Derivatives of Logarithmic Stationary Distributions for Policy Gradient Reinforcement Learning

Tetsuro Morimura[†], Eiji Uchibe[††],
Junichiro Yoshimoto[††,†††], Jan Peters[††††], Kenji Doya[††,†††,†††††]

[†] IBM Research, Tokyo Research Laboratory
[††] Initial Research Project, Okinawa Institute of Science and Technology
[†††] Graduate School of Information Science, Nara Institute of Science and Technology
[††††] Max Planck Institute for Biological Cybernetics
[†††††] ATR Computational Neuroscience Laboratories

tetsuro@jp.ibm.com, {uchibe,jun-y}@oist.jp,
jan.peters@tuebingen.mpg.de, doya@oist.jp

Abstract

Most conventional Policy Gradient Reinforcement Learning (PGRL) algorithms neglect (or do not explicitly make use of) a term in the average reward gradient with respect to the policy parameter. That term involves the derivative of the stationary state distribution which corresponds to the sensitivity of its distribution to changes in the policy parameter. Although the bias introduced by this omission can be reduced by setting the forgetting rate γ for the value functions close to 1, these algorithms do not permit γ to be set exactly at $\gamma = 1$. In this paper, we propose a method for estimating the Log Stationary state distribution Derivative (LSD) as a useful form of the derivative of the stationary state distribution through backward Markov chain formulation and a temporal difference learning framework. A new policy gradient (PG) framework with an LSD is also proposed, in which the average reward gradient can be estimated by setting $\gamma = 0$, so it becomes unnecessary to learn the value functions. We also test the performance of

the proposed algorithms using simple benchmark tasks and show that these can improve the performances of existing PG methods.

1 Introduction

Policy Gradient Reinforcement Learning (PGRL) is a popular family of algorithms in Reinforcement Learning (RL). PGRL improves a policy parameter to maximize the average reward (also called the expected return) by using the average reward gradients with respect to the policy parameter, which are called the Policy Gradients (PGs) (Gullapalli, 2000; Williams, 1992; Kimura and Kobayashi, 1998; Baird and Moore, 1999; Sutton et al., 2000; Baxter and Bartlett, 2001; Konda and Tsitsiklis, 2003; Peters and Schaal, 2006). However, most conventional PG algorithms for infinite-horizon problems neglect (or do not explicitly make use of) the term associated with the derivative of the stationary (state) distribution in the PGs with the exception of Ng et al. (2000), since to date there is not an efficient algorithm to estimate this derivative. This derivative is an indicator of how sensitive the stationary distribution is to changes in the policy parameter. While the biases introduced by this omission can be reduced by using a forgetting (or discounting[1]) rate "γ" for the value functions close to 1, that tends to increase the variance of the PG estimates and for $\gamma = 1$ the variance can become infinite which violates the conditions of these algorithms. This tradeoff makes it difficult to find an appropriate γ in practice. Furthermore, while the solution to discounted reinforcement learning is well-defined if the optimal control solution can be perfectly represented by the policy, this is no longer true in the case where function approximation is employed. For approximations of the policies, the solution will

[1]Note that the parameter γ has two different meanings: discounting and forgetting. γ is sometimes interpreted as a discounting rate to define the objective function. On the other hand, the role of γ can be regarded as the forgetting rate to enforce a horizon change for the approach of Baxter and Bartlett (2001), where the objective function is the average reward. That is, while the discounting rate is seen as a part of the problem, the forgetting rate is a part of algorithm. Since we focus on the average reward as the infinite-horizon problem, we use the name the "forgetting" rate for γ in this article.

always be determined by the start-state distributions and, thus, is in general an ill-defined problem. Average reward RL on the other hand is a well-posed problem as it only depends on the stationary distribution.

Here, we propose a new PG framework with estimating the derivative of the logarithmic stationary state distribution (Log Stationary distribution Derivative or LSD) as an alternative and useful form of the derivative of the stationary distribution for estimating the PG[2]. It is our main result and contribution of this paper that a method for estimating the LSD is derived through backward Markov chain formulation and a temporal difference learning method. Then, the learning agent estimates the LSD instead of estimating the value functions in this PG framework. Furthermore, the realization of LSD estimation will open other possibilities for RL. Especially, it will enable us to implement the state of the art natural gradient learning for RL (Morimura et al., 2008a, 2009) which was reported to be effective especially in the randomly synthesized large-scale MDPs. The Fisher information matrix as the Riemannian metric defining this natural gradient included the LSD.

This paper is an extended version of an earlier technical report (Morimura et al., 2007), including new results, and is organized as follows. In Section 2, we review the conventional PGRL methods and describe a motivation to estimate LSD. In Section 3, we propose an $\mathcal{LS}$LSD(λ) algorithm for the estimation of LSD by a $\mathcal{L}$east $\mathcal{S}$quares temporal difference method based on the backward Markov chain formulation. In Section 4, the $\mathcal{LS}$LSD(λ)-PG algorithm is instantly derived as a novel PG algorithm utilizing $\mathcal{LS}$LSD(λ). We also propose a baseline function for $\mathcal{LS}$LSD(λ)-PG which decreases the variance of the PG estimate. To verify the performances of the proposed algorithms, numerical results for simple Markov Decision Processes (MDPs) are shown in Section 5. In Section 6, we review existing (stationary) state distribution derivative estimating and average reward

[2]While the log stationary distribution derivative with respect to the policy parameter is sometimes referred to as the *likelihood ratio gradient* or *score function*, we call it the LSD in this paper.

PG methods. In Section 7, we give a summary and discussion. We also suggest other significant possibilities brought by the realization of the LSD estimation.

2 Policy Gradient Reinforcement Learning

We briefly review the conventional PGRL methods and present the motivation to estimate the LSD. A discrete time MDP with finite sets of states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ is defined by a state transition probability $p(s_{+1} \mid s, a) \equiv \Pr(s_{+1} \mid s, a)$ and a (bounded) reward function $r_{+1} = r(s, a, s_{+1})$, where s_{+1} is the state followed by the action a at the state s and r_{+1} is the observed immediate reward at s_{+1} (Bertsekas, 1995; Sutton and Barto, 1998). The state s_{+k} and the action a_{+k} denote a state and an action after k time steps from the state s and the action a, respectively, and backwards for $-k$. The decision-making rule follows a stochastic policy $\pi(s, a; \boldsymbol{\theta}) \equiv \Pr(a \mid s, \boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta} \in \mathcal{R}^d$. We assume the policy $\pi(s, a; \boldsymbol{\theta})$ is always differentiable with respect to $\boldsymbol{\theta}$. We also posit the following assumption:

Assumption 1 *The Markov chain $M(\boldsymbol{\theta}) = \{\mathcal{S}, \mathcal{A}, p, \pi, \boldsymbol{\theta}\}$ is ergodic (irreducible and aperiodic) for all policy parameters $\boldsymbol{\theta}$. Then there exists a unique stationary state distribution $d_{M(\boldsymbol{\theta})}(s) = \lim_{k\to\infty} \Pr(s_{+k} = s \mid M(\boldsymbol{\theta})) > 0$ which is independent of the initial state and satisfies the recursion*

$$d_{M(\boldsymbol{\theta})}(s) = \sum_{s_{-1} \in \mathcal{S}} \sum_{a_{-1} \in \mathcal{A}} p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1}) d_{M(\boldsymbol{\theta})}(s_{-1}), \tag{1}$$

where $p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1}) \equiv \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) p(s \mid s_{-1}, a_{-1})$.

The goal of PGRL is to find a policy parameter $\boldsymbol{\theta}^*$ that maximizes the time average of immediate rewards called the *average reward* or *expected return*:

$$\eta(\boldsymbol{\theta}) \equiv \lim_{K\to\infty} \frac{1}{K} \mathbb{E}_{M(\boldsymbol{\theta})} \Big\{ \sum_{k=1}^{K} r_{+k} \,|\, s \Big\},$$

where $\mathbb{E}_{M(\boldsymbol{\theta})}$ denotes the expectation over the Markov chain $M(\boldsymbol{\theta})$. Under Assumption 1, the average reward is independent of the initial state s and can be shown to be equal to (Bertsekas, 1995)

$$\begin{aligned}\eta(\boldsymbol{\theta}) &= \mathbb{E}_{M(\boldsymbol{\theta})}\{r(s,a,s_{+1})\} \\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a)r(s,a,s_{+1}) \\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\bar{r}(s,a), \end{aligned} \tag{2}$$

where $\bar{r}(s,a) = \sum_{s_{+1}\in\mathcal{S}} p(s_{+1} \mid s,a)r(s,a,s_{+1})$ does not depend on the policy parameter. The policy gradient RL algorithms update the policy parameters $\boldsymbol{\theta}$ in the direction of the gradient of the average reward $\eta(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ so that

$$\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) \equiv \left[\frac{\partial\eta(\boldsymbol{\theta})}{\partial\theta_1},\ldots,\frac{\partial\eta(\boldsymbol{\theta})}{\partial\theta_d}\right]^{\top},$$

where $\top$ denotes the transpose. This derivative is often referred to as the policy gradient (PG) for short. Using Eq.2, the PG is directly determined to be

$$\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\left(\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s) + \nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})\right)\bar{r}(s,a). \tag{3}$$

However, since the derivation of the gradient of the log stationary state distribution $\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)$ is nontrivial, the conventional PG algorithms (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998) utilize an alternative representation of the PG as [3]

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} & d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})Q^{\pi}_{\gamma}(s,a) \\ & + (1-\gamma)\sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)V^{\pi}_{\gamma}(s), \end{aligned} \tag{4}$$

[3] We give this derivation in Appendix.

where

$$Q_\gamma^\pi(s,a) \equiv \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{\sum_{k=1}^{K} \gamma^{k-1} r_{+k} \mid s,a\},$$
$$V_\gamma^\pi(s) \equiv \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{\sum_{k=1}^{K} \gamma^{k-1} r_{+k} \mid s\}$$

are the cumulative rewards defined with forgetting rate $\gamma \in [0,1)$, known as the state-action and state value functions, respectively (Sutton and Barto, 1998). Since the contribution of the second term in Eq.4 shrinks as γ approaches 1 (Baxter and Bartlett, 2001), the conventional algorithms omitted the second term and approximated the PG as a biased estimate by taking $\gamma \approx 1$: the PG estimate was composed of only the first term in Eq.4. Although the bias introduced by this omission shrinks as γ approaches to 1, the variance of the estimate becomes larger (Baxter and Bartlett, 2001; Baxter et al., 2001). In addition, these algorithms prohibit γ from being set exactly at 1 though the bias disappears in the limit of $\gamma \to 1$.

In this paper, we propose an estimation approach for the log stationary distribution derivative (LSD), $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. The realization of the LSD estimation instantly makes an alternative PG approach feasible, which uses Eq.3 for the computation of the PG estimate with the LSD estimate. An important feature is that this approach does not need to learn either value function, Q_γ^π or V_γ^π, and therefore the resulting algorithms are free from the bias-variance trade-off in the choice of the forgetting rate γ.

3 Estimation of the Log Stationary Distribution Derivative (LSD)

In this section, we propose an LSD estimation algorithm based on least squares, $\mathcal{LS}$LSD(λ). For this purpose, we formulate the backwardness of the ergodic

Markov chain $M(\boldsymbol{\theta})$, and show that LSD can be estimated on the basis of the temporal difference learning (Sutton, 1988; Bradtke and Barto, 1996; Boyan, 2002).

3.1 Properties of forward and backward Markov chains

According to Bayes' theorem, a backward probability q of a previous state-action pair (s_{-1}, a_{-1}) leading to the current state s is given by

$$q(s_{-1}, a_{-1} \mid s) = \frac{p(s \mid s_{-1}, a_{-1}) \Pr(s_{-1}, a_{-1})}{\sum_{s_{-1} \in \mathcal{S}} \sum_{a_{-1} \in \mathcal{A}} p(s \mid s_{-1}, a_{-1}) \Pr(s_{-1}, a_{-1})}.$$

The posterior $q(s_{-1}, a_{-1} \mid s)$ depends upon the prior distribution $\Pr(s_{-1}, a_{-1})$. When the prior distribution follows a stationary distribution under a fixed policy π, i.e., $\Pr(s_{-1}, a_{-1}) = \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) d_{M(\boldsymbol{\theta})}(s_{-1})$, the posterior is called the *stationary* backward probability denoted by $q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s)$ and satisfies

$$\begin{aligned} q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s) &= \frac{p(s \mid s_{-1}, a_{-1}) \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) d_{M(\boldsymbol{\theta})}(s_{-1})}{d_{M(\boldsymbol{\theta})}(s)} \\ &= \frac{p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1}) d_{M(\boldsymbol{\theta})}(s_{-1})}{d_{M(\boldsymbol{\theta})}(s)}. \end{aligned} \tag{5}$$

If a Markov chain follows $q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s)$, we call it the backward Markov chain $B(\boldsymbol{\theta})$ associated with $M(\boldsymbol{\theta})$ following $p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1})$. Both Markov chains—$M(\boldsymbol{\theta})$ and $B(\boldsymbol{\theta})$—are closely related as described in the following two propositions:

Proposition 1 *Let a Markov chain $M(\boldsymbol{\theta})$ characterized by a transition probability $p_{M(\boldsymbol{\theta})}(s \mid s_{-1}) \equiv \sum_{a_{-1} \in \mathcal{A}} p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1})$, which is irreducible and ergodic. Then the backward Markov chain $B(\boldsymbol{\theta})$ characterized by the backward (stationary) transition probability $q_{B(\boldsymbol{\theta})}(s_{-1} \mid s) \equiv \sum_{a_{-1} \in \mathcal{A}} q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s)$ with respect to $p_{M(\boldsymbol{\theta})}$ is also ergodic and has the same unique stationary distribution as $M(\boldsymbol{\theta})$:*

$$d_{M(\boldsymbol{\theta})}(s) = d_{B(\boldsymbol{\theta})}(s), \qquad {}^{\forall} s \in \mathcal{S}, \tag{6}$$

where $d_{M(\boldsymbol{\theta})}(s)$ and $d_{B(\boldsymbol{\theta})}(s)$ are the stationary distributions of $M(\boldsymbol{\theta})$ and $B(\boldsymbol{\theta})$, respectively.

