uncorrected and Boyan's LSTD(λ) at step T. Then θ_T^{Boy} is unbiased at each T, while θ_T^{Unc} has the following bias:*

$$\mathbb{E}[\theta_T^{\mathrm{Unc}}] - \frac{\mu}{1-\gamma} = -\frac{\gamma\,\mu}{(1-\gamma)^2 T} + o\left(\frac{1}{T}\right), \tag{18}$$

On the other hand, we have

$$\frac{\mathrm{Var}[\theta_T^{\mathrm{Boy}}]}{\mathrm{Var}[\theta_T^{\mathrm{Unc}}]} = 1 + \frac{2\,\gamma}{(1-\gamma)\,T} + o\left(\frac{1}{T}\right). \tag{19}$$

Proof. The proposition can be proved by careful analysis of the estimators for this special case (see the supplementary material (Osogami 2019)). □

Although (19) hides details in $o\left(\frac{1}{T}\right)$, it actually holds that $\mathrm{Var}[\theta_T^{\mathrm{Boy}}] > \mathrm{Var}[\theta_T^{\mathrm{Boy}}]$ for any T, as is evident in the proof of the proposition.

Proposition 1 clearly shows the bias-variance tradeoff, although it is for a special case with strong assumptions. We may expect that Uncorrected LSTD(λ) can outperform Boyan's for some cases, even though Uncorrected LSTD(λ) generally incurs larger bias than Boyan's. If Uncorrected LSTD(λ) performs better than Boyan's, it is perhaps due to the low variance of Uncorrected LSTD(λ). We hypothesize that analogous bias-variance tradeoff holds in more general settings, and it is an interesting future work to provide a proof for the general case.

3.5 Mixing Uncorrected and Boyan's LSTD(λ)

The bias-variance tradeoff discussed in Section 3.4 motivates us to mix Uncorrected and Boyan's LSTD(λ)s to strike a good tradeoff. To this end, we propose Mixed LSTD(λ), which finds the solution of $\frac{1}{T}\mathbf{A}_T^{\mathrm{Mix}}\boldsymbol{\theta} = \frac{1}{T}\mathbf{b}_T$ in a recursive manner, where $\mathbf{A}_T^{\mathrm{Mix}} \equiv \lambda\,\mathbf{A}_T^{\mathrm{Unc}} + (1-\lambda)\,\mathbf{A}_T^{\mathrm{Boy}}$ for each T. It is straightforward to verify that this leads to Algorithm 1(c).

Note that we use λ to mix Uncorrected and Boyan's LSTD(λ)s without introducing an additional hyperparameter. In this way, Mixed LSTD(0) becomes equivalent to Boyan's LSTD(0), which is equivalent to LSTD. Also, Mixed LSTD(1) becomes equivalent to Uncorrected LSTD(1), which is equivalent to LSMC. Thus, Mixed LSTD(λ) nicely interpolates LSTD and LSMC.

All of the three LSTD(λ)s in Algorithm 1 run in time quadratic in the dimension of $\boldsymbol{\phi}_t$. However, Mixed LSTD(λ) is slower than the others by a constant factor (at most two), because it applies the rank-one update twice.