Proof: By multiplying both sides of Eq.5 by $d_{M(\boldsymbol{\theta})}(s)$ and summing over all possible $a_{-1} \in \mathcal{A}$, we obtain a "detailed balance equations" (MacKay, 2003)

$$q_{B(\boldsymbol{\theta})}(s_{-1} \mid s) d_{M(\boldsymbol{\theta})}(s) = p_{M(\boldsymbol{\theta})}(s \mid s_{-1}) d_{M(\boldsymbol{\theta})}(s_{-1}), \quad {}^{\forall}s_{-1} \in \mathcal{S}, {}^{\forall}s \in \mathcal{S}. \tag{7}$$

Then

$$\sum_{s \in \mathcal{S}} q_{B(\boldsymbol{\theta})}(s_{-1} \mid s) d_{M(\boldsymbol{\theta})}(s) = d_{M(\boldsymbol{\theta})}(s_{-1})$$

holds by summing both sides of Eq.7 over all possible $s \in \mathcal{S}$, indicating that (i) $B(\boldsymbol{\theta})$ has the same stationary distribution as $M(\boldsymbol{\theta})$. By Assumption 1, (i) directly proves that (ii) $B(\boldsymbol{\theta})$ is irreducible. Eq.7 is reformulated by the matrix notation: both transition probabilities $p_{M(\boldsymbol{\theta})}(s \mid s_{-1})$ and $q_{B(\boldsymbol{\theta})}(s_{-1} \mid s)$ are assembled into $\boldsymbol{P}_{M(\boldsymbol{\theta})}$ and $\boldsymbol{Q}_{B(\boldsymbol{\theta})}$, respectively[4], and the stationary distribution into $\boldsymbol{d}_{\boldsymbol{\theta}}$:[5]

$$\boldsymbol{Q}_{B(\boldsymbol{\theta})} = \mathrm{diag}(\boldsymbol{d}_{\boldsymbol{\theta}})^{-1} \boldsymbol{P}_{M(\boldsymbol{\theta})}^{\top} \mathrm{diag}(\boldsymbol{d}_{\boldsymbol{\theta}}).$$

We can easily see that the diagonal components of $(\boldsymbol{P}_{M(\boldsymbol{\theta})})^n$ are equal to those of $(\boldsymbol{Q}_{B(\boldsymbol{\theta})})^n$ for any natural number n. This implies that (iii) $B(\boldsymbol{\theta})$ has the same aperiodic property as $M(\boldsymbol{\theta})$. Proposition 1 (Eq.6) is directly proven by (i)–(iii) (Schinazi, 1999). □

Proposition 2 *Let the distribution of s_{-K} follow $d_{M(\boldsymbol{\theta})}(s)$ and let $f(s_k, a_k)$ be an*

[4]The bold $\boldsymbol{Q}_{B(\boldsymbol{\theta})}$ has no relationship with the state-action value function $Q^{\pi}(s, a)$

[5]The function "diag($\boldsymbol{a}$)" for a vector $\boldsymbol{a} \in \mathcal{R}^d$ denotes the diagonal matrix of $\boldsymbol{a}$, so $\mathrm{diag}(\boldsymbol{a}) \in \mathcal{R}^{d \times d}$.

arbitrary function of a state-action pair at time k. Then

$$\mathbb{E}_{B(\boldsymbol{\theta})}\left\{\sum_{k=1}^{K} f(s_{-k}, a_{-k}) \mid s\right\} = \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\sum_{k=1}^{K} f(s_{-k}, a_{-k}) \mid s, d_{M(\boldsymbol{\theta})}(s_{-K})\right\} \tag{8}$$
$$= \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\sum_{k=0}^{K-1} f(s_{+k}, a_{+k}) \mid s_{+K}, d_{M(\boldsymbol{\theta})}(s)\right\},$$

where $\mathbb{E}_{B(\boldsymbol{\theta})}$ and $\mathbb{E}_{M(\boldsymbol{\theta})}$ are the expectations over the backward and forward Markov chains, $B(\boldsymbol{\theta})$ and $M(\boldsymbol{\theta})$, respectively, and $\mathbb{E}\{\cdot \mid d_{M(\boldsymbol{\theta})}(s)\} \equiv \mathbb{E}\{\cdot \mid \Pr(s) = d_{M(\boldsymbol{\theta})}(s)\}$. Eq.8 holds even at the limit $K \to \infty$.

Proof: By utilizing the Markov property and substituting Eq.5, we have the following relationship

$$\begin{aligned}
&q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1}, ..., s_{-K}, a_{-K} \mid s) \\
&\quad = q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s) \cdots q_{B(\boldsymbol{\theta})}(s_{-K}, a_{-K} \mid s_{-K+1}) \\
&\quad = \frac{p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1}) \cdots p_{M(\boldsymbol{\theta})}(s_{-K+1}, a_{-K} \mid s_{-K}) d_{M(\boldsymbol{\theta})}(s_{-K})}{d_{M(\boldsymbol{\theta})}(s)} \\
&\quad \propto p_{M(\boldsymbol{\theta})}(s, a_{-1} \mid s_{-1}) \cdots p_{M(\boldsymbol{\theta})}(s_{-K+1}, a_{-K} \mid s_{-K}) d_{M(\boldsymbol{\theta})}(s_{-K}).
\end{aligned}$$

This directly implies the proposition in the case of finite K. Since the following equations are derived from Proposition 1, the proposition in the limit case $K \to \infty$ is also instantly proven,

$$\begin{aligned}
\lim_{K\to\infty} \mathbb{E}_{B(\boldsymbol{\theta})}\{f(s_{-K}, a_{-K}) \mid s\} &= \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{f(s_{-K}, a_{-K}) \mid s, d_{M(\boldsymbol{\theta})}(s_{-K})\} \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(s, a; \boldsymbol{\theta}) d_{M(\boldsymbol{\theta})}(s) f(s, a).
\end{aligned}$$

□

Propositions 1 and 2 are significant as they indicate that the samples from the forward Markov chain $M(\boldsymbol{\theta})$ can be used directly for estimating the statistics of the backward Markov chain $B(\boldsymbol{\theta})$.

3.2 Temporal difference learning for LSD from the backward to forward Markov chains

Using Eq.5, the LSD, $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$, can be decomposed into

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) &= \frac{1}{d_{M(\boldsymbol{\theta})}(s)} \sum_{s_{-1} \in \mathcal{S}} \sum_{a_{-1} \in \mathcal{A}} p(s \mid s_{-1}, a_{-1}) \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) d_{M(\boldsymbol{\theta})}(s_{-1}) \\ &\qquad \left\{ \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1}) \right\} \\ &= \sum_{s_{-1} \in \mathcal{S}} \sum_{a_{-1} \in \mathcal{A}} q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1} \mid s) \{ \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1}) \} \\ &= \mathbb{E}_{B(\boldsymbol{\theta})} \{ \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1}) \mid s \}. \end{aligned} \tag{9}$$

Noting that there exist $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ and $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1})$ in Eq.9, the recursion of Eq.9 can be rewritten as

$$\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) = \lim_{K \to \infty} \mathbb{E}_{B(\boldsymbol{\theta})} \left\{ \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-k}, a_{-k}; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-K}) \mid s \right\}. \tag{10}$$

Eq.10 implies that the LSD of a state s is the infinite-horizon cumulation of the Log Policy distribution Derivative (LPD), $\nabla_{\boldsymbol{\theta}} \ln \pi(s, a; \boldsymbol{\theta})$, along the backward Markov chain $B(\boldsymbol{\theta})$ from state s. From Eqs. 9 and 10, LSD could be estimated using an approach which is similar to temporal difference (TD) learning (Sutton, 1988) for the following backward TD-error $\boldsymbol{\delta}$ on the backward Markov chain $B(\boldsymbol{\theta})$ rather than $M(\boldsymbol{\theta})$.

$$\boldsymbol{\delta}(s) \equiv \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-1}, a_{-1}; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1}) - \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s),$$

where the first two terms of the right hand side describe the one-step actual observation of the policy eligibility and the one-step ahead LSD on $B(\boldsymbol{\theta})$, respectively, while the last term is the current LSD. Interestingly, while the well-known TD-error for the value-function estimation uses the reward $r(s, a, s_{+1})$ on $M(\boldsymbol{\theta})$ (Sutton and Barto, 1998), this TD-error for the LSD estimation uses

$\nabla_{\boldsymbol{\theta}} \ln \pi(s_{-1}, a_{-1}; \boldsymbol{\theta})$ on $B(\boldsymbol{\theta})$.

While $\boldsymbol{\delta}(s)$ is a random variable, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}(s) \mid s\} = \mathbf{0}$ holds for all states $s \in \mathcal{S}$, which comes from Eq.9. This motivates to minimize the mean squares of the backward TD-error, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}(s)\|^2\}$, for the estimation of LSD[6], where $\hat{\boldsymbol{\delta}}(s)$ is composed of the LSD estimate $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ rather than (exact) LSD $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. Here, $\|\boldsymbol{a}\|$ denotes the Euclidean norm $(\boldsymbol{a}^\top \boldsymbol{a})^{1/2}$.

With an eligibility decay rate $\lambda \in [0, 1]$ and a back-trace time-step $K \in \mathcal{N}$, Eq.10 is generalized, where $\mathcal{N}$ denotes the set of natural numbers:

$$\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) = \mathbb{E}_{B(\boldsymbol{\theta})}\Bigg\{ \sum_{k=1}^{K} \lambda^{k-1} \big\{ \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-k}, a_{-k}; \boldsymbol{\theta}) + (1-\lambda) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-k}) \big\} + \lambda^K \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-K}) \mid s \Bigg\}.$$

Accordingly, the backward TD is extended into the backward TD(λ), $\boldsymbol{\delta}_{\lambda,K}(s)$,

$$\boldsymbol{\delta}_{\lambda,K}(s) \equiv \sum_{k=1}^{K} \lambda^{k-1} \Big\{ \nabla_{\boldsymbol{\theta}} \ln \pi(s_{-k}, a_{-k}; \boldsymbol{\theta}) + (1-\lambda) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-k}) \Big\} + \lambda^K \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-K}) - \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s),$$

where the unbiased property, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}_{\lambda,K}(s) \mid s\} = \mathbf{0}$, is still retained. The minimization of $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2\}$ at $\lambda = 1$ and the limit $K \to \infty$ is regarded as the Widrow-Hoff supervised learning procedure. Even if λ and K are not set in the above values, the minimization of $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2\}$ in a large $\lambda \in [0, 1)$ and K would be less sensitive to a non-Markovian effect as in the case of the conventional TD(λ) learning for the value functions (Peng and Williams, 1996).

In order to minimize $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2\}$ as the estimation of the LSD, we need to gather many samples drawn from the backward Markov chain $B(\boldsymbol{\theta})$. However,

[6]Actually, the classical least squares approach to $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}(s)\|^2\}$ would make the LSD estimate biased, because $\hat{\boldsymbol{\delta}}(s)$ has the different time-step LSDs, $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_{-1})$ and $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. Instead, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\iota}(s)^\top \hat{\boldsymbol{\delta}}(s)\}$ is minimized for an unbiased LSD estimation, where $\boldsymbol{\iota}(s)$ is a instrumental variable (Young, 1984; Bradtke and Barto, 1996). Such a detailed discussion is given in Section 3.3. Before that, we see only $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}(s)\|^2\}$ to enhance the readability.

the actual samples are drawn from a forward Markov chain $M(\boldsymbol{\theta})$. Fortunately, by using Propositions 1 and 2, we can derive the following exchangeable property:

$$\begin{aligned}\mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2\Big\} &= \sum_{s\in\mathcal{S}} d_{B(\boldsymbol{\theta})}(s)\,\mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2 \mid s\Big\} \\ &= \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\,\mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2 \mid s, d_{M(\boldsymbol{\theta})}(s_{-K})\Big\} \\ &= \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2 \mid d_{M(\boldsymbol{\theta})}(s_{-K})\Big\}. \qquad (11)\end{aligned}$$

In particular, the actual samples can be reused to minimize $\mathbb{E}_{B(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2\}$, provided $s_{-K} \sim d_{M(\boldsymbol{\theta})}(s)$. In real problems, however, the initial state is rarely distributed according to the stationary distribution $d_{M(\boldsymbol{\theta})}(s)$. To interpolate the gap between theoretical assumption and realistic applicability, we would need to adopt either of the following two strategies: (i) K is not set at such a large integer if $\lambda \approx 1$; (ii) λ is not set at 1 if $K \approx t$, where t is the current time-step of the actual forward Markov chain $M(\boldsymbol{\theta})$.

3.3 LSD estimation algorithm: Least squares on backward TD(λ) with constraint

In the previous sections, we introduced the theory for estimating LSD by the minimization of the mean squares of $\hat{\boldsymbol{\delta}}_{\lambda,K}(s)$ on $M(\boldsymbol{\theta})$, $\mathbb{E}_{M(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_{\lambda,K}(s)\|^2 \mid d_{M(\boldsymbol{\theta})}(s_{-K})\}$. However, LSD also has the following constraint derived from $\sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) = 1$:

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)\} = \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) = \nabla_{\boldsymbol{\theta}} \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) = \mathbf{0}. \qquad (12)$$

In this section, we propose an LSD estimation algorithm, $\mathcal{LS}$LSD(λ), based on least squares temporal difference approach (Young, 1984; Bradtke and Barto, 1996; Boyan, 2002; Lagoudakis and Parr, 2003), which simultaneously attempts to decrease the mean squares and satisfy the constraint. We consider the situation where the LSD estimate $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ is represented by a linear vector function

approximator

$$\boldsymbol{f}(s;\boldsymbol{\Omega}) \equiv \boldsymbol{\Omega}\boldsymbol{\phi}(s), \tag{13}$$

where $\boldsymbol{\phi}(s) \in \mathcal{R}^e$ is a basis function and $\boldsymbol{\Omega} \equiv [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_d]^\top \in \mathcal{R}^{d\times e}$ is an adjustable parameter matrix, and we assume that the optimal parameter $\boldsymbol{\Omega}^*$ satisfies $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) = \boldsymbol{\Omega}^*\boldsymbol{\phi}(s)$. If the estimator cannot represent the LSD exactly, $\mathcal{LS}$LSD(λ) would behave as suggested by Sutton (1988); Peng and Williams (1996), which means the estimation error for the LSD would get smaller as the value of $\lambda \in [0, 1)$ approaches 1. This will be confirmed in our numerical experiments (Section 5).

For simplicity, we focus only on the i-th element θ_i of the policy parameter $\boldsymbol{\theta}$, denoting $f(s;\boldsymbol{\omega}_i) \equiv \boldsymbol{\omega}_i^\top\boldsymbol{\phi}(s)$, $\nabla_{\theta_i} \ln \pi(s, a;\boldsymbol{\theta}) \equiv \partial \ln \pi(s, a;\boldsymbol{\theta})/\partial\theta_i$, and $\hat{\delta}_{\lambda,K}(s, \boldsymbol{\omega}_i)$ as the i-th element of $\hat{\boldsymbol{\delta}}_{\lambda,K}(s, \boldsymbol{\Omega})$. Accordingly, the objective function to be minimized is given by

$$\varepsilon(\boldsymbol{\omega}_i) = \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)^2 \mid d_{M(\boldsymbol{\theta})}(s_{-K})\} + \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(s;\boldsymbol{\omega}_i)\}^2, \tag{14}$$

where the second term of the right side comes from the constraint of Eq.12[7]. Thus, the derivative is

$$\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) = \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)\ \nabla_{\omega_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i) \mid d_{M(\boldsymbol{\theta})}(s_{-K})\} + \nabla_{\boldsymbol{\omega}_i}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(s;\boldsymbol{\omega}_i)\}, \tag{15}$$

where

$$\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i) = \sum_{k=1}^{K} \lambda^{k-1}\nabla_{\theta_i} \ln \pi(s_{-k}, a_{-k};\boldsymbol{\theta}) + \boldsymbol{\omega}_i^\top\ \nabla_{\omega_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)$$

[7]While $\mathcal{LS}$LSD(λ) weighs the two objectives equally, we can instantly extend it to the problem minimizing $\mathbb{E}_{M(\boldsymbol{\theta})}\{\|\hat{\boldsymbol{\delta}}_\lambda(x)\|^2 \mid d^\pi(s_{-K})\}$ subject to the constraint of Eq.12 with the Lagrange multiplier method.

and

$$\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i) = (1-\lambda)\sum_{k=1}^{K}\lambda^{k-1}\boldsymbol{\phi}(s_{-k}) + \lambda^K\boldsymbol{\phi}(s_{-K}) - \boldsymbol{\phi}(s).$$

Although the conventional least squares method aims to find the parameter satisfying $\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) = \mathbf{0}$ as the true parameter $\boldsymbol{\omega}_i^*$, it induces estimation bias if a correlation exists between the error $\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ and its derivative $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ concerning the first term of the right-hand side in Eq.14. That is, if

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)\ \nabla_{\omega_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*) \mid d_{M(\boldsymbol{\theta})}(s_{-K})\} \neq \mathbf{0},$$

$\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i^*) \neq \mathbf{0}$ holds because $\nabla_{\boldsymbol{\omega}_i}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(s;\boldsymbol{\omega}_i^*)\}$ comprised in the second term of the left-hand side in Eq.14 is always a zero vector due to Eq.12 and $f(s;\boldsymbol{\omega}_i^*) = \nabla_{\theta_i}\ln d_{M(\boldsymbol{\theta})}(s)$. Since this correlation exists in general RL tasks, we apply the instrumental variable method to eliminate the bias (Young, 1984; Bradtke and Barto, 1996). This requires that $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)$ be replaced by the instrumental variable $\boldsymbol{\iota}(s)$ which has a correlation with $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ but not $\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$. This condition is obviously satisfied when $\boldsymbol{\iota}(s) = \boldsymbol{\phi}(s)$ as well as LSTD(λ) (Bradtke and Barto, 1996; Boyan, 2002). Instead of Eq.15, we aim to find the parameter making the equation

$$\widetilde{\nabla}_{\boldsymbol{\omega}_i}\varepsilon(\omega_i) \equiv \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)\boldsymbol{\phi}(s) \mid d_{M(\boldsymbol{\theta})}(s_{-K})\} + \mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(s)\}\mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(s)\}^\top\boldsymbol{\omega}_i \tag{16}$$

be equal to zero, in order to compute the true parameter $\boldsymbol{\omega}_i^*$, so that $\widetilde{\nabla}_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i^*) = \mathbf{0}$.

For the remainder of this paper, we denote the current state at time-step t by s_t to clarify the time course on the actual Markov chain $M(\boldsymbol{\theta})$. In the proposed LSD estimation algorithm, $\mathcal{LS}$LSD(λ), the back-trace time-step K is set equal to the time-step t of the current state s_t while the eligibility decay rate λ can be set

in $[0, 1)$. That is,

$$\hat{\delta}_{\lambda,K}(s_t; \boldsymbol{\omega}_i) = g_{\lambda,i}(s_{t-1}) + (\boldsymbol{z}_\lambda(s_{t-1}) - \boldsymbol{\phi}(s_t))^\top \boldsymbol{\omega}_i,$$

where $g_{\lambda,i}(s_t) = \sum_{k=0}^{t} \lambda^{t-k} \nabla_{\theta_i} \ln \pi(s_k, a_k; \boldsymbol{\theta})$ and $\boldsymbol{z}_\lambda(s_t) = (1-\lambda) \sum_{k=1}^{t} \lambda^{t-k} \boldsymbol{\phi}(s_k) + \lambda^t \boldsymbol{\phi}(s_0)$. The expectations in Eq.16 are estimated without bias by (Bradtke and Barto, 1996; Boyan, 2002)

$$\begin{aligned}
&\lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s; \boldsymbol{\omega}_i)\boldsymbol{\phi}(s) \mid d_{M(\boldsymbol{\theta})}(s_{-K})\} \\
&\qquad \simeq \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\phi}(s_t)\{g_{\lambda,i}(s_{t-1}) - (\boldsymbol{\phi}(s_t) - \boldsymbol{z}_\lambda(s_{t-1}))^\top \boldsymbol{\omega}_i\} \\
&\qquad = \boldsymbol{b}_T - \boldsymbol{A}_T\, \boldsymbol{\omega}_i,
\end{aligned}$$

where $\boldsymbol{b}_T \equiv \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\phi}(s_t)\, g_{\lambda,i}(s_{t-1})$ and $\boldsymbol{A}_T \equiv \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\phi}(s_t)(\boldsymbol{\phi}(s_t) - \boldsymbol{z}_\lambda(s_{t-1}))^\top$, and

$$\begin{aligned}
\mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(x)\} &\simeq \frac{1}{T+1}\sum_{t=0}^{T} \boldsymbol{\phi}(s_t) \\
&\equiv \boldsymbol{c}_T.
\end{aligned}$$

Therefore, by substituting these estimators into Eq.16, the estimate $\hat{\boldsymbol{\omega}}_i^*$ at time-step T is computed as

$$\begin{aligned}
&\boldsymbol{b}_T - \boldsymbol{A}_T \hat{\boldsymbol{\omega}}_i^* + \boldsymbol{c}_t \boldsymbol{c}_T^\top \hat{\boldsymbol{\omega}}_i^* = \boldsymbol{0} \\
\Leftrightarrow \quad &\hat{\boldsymbol{\omega}}_i^* = (\boldsymbol{A}_T - \boldsymbol{c}_T \boldsymbol{c}_T^\top)^{-1}\, \boldsymbol{b}_T\,.
\end{aligned}$$

The $\mathcal{LS}$LSD(λ) for the matrix parameter $\hat{\boldsymbol{\Omega}}^*$ rather than $\hat{\boldsymbol{\omega}}_i^*$ is shown in Algorithm 1, where the notation := denotes the right-to-left substitution[8].

[8]Incidentally, although there is calculation of an inverse matrix in the Algorithms, a pseudo-inverse matrix may be used instead of direct calculation of the inverse matrix so as to secure stability in numeric calculation.

Algorithm 1
$\mathcal{LS}$LSD(λ): Estimation for $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$

Given:
- a policy $\pi(s, a; \boldsymbol{\theta})$ with a fixed $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(s)$.

Initialize: $\lambda \in [0, 1)$.
Set: $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{z} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{g} := \boldsymbol{0}$; $\boldsymbol{A} := \boldsymbol{0}$; $\boldsymbol{B} := \boldsymbol{0}$.
  for $t = 0$ **to** $T - 1$ **do**
    $\boldsymbol{c} := \boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
    $\boldsymbol{g} := \lambda \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta})$;
    $\boldsymbol{A} := \boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
    $\boldsymbol{B} := \boldsymbol{B} + \boldsymbol{\phi}(s_{t+1})\boldsymbol{g}^\top$;
    $\boldsymbol{z} := \lambda \boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(s_{t+1})$;
  end for
  $\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top / t)^{-1} \boldsymbol{B}$;
Return: $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) = \boldsymbol{\Omega}\, \boldsymbol{\phi}(s)$.

It is intriguing that $\mathcal{LS}$LSD(λ) has a relationship to a model-based method, as noted by Boyan (2002); Lagoudakis and Parr (2003) in the references for LSTD(λ) and LSTD$Q(\lambda)$, but $\mathcal{LS}$LSD(λ) is concerned with the "backward" model $B(\boldsymbol{\theta})$ instead of the forward model $M(\boldsymbol{\theta})$. This is due to the fact that the sufficient statistics $\boldsymbol{A}$ in $\mathcal{LS}$LSD(λ) can be regarded as a compressed "backward" model, since $\boldsymbol{A}$ is equivalent to one of the sufficient statistics to estimate the backward state transition probability $q_{B(\boldsymbol{\theta})}(s_{-1} \mid s)$ when $\lambda = 0$ and the feature vector $\boldsymbol{\phi}$ corresponding to $\boldsymbol{\phi}(1) = (1, 0, ..., 0)$; $\boldsymbol{\phi}(2) = (0, 1, ..., 0)$; etc. We give the detail explanation about it in Appendix.

4 Policy gradient algorithms with the LSD estimate

We propose a PG algorithm as a straightforward application with the LSD estimates in Section 4.1. In Section 4.2, we introduce baseline functions to reduce the variace of the PG estimated by our PG algorithm.