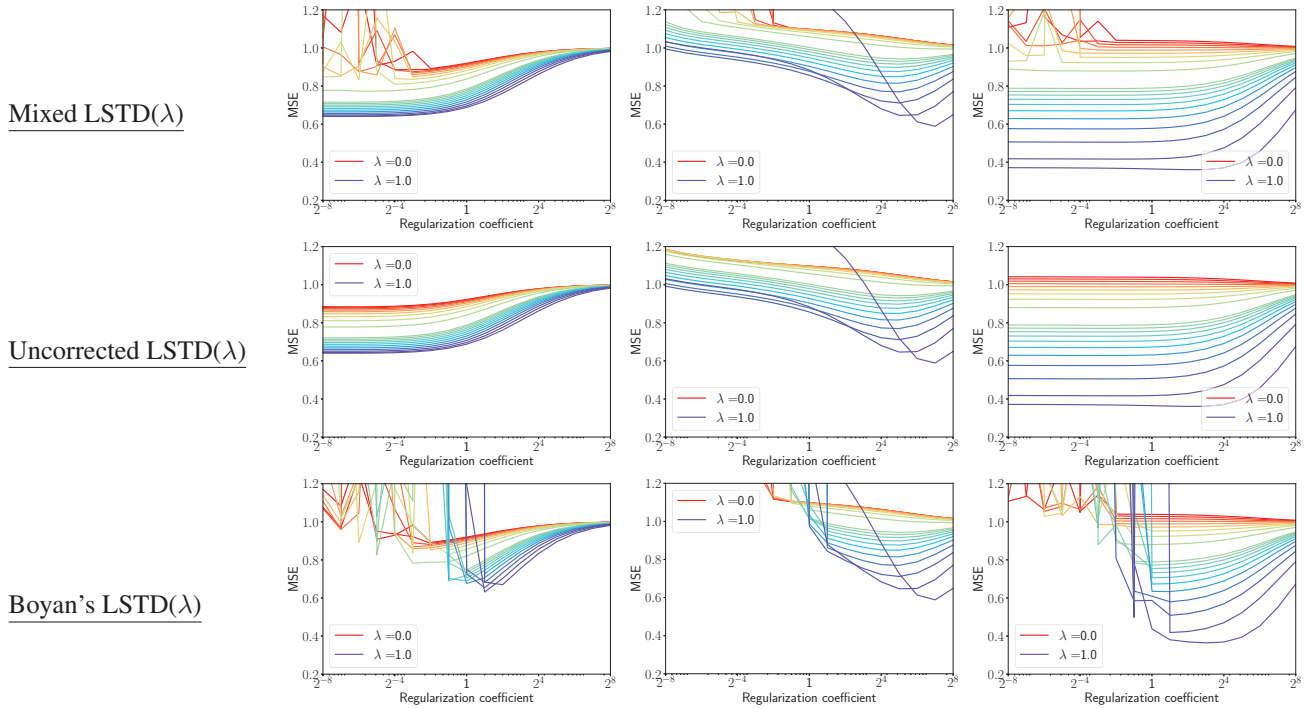
4 Numerical experiments

Boyan's LSTD(λ), Uncorrected LSTD(λ), and Mixed LSTD(λ) become equivalent as the number of time steps T tends to infinity, but the three LSTD(λ)s behave differently for small T. The relative performance of the LSTD(λ)s depends on the domains, and we cannot definitively conclude one LSTD(λ) is better than the others. In this section, we evaluate and compare the performance of the three LSTD(λ)s on randomly constructed Markov reward processes (MRPs), which have been designed in van Seijen et al. (2016) to study the relative performance of various TD(λ) methods during the initial periods of learning.

To generate the random MRPs, we use the code published online[5] by van Seijen et al. (2016). One may thus refer to van Seijen et al. (2016) for the exact settings of the experiments. Here, we briefly summarize the experimental settings. Each random MRP is represented as a tuple (k, b, σ), where k is the number of states, b is the branching factor of the transition from each state, and σ is the standard deviation of reward. Three types of MRPs are considered. The *small* MRP is $(10, 3, 0.1)$, *large* is $(100, 10, 0.1)$, and *deterministic* is $(100, 3, 0)$. For each MRP, three representations (features) of states are studied: *tabular*, *binary*, and *non-binary*. With *tabular*, each state is uniquely represented with a standard basis vector of k dimensions. With *binary*, each state is first represented by a unique integer from 1 to k, which is then encoded into a binary representation of length $\lceil \log_2(k+1) \rceil$. With *non-binary*, each state is randomly mapped to a five dimensional vector according to the standard normal distribution. The performance is evaluated with the mean squared error (MSE) during the first 100 steps for *small* and the first 1,000 steps for *large* and *deterministic*. More precisely, the MSE is the error in the estimated weight, normalized by the MSE when the weights are zero. Throughout, the discount factor is $\gamma = 0.99$.

Figure 1 shows the MSE for each LSTD(λ) on each MRP with each value of λ and regularization coefficient (α in Al-

[5] https://github.com/armahmood/totd-rndmdp-experiments



(a) *small* MRP (b) *large* MRP (c) *deterministic* MRP

Figure 1: Mean squared error (MSE) of Uncorrected, Mixed, and Boyan's LSTD(λ) on the three MRPs with non-binary features as a function of the value of regularization coefficient. Each curve shows the MSE (over 50 runs) with a particular value of λ for $0 \leq \lambda \leq 1$. The legend only shows the color with $\lambda \in \{0, 1\}$, but the intermediate values of λ follow the color map of rainbow.

gorithm 1). Due to space limitations, we show only the results with *non-binary* features in Figure 1, but our discussion and conclusion will also consider other results shown in the supplementary material (Osogami 2019) (Figures 3-5). Also, Figure 1 does not show error bars, which are shown with Figure 2 in the supplementary material (Osogami 2019). Following van Seijen et al. (2016), λ is varied in $\{i/100 \mid i = 0, 10, \ldots, 90, 91, \ldots, 100\}$. We vary α in $\{2^i \mid i = -8, -7, \ldots, 8\}$.

Overall, we find that the performance of Boyan's LSTD(λ) is relatively sensitive to the particular values of λ and α. While Boyan's LSTD(λ) can perform the best with appropriate choice of λ and α, it can perform quite poorly with other choices. On the other hand, Uncorrected LSTD(λ) performs more stably across the range of λ and α, although it may not necessarily perform the best even with the optimal choice of λ and α (see the top panel of Figure 3(a) in the supplementary material (Osogami 2019)). In fact, for all MRPs in Figure 1, Boyan's LSTD(λ) with the optimal choice of λ and α slightly (up to 2 %) outperforms Uncorrected and LSTD(λ) with its optimal choices of λ and α. Then Mixed LSTD(λ) interpolates Boyan's and Uncorrected LSTD(λ). See the supplementary material (Osogami 2019) for computational environment and the running time with our experiments as well as our implementation of the three LSTD(λ)s.