4.1 Policy update with the LSD estimate

Now let us define the PGRL algorithm based on the LSD estimate. The realization of the estimation for $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ by $\mathcal{LS}\mathrm{LSD}(\lambda)$ directly leads to the following estimate for the PG (Eq.3), due to its independence from the forgetting factor γ for the value functions:

$$\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) \simeq \frac{1}{T} \sum_{t=0}^{T-1} \big(\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_t) \big) r_{t+1} \tag{17}$$

$$\simeq \frac{1}{T} \sum_{t=0}^{T-1} \big(\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta}) + \widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_t) \big) r_{t+1}, \tag{18}$$

where r_{t+1} is the immediate reward defined by the reward function $r(s_t, a_t, s_{t+1})$. The policy parameter can then be updated through the stochastic gradient method with an appropriate step-size α (Bertsekas and Tsitsiklis, 1996):[9]

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha (\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta}) + \widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s_t)) r_{t+1}.$$

$\mathcal{LS}\mathrm{LSD}(\lambda)$-PG without a baseline function is shown in Algorithm 2 as one of the simplest realizations of PG algorithm that uses $\mathcal{LS}\mathrm{LSD}(\lambda)$. In Algorithm 2, the forgetting rate parameter $\beta \in [0, 1)$ is introduced to discard the past estimates given by old values of $\boldsymbol{\theta}$.

[9] Alternatively, $\boldsymbol{\theta}$ can also be updated through the bath gradient method: $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha \widehat{\nabla}_{\boldsymbol{\theta}} R(\boldsymbol{\theta})$.

Algorithm 2
$\mathcal{LS}$LSD(λ)-PG: Optimization for the policy without baseline function

Given:
- a policy $\pi(s, a; \boldsymbol{\theta})$ with an adjustable $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(s)$.

Initialize: $\boldsymbol{\theta}$, $\lambda \in [0, 1)$, $\beta \in [0, 1]$, α_t.
Set: $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{z} = \boldsymbol{\phi}(s_0)$; $\boldsymbol{g} := \boldsymbol{0}$; $\boldsymbol{A} := \boldsymbol{0}$; $\boldsymbol{B} := \boldsymbol{0}$.
for $t = 0$ **to** $T - 1$ **do**
$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t \{\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta}) + \boldsymbol{\Omega}^\top \boldsymbol{\phi}(s_t)\} r_{t+1}$;
$\boldsymbol{c} := \beta \boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
$\boldsymbol{g} := \beta \lambda \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta})$;
$\boldsymbol{A} := \beta \boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
$\boldsymbol{B} := \beta \boldsymbol{B} + \boldsymbol{\phi}(s_{t+1}) \boldsymbol{g}^\top$;
$\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top / \|\boldsymbol{c}\|)^{-1} \boldsymbol{B}$;
$\boldsymbol{z} := \lambda \boldsymbol{z} + (1 - \lambda) \boldsymbol{\phi}(s_{t+1})$;
end for
Return: $p(a \mid s; \boldsymbol{\theta}) = \pi(s, a; \boldsymbol{\theta})$.

An other important topic for function approximation is the choice of the basis function $\boldsymbol{\phi}(s)$ of the approximator, particularly in continuous state problems. For the PG algorithm, the objective of the LSD estimate is just to provide one term of the PG estimate, such as $\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \pi(s, a; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) \bar{r}(s, a)$, but not to provide a precise estimate of the LSD $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$, where $\bar{r}(s, a) \equiv \sum_{s_{+1} \in \mathcal{S}} p(s_{+1} \mid s, a) r(s, a, s_{+1})$. Therefore, the following proposition would be useful:

Proposition 3 *Let the basis function of the LSD estimator be*

$$\phi(s) = \sum_{a \in \mathcal{A}} \pi(s, a; \boldsymbol{\theta}) \bar{r}(s, a),$$

and then the function estimator, $\boldsymbol{f}(s; \boldsymbol{\omega}) = \boldsymbol{\omega} \sum_{a \in \mathcal{A}} \pi(s, a; \boldsymbol{\theta}) \bar{r}(s, a)$*, can represent the second term of the PG,* $\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \pi(s, a; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) \bar{r}(s, a)$*,*

where the adjustable parameter $\boldsymbol{\omega}$ is a d dimensional vector:

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\bar{r}(s,a)\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\bar{r}(s,a)\boldsymbol{f}(s;\boldsymbol{\omega}^\star),$$

where $\boldsymbol{\omega}^\star$ minimizes the mean error, $\epsilon(\boldsymbol{\omega}) = \frac{1}{2}\sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\{\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s) - \boldsymbol{f}(s;\boldsymbol{\omega})\}^2$.

Proof: The proposition follows directly from

$$\nabla_{\boldsymbol{\omega}}\epsilon(\boldsymbol{\omega}^\star) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})\bar{r}(s,a)\{\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s) - \boldsymbol{f}(s;\boldsymbol{\omega}^\star)\} = \mathbf{0}.$$

□

4.2 Baseline function for variance reduction of policy gradient estimates with LSD

As the variance of the PG estimates using the LSD, Eq.18, might be large, we consider variance reduction using a baseline function for immediate reward r. The following proposition provides the kind of functions that can be used as the baseline function for PG estimation using the LSD[10].

Proposition 4 *With the following function of the state s and the following state s_{+1} on $M(\boldsymbol{\theta})$,*

$$\rho(s, s_{+1}) = c + g(s) - g(s_{+1}), \tag{19}$$

where c and $g(s)$ are an arbitrary constant and an arbitrary bounded function of the state, respectively. The derivative of the average reward $\eta(\boldsymbol{\theta})$ with respect to

[10]Though a baseline might be a constant from a traditional perspective, we call the function defined in Eq.19 a baseline function for Eq.3, because it does not add any bias to $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$.

the policy parameter $\boldsymbol{\theta}$ (Eq.3), $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$, is then transformed to

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a) \\ &\qquad \left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\} r(s,a,s_{+1}) \\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a) \\ &\qquad \left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\} \{r(s,a,s_{+1}) - \rho(s,s_{+1}).\} \quad (20)\end{aligned}$$

Proof: see Appendix.

Proposition 4 implies that any $\rho(s,s_{+1})$ defined in Eq.19 can be used as the baseline function of immediate reward $r_{+1}\equiv r(s,a,s_{+1})$ for computing the PG, as in Eq.20. Therefore, the PG can be estimated with the baseline function $\rho(s_t,s_{t+1})$ with large time-steps T,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}R(\boldsymbol{\theta}) &\simeq \frac{1}{T}\sum_{t=0}^{T-1}\left(\nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_t)\right)\{r(s_t,a_t,s_{t+1}) - \rho(s_t,s_{t+1})\} \\ &\equiv \widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}). \quad (21)\end{aligned}$$

In view of the form of the baseline function in Eq.19, we consider the following linear fucntion as a representation of the baseline function,

$$\rho(s,s_{+1};\boldsymbol{\upsilon}) = \begin{pmatrix}\boldsymbol{\upsilon}_u \\ \upsilon_d\end{pmatrix}^{\top}\begin{pmatrix}\boldsymbol{\phi}(s)-\boldsymbol{\phi}(s_{+1}) \\ 1\end{pmatrix} \equiv \boldsymbol{\upsilon}^{\top}\boldsymbol{\psi}(s,s_{+1}),$$

where $\boldsymbol{\upsilon}$ and $\boldsymbol{\phi}(s)$ are its coefficient parameter and feature vector function of state.

When we consider the trace of the covariance matrix of the PG estimates $\widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ as the variance of $\widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ and utilize the results of Greensmith et al.

(2004), an upper bound of the variance is derived as

$$
\begin{aligned}
&\mathrm{Var}_{M(\boldsymbol{\theta})}\left[\widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})\right] \\
&\leq h\big(\,\mathbb{E}_{M(\boldsymbol{\theta})}\left[\|\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\|^2\left\{r(s,a,s_{+1})-\rho(s,s_{+1};\boldsymbol{\upsilon})\right\}^2\right]\,\big) \\
&\equiv h\big(\sigma^2_{\widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})}(\boldsymbol{\upsilon})\big).
\end{aligned}
$$

where $h(a)$ is a monotonically increasing function of its argument a. Accordingly, since the optimal coefficient parameter $\boldsymbol{\upsilon}^*$ for the optimal (linear) baseline fucntion $b^*(s,s_{+1})\equiv\rho(s,s_{+1};\boldsymbol{\upsilon}^*)$ satisfies

$$
\left.\frac{\partial\,\sigma^2_{\widehat{\nabla}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})}(\boldsymbol{\upsilon})}{\partial\,\boldsymbol{\upsilon}}\right|_{\boldsymbol{\upsilon}=\boldsymbol{\upsilon}^*}=0,
$$

the optimal coefficient parameter is computed as[11]

$$
\begin{aligned}
\boldsymbol{\upsilon}^*=&\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\|\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\|^2\boldsymbol{\psi}(s,s_{+1})\boldsymbol{\psi}(s,s_{+1})^\top\right\}^{-1} \\
&\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\|\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\|^2\boldsymbol{\psi}(s,s_{+1})\boldsymbol{r}(s,a,s_{+1})\right\}. \qquad (22)
\end{aligned}
$$

There is also an alternative considerable function for the baseline, termed the 'decent' baseline function $b(s,s_{+1})\equiv\rho(s,s_{+1};\boldsymbol{\upsilon}^\star)$, which satisfies the following condition if the rank of $\mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\psi}(s)\boldsymbol{\psi}(s)^\top\}$ is equal to the number of states,

$$
\mathbb{E}_{M(\boldsymbol{\theta})}\left\{b(s,s_{+1}))\mid s\right\}=\mathbb{E}_{M(\boldsymbol{\theta})}\left\{r(s,a,s_{+1}))\mid s\right\},\quad {}^\forall s, \qquad (23)
$$

and has a statistical meaning. It comes from the fact that, under the condition of Eq.23, this decent baseline function becomes a solution of Poisson's equation:

$$
\upsilon_d^\star+\boldsymbol{\upsilon}_u^{\star\top}\boldsymbol{\phi}(s)=\mathbb{E}_{M(\boldsymbol{\theta})}\left\{r(s,a,s_{+1})-\boldsymbol{\upsilon}_u^\star\boldsymbol{\phi}(s_{+1})\mid s\right\},
$$

[11] The optimal baseline function is instantly computed with $\boldsymbol{\upsilon}^*$ as $b^*(s,s_{+1})=\boldsymbol{\upsilon}^{*\top}\boldsymbol{\psi}(s,s_{+1})$. Note that there is a similarity to the optimal baseline in Peters and Schaal (2006).

thus, $v_d^\star$ and $\boldsymbol{v}_u^{\star\top}\boldsymbol{\phi}(s)$ are equal to the average reward and the (discounted) value function, respectively (Konda and Tsitsiklis, 2003). The parameter $\boldsymbol{v}^\star$ can be computed as[12]

$$\boldsymbol{v}^\star = \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\widetilde{\boldsymbol{\phi}}(s)\boldsymbol{\psi}(s,s_{+1})^\top\right\}^{-1}\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\widetilde{\boldsymbol{\phi}}(s)\boldsymbol{r}(s,a,s_{+1})\right\}, \tag{24}$$

where $\widetilde{\boldsymbol{\psi}}(s) \equiv (\boldsymbol{\phi}(s)^\top, 1)^\top$(Ueno et al., 2008).

By Eq.22 and 24, both the coefficient parameters $\boldsymbol{v}^*$ and $\boldsymbol{v}^\star$ for the optimal $b^*(s,s_{+1})$ and decent $b(s,s_{+1})$ baseline functions can be estimated by least squares and LSTD(λ), respectively, though the estimation for b^* requires LSD estimates. The $\mathcal{LS}$LSD(λ)-PG algorithms with both baseline functions are shown in Algorithm 3 and 4.

Algorithm 3
$\mathcal{LS}$LSD(λ)-PG: Optimization for the policy
with 'optimal' baseline function $b^*(s,s_{+1})$

Given:
- a policy $\pi(s,a;\boldsymbol{\theta})$ with an adjustable $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(s)$.

Define: $\boldsymbol{\psi}(s_t,s_{t+1}) \equiv [\boldsymbol{\phi}(s_t)^\top - \boldsymbol{\phi}(s_{t+1})^\top, 1]^\top$
Initialize: $\boldsymbol{\theta}$, $\lambda \in [0,1)$, $\beta \in [0,1]$, α_t, $\beta_b \in [0,1]$.
Set: $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{z} := \boldsymbol{\phi}(s_0)/\beta$; $\boldsymbol{g} := \boldsymbol{0}$; $\boldsymbol{A} := \boldsymbol{0}$; $\boldsymbol{B} := \boldsymbol{0}$; $\boldsymbol{X} := \boldsymbol{0}$; $\boldsymbol{y} := \boldsymbol{0}$;
for $t = 0$ **to** $T-1$ **do**
  if $t \geq 1$ **then**
    $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t\{\nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta}) + \boldsymbol{\Omega}^\top\boldsymbol{\phi}(s_t)\}\{r_{t+1} - \boldsymbol{\psi}(s_t,s_{t+1})^\top\boldsymbol{X}^{-1}\boldsymbol{y}\}$;
  end if
  $\boldsymbol{c} := \beta\boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
  $\boldsymbol{z} := \beta\lambda\boldsymbol{z} + (1-\lambda)\boldsymbol{\phi}(s_t)$;
  $\boldsymbol{g} := \beta\lambda\boldsymbol{g} + \nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta})$;
  $\boldsymbol{A} := \beta\boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
  $\boldsymbol{B} := \beta\boldsymbol{B} + \boldsymbol{\phi}(s_{t+1})\boldsymbol{g}^\top$;
  $\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top/\|\boldsymbol{c}\|)^{-1}\boldsymbol{B}$;
  $w := \|\nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta}) + \boldsymbol{\Omega}^\top\boldsymbol{\phi}(s_t)\|^2$;
  $\boldsymbol{X} := \beta_b\boldsymbol{X} + w\boldsymbol{\psi}(s_t,s_{t+1})\boldsymbol{\psi}(s_t,s_{t+1})^\top$;
  $\boldsymbol{y} := \beta_b\boldsymbol{y} + w\boldsymbol{\psi}(s_t,s_{t+1})r_{t+1}$;
end for
Return: $p(a \mid s;\boldsymbol{\theta}) = \pi(s,a;\boldsymbol{\theta})$.

[12] $\boldsymbol{v}^\star$ is given by solving the estimating function $\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\widetilde{\boldsymbol{\psi}}(s)^\top(r(s,a,s_{+1}) - \rho(s,s_{+1};\boldsymbol{v}))\right\} = 0$.

Algorithm 4
$\mathcal{LS}$LSD(λ)-PG: Optimization for the policy
with 'decent' baseline function $b^\star(s, s_{+1})$

Given:
- a policy $\pi(s, a; \boldsymbol{\theta})$ with an adjustable $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(s)$.

Define: $\boldsymbol{\psi}(s_t, s_{t+1}) \equiv [\boldsymbol{\phi}(s_t)^\top - \boldsymbol{\phi}(s_{t+1})^\top, 1]^\top$, $\widetilde{\boldsymbol{\phi}}(s_t) \equiv [\boldsymbol{\phi}(s_t)^\top, 1]^\top$.
Initialize: $\boldsymbol{\theta}$, $\lambda \in [0, 1)$, $\beta \in [0, 1]$, α_t, $\lambda_b \in [0, 1)$, $\beta_b \in [0, 1]$
Set: $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{z} := \boldsymbol{\phi}(s_0)/\beta$; $\boldsymbol{g} := \boldsymbol{0}$; $\boldsymbol{A} := \boldsymbol{0}$; $\boldsymbol{B} := \boldsymbol{0}$;
$\boldsymbol{X} := \boldsymbol{0}$; $\boldsymbol{y} := \boldsymbol{0}$; $\boldsymbol{z}_b := \boldsymbol{0}$
for $t = 0$ **to** $T - 1$ **do**
if $t \geq 1$ **then**
$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t \{\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta}) + \boldsymbol{\Omega}^\top \boldsymbol{\phi}(s_t)\} \{r_{t+1} - \boldsymbol{\psi}(s_t, s_{t+1})^\top \boldsymbol{X}^{-1} \boldsymbol{y}\}$;
end if
$\boldsymbol{c} := \beta \boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
$\boldsymbol{z} := \beta\lambda \boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(s_t)$;
$\boldsymbol{g} := \beta\lambda \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t; \boldsymbol{\theta})$;
$\boldsymbol{A} := \beta \boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
$\boldsymbol{B} := \beta \boldsymbol{B} + \boldsymbol{\phi}(s_{t+1}) \boldsymbol{g}^\top$;
$\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top / \|\boldsymbol{c}\|)^{-1} \boldsymbol{B}$;
$\boldsymbol{z}_b := \beta_b \lambda_b \boldsymbol{z}_b + \widetilde{\boldsymbol{\phi}}(s_t)$;
$\boldsymbol{X} := \beta_b \boldsymbol{X} + \boldsymbol{z}_b \boldsymbol{\psi}(s_t, s_{t+1})^\top$;
$\boldsymbol{y} := \beta_b \boldsymbol{y} + \boldsymbol{z}_b r_{t+1}$;
end for
Return: $p(a \mid s; \boldsymbol{\theta}) = \pi(s, a; \boldsymbol{\theta})$.

5 Numerical Experiments

We verify the performance of our proposed algorithms in stochastic "torus" MDPs in Section 5.1. The proposed algorithms and other existing PG algorithms are also applied to a pendulum balancing problem as a continuous state-action problem in Section 5.2.

5.1 Torus $|\mathcal{S}|$-state MDP

We tested the performance of our proposed algorithms in a stochastic "one-dimensional torus grid-world" with a finite set of grids $\mathcal{S} = \{1, .., |\mathcal{S}|\}$ and a set of two possible actions $\mathcal{A} = \{L, R\}$. This is a typical $|\mathcal{S}|$-state MDP task

where the state transition probabilities p are given by

$$\begin{cases} p(s-1 \mid s, L) & = q_s \\ p(s \mid s, L) & = \frac{1-q_s}{2} \\ p(s+1 \mid s, L) & = \frac{1-q_s}{2} \end{cases} \qquad \begin{cases} p(s-1 \mid s, R) & = \frac{1-q_s}{2} \\ p(s \mid s, R) & = \frac{1-q_s}{2} \\ p(s+1 \mid s, R) & = q_s, \end{cases}$$

otherwise $p = 0$, where $s = 0$ and $s = |\mathcal{S}|$ ($s = 1$ and $s = |\mathcal{S}| + 1$) are the identical states and $q_s \in [0, 1]$ is a task-dependent constant. In this experiment, a stochastic policy is a so-called Boltzmann or Gibbs policy represented by a sigmoidal function

$$\pi(s, a = L; \boldsymbol{\theta}) = 1 - \pi(s, a = R; \boldsymbol{\theta}) = \frac{1}{1 + \exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(s))}.$$

Here, all of the elements of the state-feature vectors $\boldsymbol{\phi}(1), \ldots, \boldsymbol{\phi}(|\mathcal{S}|) \in \mathcal{R}^{|\mathcal{S}|}$ were independently drawn from the Gaussian distribution $\mathtt{N}(\mu = 0,\, \sigma^2 = 1)$ for each episode (simulation run). This was to assess how the parameterization of the stochastic policy affected the performance of our algorithms. The state-feature vectors $\boldsymbol{\phi}(s)$ were also used as the basis function for the LSD estimate $\boldsymbol{f}(s; \boldsymbol{\Omega})$, cf. Eq.13.

[Figure 1 about here.]

5.1.1 Performance of $\mathcal{LS}$LSD(λ) algorithm

First, we verified how precisely the $\mathcal{LS}$LSD(λ) algorithm estimated $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ without regard to the setting of q_s and the policy parameter $\boldsymbol{\theta}$. Each element of $\boldsymbol{\theta}$ and the task-dependent constant q_s were randomly initialized according to $\mathtt{N}(\mu = 0,\, \sigma^2 = 0.5^2)$ and $\mathtt{U}(a = 0.7,\, b = 1)$, respectively, where $\mathtt{U}(a = 0.7,\, b = 1)$ is the uniform distribution over the interval of $[a, b]$. These values were fixed during each episode.

Figure 1(A) shows a typical time course of the LSD estimates $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$ for

$|\mathcal{S}|=3$-state MDP, where nine different colors indicate all of the different elements of the LSD, respectively. The solid lines denote the values estimated by $\mathcal{LS}$LSD(0), and the dotted lines denote the analytical solution of the LSD. This result shows that the proposed algorithm, $\mathcal{LS}$LSD(λ), can estimate the LSD $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. Besides this result, we have confirmed that the estimates by $\mathcal{LS}$LSD(0) always converged to the analytical solution for $|\mathcal{S}| = 3$ as in Figure 1 (A).

Second, we investigated the effect of the eligibility decay rate λ using 7-state MDPs. In order to evaluate the average performance over various settings, we employed a "relative error" criterion that is defined by $\mathbb{E}_{M(\boldsymbol{\theta})}\{\|\boldsymbol{f}(x;\boldsymbol{\Omega}) - \boldsymbol{f}(x;\boldsymbol{\Omega}^\star)\|^2\}/\mathbb{E}_{M(\boldsymbol{\theta})}\{\|\boldsymbol{f}(x;\boldsymbol{\Omega}^\star\|^2\}$, where $\boldsymbol{\Omega}^\star$ is the optimal parameter defined in Proposition 3. Figure 1 (B) and (C) show the time courses of the relative error averages over 200 episodes for $\lambda = 0$, 0.3, 0.9, and 1. The only difference between these two figures was the number of elements of the feature-vectors $\boldsymbol{\phi}(s)$. The feature-vectors $\boldsymbol{\phi}(s) \in \mathcal{R}^7$ used in (B) were appropriate and sufficient to distinguish all of the different states, while the feature-vectors $\boldsymbol{\phi}(s) \in \mathcal{R}^6$ used in (C) were inappropriate and insufficient. These results were consistent with the theoretical prospects. In particular, we could set λ arbitrarily in $[0, 1)$ if the basis function was appropriate (Figure 1 (B)), otherwise we would need to set λ close but not equal to 1 (Figure 1 (C)).

5.1.2 Comparison to other PG methods

We compared the $\mathcal{LS}$LSD(λ=0)-PG algorithm with the other PG algorithms for 3-state MDPs, concerned with the estimation of PG $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ and the optimization of the policy parameter $\boldsymbol{\theta}$. The policy and the state transition probability were set as $\theta_i \sim \mathtt{N}(0, 0.5^2)$ and $q_i \sim \mathtt{U}(0.95, 1)$ for every $i \in \{1, 2, 3\}$, respectively.

[Figure 2 about here.]

Figure 2 shows the reward setting in the MDP. There are two types of rewards: "$r = (\pm)2/Z(c)$" and "$r = (\pm)c/Z(c)$", where the variable c was initialized using

the uniform distribution over $[0.95, 1)$ for each episode (simulation run) and the function $Z(c)$ was the normalizing constant to assure $\mathtt{max}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = 1$[13]. Note that the reward c defines the infimum value of γ to find the optimal policy: $\gamma^2 + \gamma > \frac{2c}{2-c}$. Therefore, the setting of γ is important but difficult in this task. From the performance baselines of the existing PG methods, we adopted two algorithms: GPOMDP (Baxter and Bartlett, 2001) and Konda's Actor-Critic (Konda and Tsitsiklis, 2003). These algorithm used the baseline function being state value estimates which were estimated by LSTD(λ) (Bradtke and Barto, 1996; Boyan, 2002; Yu and Bertsekas, 2006), while the original versions did not use the baseline functions.

Figure 3 shows the results for the estimation of PG $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ from Eq.17. The forgetting rate for the statistics and the eligibility decay rate were set as $\beta = 1$ and $\lambda = 0$ for all of the algorithms (A) and (B) represent the mean and the standard deviation of the angles between the estimates and the exact PG, respectively. These results show that $\mathcal{LS}$LSD-PG with estimating the optimal baseline function $b^*(s, s_{+1})$, termed $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$, worked best to estimate the PG. $\mathcal{LS}$LSD-PG with whether $b(s, s_{+1})$ or $b^*(s, s_{+1})$ drastically improved the PG estimation performance of $\mathcal{LS}$LSD-PG without a baseline function, termed $\mathcal{LS}$LSD-PG:None, which was poorer than even GPOMPD:$V(s)$, though $\mathcal{LS}$LSD-PG:None worked better than GPOMDP:None. Thus, we confirmed that these baseline functions for $\mathcal{LS}$LSD-PG could be essential.

[Figure 3 about here.]

Finally, we examined the optimization of the policy parameter $\boldsymbol{\theta}$, i.e. the average reward, with these PG methods. In this experiment, the forgetting rate and the eligibility decay rate were set as $\beta = 0.99$ and $\lambda = 0$. In order to avoid the effect from poor estimations of the functions for the PG estimate, there

[13]The normalizing constant $Z(c)$ was computed with the un-normalized reward function as $Z(c) = \mathtt{max}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ analytically.

was a pre-learning period of 50 time-steps, where the learning rate α was set to zero. It means that the policy remained unchanged in the first 50 time-steps. Figure 4 shows the means and the standard deviations of the average rewards at an earlier stage (500 time-step) and a later stage (10^4 time-step) over 1,000 independent simulations of various learning rates α, in order to give comparisons among the PG algorithms for the optimization of the policy parameter. It was confirmed that $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$ worked best, except for the high learning rate where the learning speed of $b^*(s, s_{+1})$ could not properly follow the changes of the policy rather than that of $b(s, s_{+1})$. Figure 5 shows the time courses of the average reward, where we chose appropriate learning rates for the PG algorithms by drawing upon the previous results; $\alpha = 0.16$ in $\mathcal{LS}$LSD-PG:$b(s, s_{+1})$, $\alpha = 0.08$ in $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$, $\alpha = 0.08$ in Actor-Critic:$V(s)$, and $\alpha = 0.007$ in GPOMDP:$V(s)$. This result also indicates that our $\mathcal{LS}$LSD-PG algorithm with the optimal baseline function $b^*(s, s_{+1})$ outperformed the other PG algorithms in the sense of realizing both the highest average and the lowest standard deviation of the average rewards.