A conclusion from our experiments is that the three LSTD(λ)s perform differently with limited training data (small T), while in theory they converge to the same weights as $T \to \infty$. Small T is relevant for example in reinforcement learning, where policies are iteratively updated, and the training data with the latest policy is often limited. The relative performance of the three LSTD(λ)s depends on the characteristics of the MRPs, but our experiments suggest that Uncorrected and Mixed LSTD(λ) certainly have advantages over Boyan's for *some* MRPs. Mixed LSTD(λ) nicely interpolates Boyan's and Uncorrected LSTD(λ), striking a good balance between bias and variance.

5 Conclusion

We have derived Uncorrected LSTD(λ) in a way that it matches "the Least-Squares method for the linear regression of Monte Carlo return" at $\lambda = 1$. We have shown that Uncorrected LSTD(λ) can have smaller variance than conventional Boyan's LSTD(λ), and this allows Uncorrected LSTD(λ) to outperform Boyan's for some cases, even though Uncorrected LSTD(λ) is generally biased, while Boyan's is proved to be unbiased. To strike a good tradeoff of bias and variance, we have also proposed Mixed LSTD(λ). The three LSTD(λ)s are shown to converge to the common weights as the number of time steps tends to infinity.

Our numerical experiments confirm that the three LSTD(λ)s indeed behave differently with small T. In particular, Boyan's LSTD(λ) tends to perform the best with its optimal choice of hyperparameters but is relatively more sen-

sitive to the particular values of hyperparameters than the other LSTD(λ)s. One may find a well performing LSTD(λ) for a given domain from the family of Mixed LSTD(λ).

Future work includes an application of Mixed or Uncorrected LSTD(λ) to reinforcement learning, where policy evaluation and policy improvement are iterated. Although any LSTD(λ) may be used for policy evaluation, one would prefer the one that can evaluate any policy quickly (with small amount of training data). Our results suggest that our LSTD(λ)s are relatively insensitive to the particular values of hyperparameters even when the amount of training data is limited, which suggests that our LSTD(λ)s, with fixed values of hyperparameters, can quickly and reliably evaluate a wide range of policies.

Acknowledgments

We thank an anonymous reviewer of NeurIPS 2019 for exceptionally detailed and constructive comments on a previous version of this paper.

References

Boyan, J. 2002. Technical update: Least-squares temporal difference learning. *Machine Learning* 49(2–3):233–246.

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

Gehring, C.; Pan, Y.; and White, M. 2016. Incremental truncated LSTD. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 1505–1511.

Geramifard, A.; Bowling, M.; Zinkevich, M.; and Sutton, R. S. 2007. iLSTD: Eligibility traces and convergence analysis. In Schölkopf, B.; Platt, J. C.; and Hoffman, T., eds., *Advances in Neural Information Processing Systems 19*. MIT Press. 441–448.

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

Lin, L.-J. 1993. *Reinforcement Learning for Robots Using Neural Networks*. Ph.D. Dissertation.

Mahmood, A. R.; van Hasselt, H. P.; and Sutton, R. S. 2014. Weighted importance sampling for off-policy learning with linear function approximation. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 3014–3022.

Nedić, A., and Bertsekas, D. P. 2003. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems* 13(1–2):79–110.

Osogami, T. 2019. Supplementary material for uncorrected least-squares temporal difference with lambda-return. *CoRR* abs/1911.06057.

Peng, J., and Williams, R. J. 1996. Incremental multi-step Q-learning. *Machine Learning* 22(1–3):283–290.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, second edition.

Sutton, R. S.; Mahmood, A. R.; and White, M. 2016. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research* 17(73):1–29.

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

Szepesvári, C. 2010. *Algorithms for Reinforcement Learning*. Morgan & Claypool.

Ueno, T.; Maeda, S.; Kawanabe, M.; and Ishii, S. 2011. Generalized TD learning. *Journal of Machine Learning Research* 12:1977–2020.

van Seijen, H., and Sutton, R. S. 2014. True online TD(λ). In *Proceedings of the 31th International Conference on Machine learning (ICML)*, 692–700.

van Seijen, H.; Mahmood, A. R.; Pilarski, P. M.; Machado, M. C.; and Sutton, R. S. 2016. True online temporal-difference learning. *Journal of Machine Learning Research* 17(145):1–40.

Vanseijen, H., and Sutton, R. S. 2015. A deeper look at planning as learning from replay. In *Proceedings of the 32nd International Conference on Machine Learning*, 2314–2322.

Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, King's College, UK.

Xu, X.; He, H.; and Hu, D. 2002. Efficient reinforcement learning using recursive least-squares methods. *Journal of Artificial Intelligence Research* 16:259–292.

Young, P. C. 2011. *Recursive Estimation and Time-series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, second edition.