[Figure 4 about here.]

[Figure 5 about here.]

5.2 Continuous state-action problem

The $\mathcal{LS}$LSD(λ)-PG and the other existing PG algorithms were also applied to a continuous state-action problem. This task is to balance a pendulum near the top.

5.2.1 Interpretation of continuous state problem

Although this task obviously violates Assumption 1, on which the most policy gradient algorithms including $\mathcal{LS}$LSD(λ)-PG have also been based, it must be

valuable to acquire insights of feasibility about PG algorithms. The reason comes from the following interpretations: a continuous problem to (i) a numerous-state MDP problem with some structures, or (ii) a partially observable MDP (POMDP) problem with a belief states (Aberdeen, 2003). While the interpretation of (i) is apparent, (ii) comes from the fact that, when the policy (or the baseline function) in a continuous state problem is represented by a linear function with finite basis functions that output bounded activation values, the activations biased to non-negative values and normalized can be regarded as the belief states of finite-state POMDP (Aberdeen, 2003).

5.2.2 Pendulum balancing problem

A pendulum balancing problem is a well known benchmark in continuous RL problems (Peters et al., 2005; Morimura et al., 2005). The state $s \equiv \{x, \dot{x}\}$ comprised the angle and the angular speed of the pendulum, which were limited in ranges $[-\pi/6, \pi/6]$ and $[-\pi/2, \pi/2]$, respectively, as shown in Figure 6. Its dynamics was given by

$$\ddot{x}_{t+1} = \frac{-\mu \dot{x}_t + mgl \sin(x_t) + a}{ml^2},$$

where a was the torque as an action selected by a learning agent. The physical parameters were set $m = l = 1$, $g = 9.8$, and $\mu = 0.01$. The reward function was set as

$$r(s_t, a_t, s_{t+1}) \equiv \begin{cases} -x_{t+1}^2 - 0.5\dot{x}_{t+1}^2 - 0.001a_t^2 - 1, & \text{if } |x_{t+1}| > \pi/6 \text{ or } |\dot{x}_{t+1}| > \pi/2, \\ -x_{t+1}^2 - 0.5\dot{x}_{t+1}^2 - 0.001a_t^2, & \text{otherwise.} \end{cases}$$

The state s_t was initialized by the uniform distributions as $x_t \sim \mathtt{U}(-\pi/8, \pi/8)$ and $\dot{x}_t \sim \mathtt{U}(-1, 1)$, at the beginning of each episode or when the previous state s_{t-1} deviated from the defined ranges, i.e., $|x_{t-1}| > \pi/6$ or $|\dot{x}_{t-1}| > \pi/2$. The state was

also initialized with the probability 0.01 at each time step, in order to explore the state space more efficiently.

[Figure 6 about here.]

Since this problem has potentially an infinite number of states, we used a normalized radial basis function (nRBF) model (Doya, 2000) for the policy and (the part of) the baseline function:

$$\pi(s, a; \boldsymbol{\theta}) \equiv \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\left\{a - \boldsymbol{\theta}^\top \boldsymbol{\phi}(s)\right\}^2}{2}\right],$$
$$g(s; \boldsymbol{\upsilon}) \equiv \boldsymbol{\theta}^\top \boldsymbol{\phi}(s),$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\upsilon}$ were the parameters of the policy and the baseline function. $\boldsymbol{\phi}(s)$ was the output of the nRBF model having 3×3 RBFs as follows. The centers of the RBFs were set as $x \in \{-\pi/9, 0, \pi/9\}$ and $\dot{x} \in \{-\pi/3, 0, \pi/3\}$. The standard deviations of the all RBFs were set as $\pi/9$ and $\pi/3$ for x and $\dot{x}$, respectively. The policy parameter $\boldsymbol{\theta} \in \mathcal{R}^9$ was set to $\mathbf{0}$ at the beginning of each episode.

We applied our $\mathcal{LS}$LSD(λ)-PGs with optimal baseline function $b^*(s, s_{+1})$ and decent baseline function $b(s, s_{+1})$, (Konda's) Actor-Critic:$V(s)$, and GPOMDP:$V(s)$, as those in the previous experiment (Section 5.1.2). We set meta-parameters of those algorithms appropriately: $\alpha = 0.2$ and $\beta = 0.9995$ for the $\mathcal{LS}$LSD-PGs and Actor-Critic, and $\alpha = 0.04$, $\gamma = 0.99$, and $\beta = 0.9995$ for the GPOMDP were set. And λ of $\mathcal{LS}$LSD-PGs with $b^*(s, s_{+1})$ and $b(s, s_{+1})$, Actor-Critic, and the GPOMDP were set to 0.7, 0.5, 0.5, and 0. There was a pre-learning period of 10^3 time steps to avoid the effect from poor PG estimates, where α was set to zero.

Figure 7 shows the time courses of (A) the mean and (B) the standard deviations of the average rewards. We confirmed that the performance of $\mathcal{LS}$LSD-PG:$b(s, s_{+1})$ was better than that of $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$. Because its order in the previous MDP problem (Figure 5) was opposite, it must be coursed by the difficulty of the learning of the parameter $\boldsymbol{\upsilon}^*$ of the 'optimal' baseline function

$b^*(s, s_{+1})$ in Eq.22 than $\boldsymbol{\upsilon}^\star$ of the 'decent' one $b(s, s_{+1})$ in Eq.24 with the nRBF model. This distinction of the difficulties indicates that $b^*(s, s_{+1})$ could be more complex than $b(s, s_{+1})$ and $b^*(s, s_{+1})$ would need more representation capability of the baseline function approximator. By the comparison of these forms in Eq.22 and 24, these things would also come from the existence of the weights $\|\nabla_{\boldsymbol{\theta}} \ln \pi(s, a; \boldsymbol{\theta}) + \widehat{\nabla}_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)\|^2$, i.e., the weights would often make the estimation of $b^*(s, s_{+1})$ with a function approximation difficult. We also confirmed from Figure 7 that the performance of the $\mathcal{LS}$LSD-PG:$b(s, s_{+1})$ was slightly or much better than Actor-Critic:$V(s)$ or GPOMDP:$V(s)$ algorithm.

[Figure 7 about here.]

6 Related works

There are two alternative methods that estimate the derivative of the (stationary) state distribution and that have already been proposed in Glynn (1991) or Rubinstein (1991) and Ng et al. (2000). However, these are different from our approach and have the following problems. The method in Glynn (1991) or Rubinstein (1991) is in operations research called "the likelihood ratio gradient" or "the score function". This method can be problematic as to how to design the recurrence state, since the applicability is limited to regenerative processes (see Baxter and Bartlett (2001) in detail). The other method proposed in Ng et al. (2000) is not a direct estimation of the derivative of the state distribution but is done via the estimation of the state distribution with density propagation. Accordingly, both these methods require knowledge of which state the agent is in, while our method only needs to observe the feature vector of the state.

Meanwhile, there is an average reward PG algorithm (Tsitsiklis and Van Roy, 1999; Konda and Tsitsiklis, 2003; Sutton et al., 2000) that eliminates the use of the forgetting rate by introducing a differential cost function as a solution of

Poisson's equation (also known as the average reward Bellman equation in RL). However, since to date this is a unique PG framework proposed for maximizing the average reward[14], more studies of the average reward optimization would be needed and significant. At least one possible advantage of the proposed framework over the existing average reward PG method is that a closed-form solution of an optimal baseline function for minimizing the variance bound of PG estimate can be computed by least squares approaches, while such a solution in conventional PG frameworks has not been explored and would be intractable (Greensmith et al., 2004).

7 Concluding remarks

Our propositions show that the actual forward and virtual backward Markov chains are closely related and have common properties. Utilizing these properties, we proposed $\mathcal{LS}$LSD(λ) as an estimation algorithm for the log stationary distribution derivative (LSD), and $\mathcal{LS}$LSD(λ)-PG as a PG algorithm utilizing the LSD estimate. The experimental results also demonstrated that $\mathcal{LS}$LSD(λ) worked for $\lambda \in [0, 1)$ and $\mathcal{LS}$LSD(λ)-PG could learn regardless of the task's requirements for the value of γ to optimize the average reward. At the same time, it has been suggested that there is theoretically no significant difference in performances between the average-reward-based PG methods and the alternative PG methods with forgetting rate γ for the value functions close to one (Tsitsiklis and Van Roy, 2002). This might be true for the case of our proposed PG, $\mathcal{LS}$LSD-PG, which would mean that $\mathcal{LS}$LSD-PG might not drastically improve the performances of the PG methods with γ as does the algorithm proposed by Kimura and Kobayashi (1998) when γ was set appropriately. However, it is noted that $\mathcal{LS}$LSD-PG is free of the setting of γ. In contrast, the learning performances of Konda's actor-critic and the

[14] Although R-learning also maximizes the average reward, it is based on the value function not the PG algorithm (Sutton and Barto, 1998).

$\mathcal{LS}$LSD-PG approaches do not seem to be the significantly different, as confirmed in our numerical experiments. This would seem to come from the fact that the $\mathcal{LS}$LSD-PG is just a dual approach compared to Konda's actor-critic approach. In dyanamic programming (DP), Wang et al. (2007) formalize the dual approaches, in which stationary distributions are maintained instead of the value functions (primal approaches). On the other hand, in PGRL, $\mathcal{LS}$LSD-PG approache maintains the LSD instead of the value functions, to estimate the policy gradient. As considering the relationship between DP and PGRL, Konda's actor-critic and our $\mathcal{LS}$LSD-PG approaches are corresponds to primal and dual approaches in PGRL, respectively. Since Wang et al. (2008) also prove the advantage of the dual DP approaches that the dual updates cannot diverge even with function approximations, the $\mathcal{LS}$LSD-PG approaches might have such the advantages and thus more theoritical and experimental work is necessary to further understand its effectiveness.

Meanwhile, in our numerical experiments, $\mathcal{LS}$LSD-PG approaches improved the performances of GPOMDP drastically. It could come from the fact that $\mathcal{LS}$LSD-PG and GPOMD are fundamentally different approaches, because, GPOMDP can be regarded as the method based on a fully model-free estimation of the PG, while $\mathcal{LS}$LSD-PG can be regarded as a method based on somewhat model-based estimation of it. Although the latent difficulty of the PG estimation problem does not change, $\mathcal{LS}$LSD-PG utilizes a model information or a prior knowledge of the model. Furthermore, since the policy is usually a parametric model and the LSD approximator would be defined as a similar model to the policy model, the utilization the policy model in $\mathcal{LS}$LSD-PG must be useful. Accordingly, it is an important future work to discuss and define a necessary and suffered basis function of the (linear) LSD approximator based on the parameterization of the policy.

On the ohter hand, the use of LSD estimation will open up new possibilities for

the natural gradient learning (Kakade, 2001; Peters et al., 2005; Peters and Schaal, 2008; Morimura et al., 2008b). It enables us to compute a valid Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta})$ for an NPG, which is effective especially in the large-scale MDPs,

$$\boldsymbol{G}(\boldsymbol{\theta}) := \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\nabla_{\boldsymbol{\theta}} \ln \pi(s,a;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln \pi(s,a;\boldsymbol{\theta})^\top + \iota \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)^\top \Big\},$$

where $\iota \in [0,1]$ interpolates the natural policy gradient ($\lambda = 0$) and the natural state-action gradient ($\lambda = 1$) (Morimura et al., 2008b).

In addition, the use of LSD estimation might offer novel methods for addressing the trade-offs between exploration and exploitation. This is because LSD gives statistical information about how much a change of the state stationary distribution is caused by the perturbation of each element of each policy parameter, while the stationary distribution having a low entropy would make the exploration hard.

References

Aberdeen, D. (2003) "Policy-Gradient Algorithms for Partially Observable Markov Decision Processes," Ph.D. dissertation, Australian National University.

Baird, L. and Moore, A. (1999) "Gradient Descent for General Reinforcement Learning," in *Advances in Neural Information Processing Systems*, Vol. 11: MIT Press.

Baxter, J. and Bartlett, P. (2001) "Infinite-Horizon Policy-Gradient Estimation," *Journal of Artificial Intelligence Research*, Vol. 15, pp. 319–350.

Baxter, J., Bartlett, P., and Weaver, L. (2001) "Experiments with Infinite-Horizon Policy-Gradient Estimation," *Journal of Artificial Intelligence Research*, Vol. 15, pp. 351–381.

Bertsekas, D. P. (1995) *Dynamic Programming and Optimal Control, Volumes 1 and 2*: Athena Scientific.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996) *Neuro-Dynamic Programming*: Athena Scientific.

Boyan, J. A. (2002) "Technical Update: Least-Squares Temporal Difference Learning," *Machine Learning*, Vol. 49, No. 2-3, pp. 233–246.

Bradtke, S. J. and Barto, A. G. (1996) "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, Vol. 22, No. 1-3, pp. 33–57.

Doya, K. (2000) "Reinforcement learning in continuous time and space," *Neural Computation*, Vol. 12, pp. 219–245.

Glynn, P. W. (1991) "Likelihood ratio gradient estimation for stochastic systems," *Communications of the ACM*, Vol. 33, No. 10, pp. 75–84.

Greensmith, E., Bartlett, P., and Baxter, J. (2004) "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning," *Journal of Machine Learning Research*, Vol. 5, pp. 1471–1530.

Gullapalli, V. (2000) "A stochastic reinforcement learning algorithm for learning real-valued functions," *Neural Networks*, Vol. 3, No. 6, pp. 671–692.

Kakade, S. (2001) "Optimizing Average Reward Using Discounted Rewards," in *Annual Conference on Computational Learning Theory*, Vol. 14: MIT Press.

Kimura, H. and Kobayashi, S. (1998) "An Analysis of Actor/Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Function," in *International Conference on Machine Learning*, pp. 278–286.

Konda, V. S. and Tsitsiklis, J. N. (2003) "On Actor-Critic Algorithms," *SIAM Journal on Control and Optimization*, Vol. 42, No. 4, pp. 1143–1166.

Lagoudakis, M. G. and Parr, R. (2003) "Least-Squares Policy Iteration," *Journal of Machine Learning Research*, Vol. 4, pp. 1107–1149.

MacKay, D. (2003) *Information theory, inference, and learning algorithms*: Cambridge.

Morimura, T., Uchibe, E., and Doya, K. (2005) "Utilizing Natural Gradient in Temporal Difference Reinforcement Learning with Eligibility Traces," in *International Symposium on Information Geometry and its Applications*, pp. 256–263.

Morimura, T., Uchibe, E., Yoshimoto, J., and Doya, K. (2007) "Reinforcement Learning with Log Stationary Distribution Gradient,"Technical report, Nara Institute of Science and Technology.

Morimura, T., Uchibe, E., and Doya, K. (2008a) "Natural Actor-Critic with Baseline Adjustment for Variance Reduction," in *International Symposium on Artificial Life and Robotics*.

Morimura, T., Uchibe, E., Yoshimoto, J., and Doya, K. (2008b) "A new natural gradient of average reward for policy search," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Morimura, T., Uchibe, E., Yoshimoto, J., and Doya, K. (2009) "A Generalized Natural Actor-Critic Algorithm." (submitted).

Ng, A. Y., Parr, R., and Koller, D. (2000) "Policy Search via Density Estimation," in *Advances in Neural Information Processing Systems*: MIT Press.

Peng, J. and Williams, R. J. (1996) "Incremental Multi-Step Q-Learning," *Machine Learning*, Vol. 22, No. 1-3, pp. 283–290.

Peters, J. and Schaal, S. (2006) "Policy Gradient Methods for Robotics," in *IEEE International Conference on Intelligent Robots and Systems*.

Peters, J. and Schaal, S. (2008) "Natural Actor-Critic," *Neurocomputing*, Vol. 71, No. 7-9, pp. 1180–1190.

Peters, J., Vijayakumar, S., and Schaal, S. (2005) "Natural Actor-Critic," in *European Conference on Machine Learning*.

Rubinstein, R. Y. (1991) "How to optimize discrete-event system from a single sample path by the score function method," *Annals of Operations Research*, Vol. 27, No. 1, pp. 175–212.

Schinazi, R. B. (1999) *Classical and Spatial Stochastic Processes*: Birkhauser.

Singh, S. P., Jaakkola, T., and Jordan, M. I. (1994) "Learning Without State-Estimation in Partially Observable Markovian Decision Processes," in *International Conference on Machine Learning*, pp. 284–292.

Sutton, R. S. (1988) "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, Vol. 3, pp. 9–44.

Sutton, R. S. and Barto, A. G. (1998) *Reinforcement Learning*: MIT Press.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (2000) "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in Neural Information Processing Systems*, Vol. 12: MIT Press.

Tsitsiklis, J. N. and Van Roy, B. (1999) "Average Cost Temporal-Difference Learning," *Automatica*, Vol. 35, No. 11, pp. 1799–1808.

Tsitsiklis, J. N. and Van Roy, B. (2002) "On Average Versus Discounted Reward Temporal-Difference Learning," *Machine Learning*, Vol. 49, No. 2, pp. 179–191.

Ueno, T., Kawanabe, M., Mori, T., Maeda, S., and Ishii, S. (2008) "A semiparametric statistical approach to model-free policy evaluation," in *International Conference on Machine Learning*, pp. 857–864.

Wang, T., Bowling, M., and Schuurmans, D. (2007) "Dual Representations for Dynamic Programming and Reinforcement Learning," in *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 44–51.

Wang, T., Lizotte, D., Bowling, M., and Schuurmans, D. (2008) "Stable dual dynamic programming," in *Advances in Neural Information Processing Systems*: MIT Press.

Williams, R. J. (1992) "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning*, Vol. 8, pp. 229–256.

Young, P. (1984) *Recursive Esimation and Time-series Analysis*: Springer-Verlag.

Yu, H. and Bertsekas, D. P. (2006) "Convergence Results for Some Temporal Difference Methods Based on Least Squares,"Technical report, LIDS report 2697, M.I.T.

Appendix

7.1 Derivation of Eq.4

Using the relation between the forgetting (or discounted) state-value function $V_\gamma^\pi(s)$ and the average reward $\eta(\boldsymbol{\theta})$ (Singh et al., 1994)

$$\eta(\boldsymbol{\theta}) = (1-\gamma)\sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) V_\gamma^\pi(s),$$

the derivative of average reward with respect to $\boldsymbol{\theta}$ is given by

$$\nabla_\theta \eta(\boldsymbol{\theta}) = (1-\gamma)\left(\sum_{s\in\mathcal{S}} \nabla_{\boldsymbol{\theta}} d_{M(\boldsymbol{\theta})}(s) V_\gamma^\pi(s) + \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} V_\gamma^\pi(s)\right), \tag{25}$$

where $\nabla_\alpha AB$ implies $(\nabla_\alpha A)B$. The second term is modified as follows:

$$\begin{aligned}
&\sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} V_\gamma^\pi(s) \\
&\quad= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} \{\pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a)\} \\
&\quad= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \Big[\nabla_{\boldsymbol{\theta}} \pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a) + \pi(s,a;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} Q_\gamma^\pi(s,a)\Big] \\
&\quad= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \Big[\nabla_{\boldsymbol{\theta}} \pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a) \\
&\qquad\qquad + \pi(s,a;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \sum_{s_{+1}\in\mathcal{S}} p(s_{+1} \mid s,a)\big\{r(s,a,s_{+1}) + \gamma V_\gamma^\pi(s_{+1})\big\}\Big] \\
&\quad= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} \pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a) + \gamma \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} V_\gamma^\pi(s) \qquad (26) \\
&\quad= \frac{1}{1-\gamma} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} \pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a). \qquad (27)
\end{aligned}$$

Eq.26 is given by the property of stationary distribution (Eq.1). Substituting Eq.27 in Eq.25, we can derive Eq.4:

$$\begin{aligned}
\nabla \eta(\boldsymbol{\theta}) = &\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_{M(\boldsymbol{\theta})}(s) \pi(s,a;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln \pi(s,a;\boldsymbol{\theta}) Q_\gamma^\pi(s,a) \\
&+ (1-\gamma) \sum_{s\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s) \nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s) V_\gamma^\pi(s). \qquad (4)
\end{aligned}$$

□

$\mathcal{LS}$LSD(λ) as model-based-learning

>> not yet <<

Proof of Proposition 4

If the following equation holds,

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a)\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\}\rho(s,s_{+1})=\mathbf{0} \tag{28}$$

then the transformation to Eq.20 obviously true. Because of Eq.19 and

$$\begin{cases}\sum_{a\in\mathcal{A}}\pi(s,a;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})\,c=\nabla_{\boldsymbol{\theta}}c=\mathbf{0},\\ \sum_{s\in\mathcal{S}}d_{M(\boldsymbol{\theta})}(s)\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\,c\qquad=\nabla_{\boldsymbol{\theta}}c=\mathbf{0},\end{cases}$$

we know that

$$\begin{aligned}&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a)\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\}\rho(s,s_{+1})\\ =&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d_{M(\boldsymbol{\theta})}(s)\pi(s,a;\boldsymbol{\theta})p(s_{+1}\mid s,a)\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\}\left\{g(s)-g(s_{+1})\right\}.\end{aligned} \tag{29}$$

Since a time average is equivalent to a state-action space average in an ergodic Markov chain $M(\boldsymbol{\theta})$ by Assumption 1, Eq.29 is transformed to

$$
\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_t)\right\}\left\{g(s_t)-g(s_{t+1})\right\}\\
&=\lim_{T\to\infty}\frac{1}{T}\Big[\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s_0,a_0;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_0)\right\}g(s_0)\\
&\quad+\left\{\nabla_{\boldsymbol{\theta}}\ln\pi(s_T,a_T;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_T)\right\}g(s_T)\\
&\quad+\sum_{t=1}^{T}\left\{-\nabla_{\boldsymbol{\theta}}\ln\pi(s_{t-1},a_{t-1};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_{t-1})+\nabla_{\boldsymbol{\theta}}\ln\pi(s_t,a_t;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_t)\right\}g(s_t)\Big]\\
&=\sum_{s_{-1}\in\mathcal{S}}\sum_{a_{-1}\in\mathcal{A}}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}d_{M(\boldsymbol{\theta})}(s_{-1})\pi(s_{-1},a_{-1};\boldsymbol{\theta})p(s\mid s_{-1},a_{-1})\pi(s,a;\boldsymbol{\theta})\\
&\quad\left\{-\nabla_{\boldsymbol{\theta}}\ln\pi(s_{-1},a_{-1};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_{-1})+\nabla_{\boldsymbol{\theta}}\ln\pi(s,a;\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\}g(s)\\
&=\sum_{s_{-1}\in\mathcal{S}}\sum_{a_{-1}\in\mathcal{A}}\sum_{s\in\mathcal{S}}d_{M(\boldsymbol{\theta})}(s_{-1})\pi(s_{-1},a_{-1};\boldsymbol{\theta})p(s\mid s_{-1},a_{-1})\\
&\quad\left\{-\nabla_{\boldsymbol{\theta}}\ln\pi(s_{-1},a_{-1};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\right\}g(s)\\
&=\sum_{s\in\mathcal{S}}d_{M(\boldsymbol{\theta})}(s)g(s)\\
&\quad\Big[\mathbb{E}_{B(\boldsymbol{\theta})}\{\nabla_{\boldsymbol{\theta}}\ln\pi(s_{-1},a_{-1};\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s_{-1})\mid s\}-\nabla_{\boldsymbol{\theta}}\ln d_{M(\boldsymbol{\theta})}(s)\Big] \qquad (30)\\
&=\mathbf{0}, \qquad (31)
\end{aligned}
$$

where Eq.8 (Proposition 2) and Eq.9 are used for the transformations to Eq.30 and Eq.31. Therefore, Eq.28 holds. □

List of Figures

1 Performances of $\mathcal{LS}$LSD(λ) for the estimation of the LSD $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. (A) A typical time course of LSD estimates in a 3-state MDP. (B, C) The relative errors averaged over 200 episodes in 7-state MDPs for various λ: (B) with a proper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^7$, (C) with an improper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^6$. 43

2 Reward setting of 3-state MDPs used in our comparative studies. The value of c is selected from the uniform distribution $\mathtt{U}[0.95, 1)$ for each episode. $Z(c)$ is a normalizing function to assure $\mathtt{max}_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) = 1$. 44

3 Comparison with various PG algorithms for the estimation of the PG over 2,500 episodes: (A) and (B) are the mean and the standard deviation of the angles between the estimates and the exact PG, respectively. 45

4 Comparisons with various PG algorithms for the means of the average rewards in the 3-state torus MDPs with 1,000 episodes about various learning rates. (A) and (B) are the mean and the standard deviation of the average rewards at 500 time-step, respectively. (C) and (D) are at 10^4 time-step. 46

5 Comparison with various PG algorithms for the optimization of the policy parameters with the appropriate learning rate in the 3-state torus MDPs over 1,000 episodes. 47

6 Pendulum balancing problem near the top ranges; $x \in [-\pi/6, \pi/6]$ and $\dot{x} \in [-\pi/2, \pi/2]$. 48

7 Comparison with various PG algorithms for the optimization of the policy parameters in the pendulum balancing problem over 500 episodes. 49

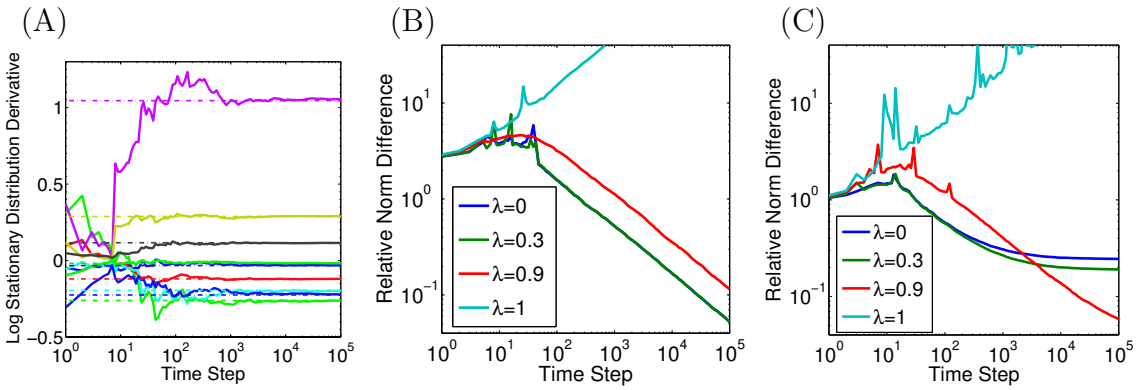

Figure 1: Performances of $\mathcal{LS}$LSD(λ) for the estimation of the LSD $\nabla_{\boldsymbol{\theta}} \ln d_{M(\boldsymbol{\theta})}(s)$. (A) A typical time course of LSD estimates in a 3-state MDP. (B, C) The relative errors averaged over 200 episodes in 7-state MDPs for various λ: (B) with a proper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^7$, (C) with an improper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^6$.

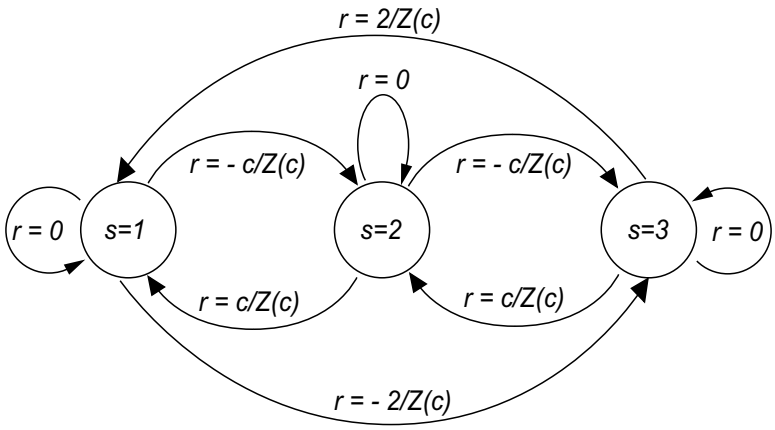

Figure 2: Reward setting of 3-state MDPs used in our comparative studies. The value of c is selected from the uniform distribution $\mathtt{U}[0.95, 1)$ for each episode. $Z(c)$ is a normalizing function to assure $\mathtt{max}_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = 1$.

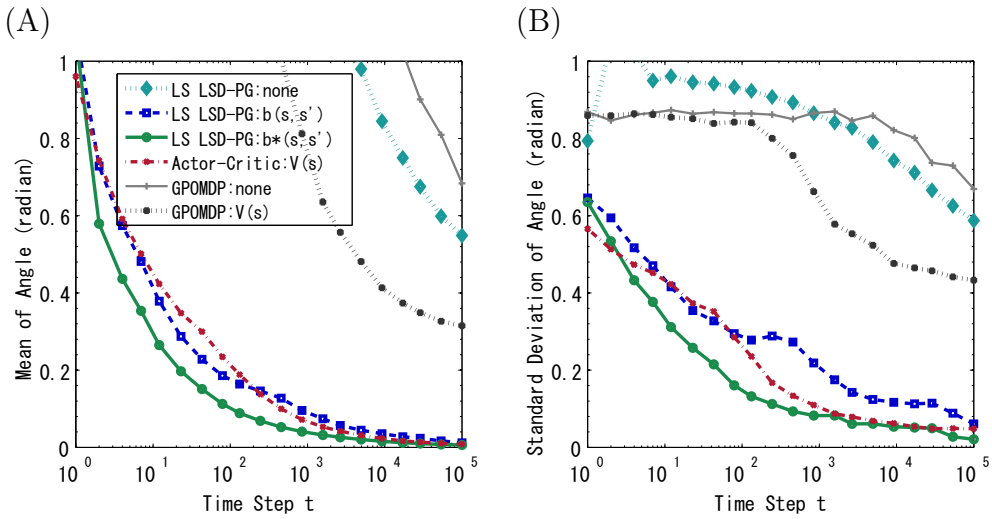

Figure 3: Comparison with various PG algorithms for the estimation of the PG over 2,500 episodes: (A) and (B) are the mean and the standard deviation of the angles between the estimates and the exact PG, respectively.

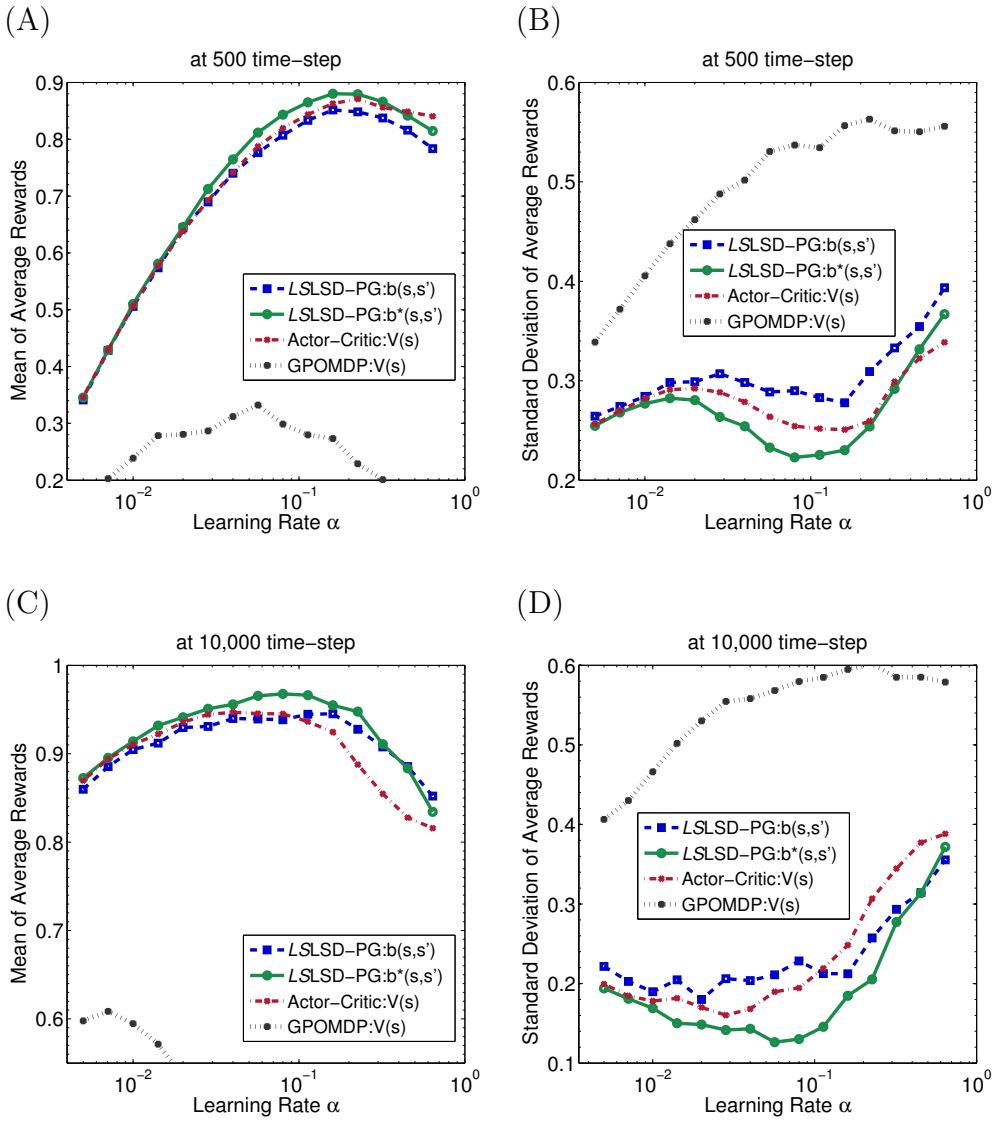

Figure 4: Comparisons with various PG algorithms for the means of the average rewards in the 3-state torus MDPs with 1,000 episodes about various learning rates. (A) and (B) are the mean and the standard deviation of the average rewards at 500 time-step, respectively. (C) and (D) are at 10^4 time-step.

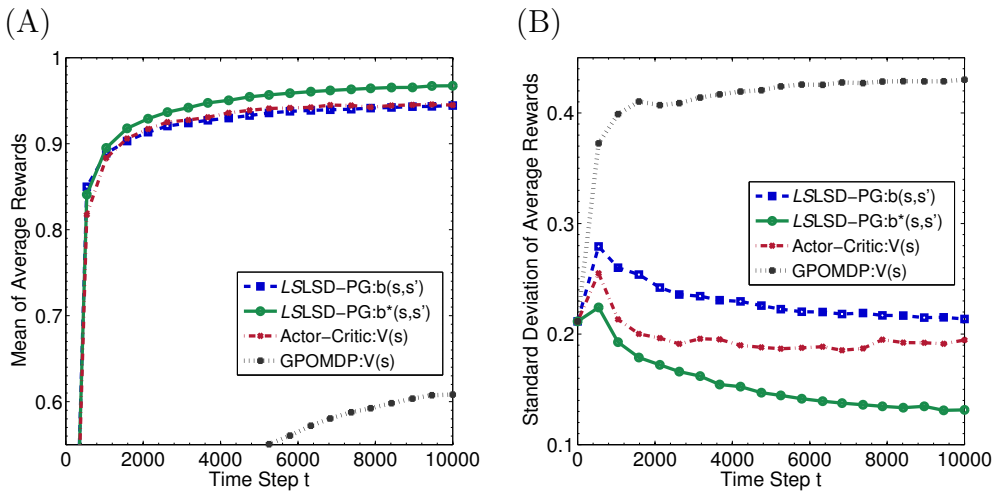

Figure 5: Comparison with various PG algorithms for the optimization of the policy parameters with the appropriate learning rate in the 3-state torus MDPs over 1,000 episodes.

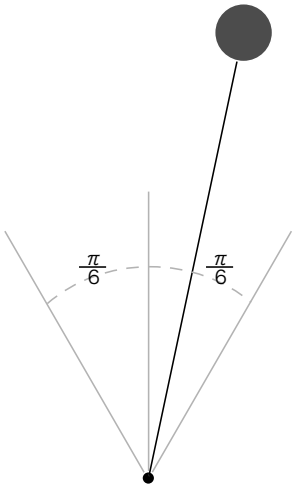

Figure 6: Pendulum balancing problem near the top ranges; $x \in [-\pi/6, \pi/6]$ and $\dot{x} \in [-\pi/2, \pi/2]$.

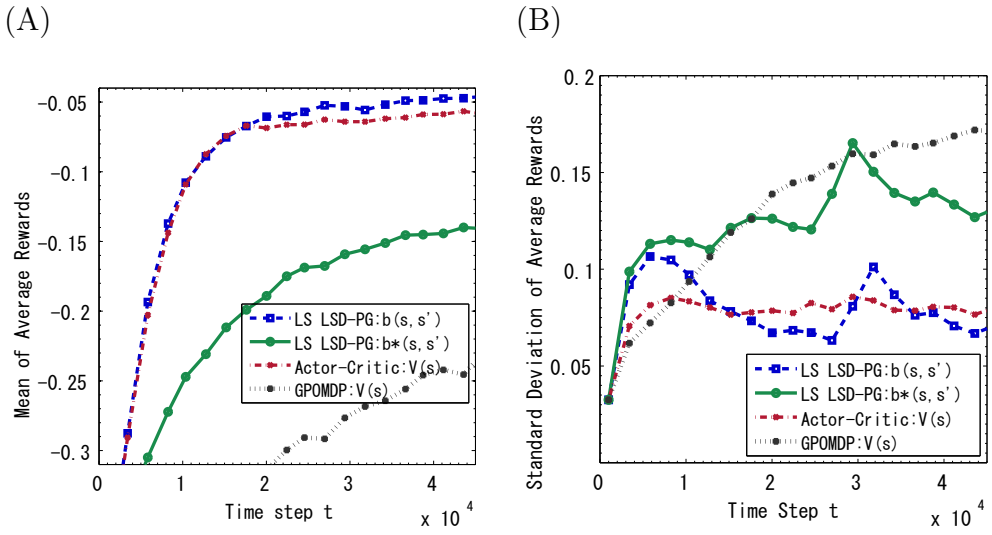

Figure 7: Comparison with various PG algorithms for the optimization of the policy parameters in the pendulum balancing problem over 500 episodes.