Policy Gradient Reinforcement Learning with Log Stationary Distribution Gradients

Tetsuro Morimura, Eiji Uchibe, Junichiro Yoshimoto, and Kenji Doya
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{morimura,uchibe,jun-y,doya}@oist.jp

September 21, 2007

Abstract

Conventional policy gradient reinforcement learning (PGRL) algorithms neglect a term in the average reward gradient, which is dependent upon the change in the stationary distribution by the policy parameter change. Although the bias introduced by this omission can be reduced by taking the discount factor γ close to 1, it increases the variance of the gradient estimate. Here, we propose a new policy gradient (PG) framework in which the gradient of the log stationary distribution (LSDG) is derived through backward Markov chain formulation and temporal difference learning algorithm. In this framework, the average reward gradient is estimated independent of γ, and by setting $\gamma = 0$, it becomes unnecessary to learn value functions. We also verify the performance of the proposed algorithms using simple benchmark tasks.

1 Introduction

Policy gradient reinforcement learning (PGRL) is a popular family of algorithms in reinforcement learning (RL) for improving a policy parameter to maximize the average reward by using the average reward gradients with respect to a policy parameter, which are called policy gradients (PG). Conventional PG algorithms [1, 2] neglect a term in PGs that is dependent upon the changes in the stationary distribution induced by the change in a the policy parameter; these are termed the (Log) Stationary Distribution Gradients—LSDG—since they are hard to identify. While the biases introduced by this omission can be reduced by taking a so-called discount factor γ close to 1, it increases the variance of the gradient estimates. This tradeoff makes it difficult to find an appropriate γ in practice. The mixing time of a Markov chain can be a criterion for determining an appropriate γ [1, 3]. However, the mixing time depends upon the policy, and hence it is generally hard to predict before learning is complete.

The average reward PG algorithm [4, 5] eliminates the use of the discount factor by introducing a differential cost function as a solution of Poisson's equation. However, it has been suggested [6] that there is theoretically no significant difference in performances between average reward PG and regular PG with discount factor γ close to 1.

Here, we propose a new PG framework in which LSDG as the gradient of the stationary distribution is derived through backward Markov chain formulation and a temporal difference learning algorithm. In this framework, the average reward gradient is also estimated independent of γ, and by setting $\gamma = 0$, it becomes unnecessary to learn value functions.

We should note that two methods to estimate the gradient of the (stationary) state distribution have already been proposed, although these are different from our proposal and have the following problems. The first is the method in operations research called "the likelihood ratio gradient" or "the score function" [7, 8]. However, their applicability is limited to regenerative processes [1] [1]. Another method proposed in [9] is not a direct estimation of the gradient of the state distribution and is done via the estimation of the state distribution with density propagation. Therefore, these methods require the knowledge of which state the agent is in, while our method only needs to observe the feature vector of the state, which may include some noise.

In Section 2, we review the conventional PG methods and outline our aim in estimating LSDG. In Section 3, we propose an LSDG(λ) algorithm for the estimation of LSDG by a least squares temporal difference method based on the backward Markov chain formulation. In Section 4, the LSDG(λ)-PG algorithm is presented, which utilizes LSDG(λ) and is a γ-free method. To verify the performances of the proposed algorithms, the numerical results in simple Markov Decision Processes (MDP) are shown in Section 5.

2 Policy Gradient Reinforcement Learning

We review the conventional PGRL methods and present the main idea of our new algorithm. A discrete time MDP with finite sets of states $\mathbb{X} \ni x$ and actions $\mathbb{U} \ni u$ is defined by a state transition probability $p(x_{+1}|x,u)$ and a reward function $r_{+1} = r(x_{+1}, x, u) \in [R_{min}, R_{max}]$, where x_{+1} is the next state given by an action u at a state x and r_{+1} is the observed immediate rewared at x_{+1}[10, 11]. x_{+k} and u_{+k} denote a state and an action after k time-steps from a state x and an action u, respectively, and vice versa in $-k$. The decision-making follows a stochastic policy $\pi_\theta(x,u) \equiv p(u|x;\boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$. We assume the policy $\pi_\theta(x,u)$ is differentiable with $\boldsymbol{\theta}$ for all $x \in \mathbb{X}$ and $u \in \mathbb{U}$ [2]. We also posit the following assumption:

[1] While the log stationary distribution gradient with respect to the policy parameter is one of the notations of the *likelihood ratio gradient* or *score function* and might be referred to as such; we term it LSDG in this paper.

[2] We set $\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x,u) = \mathbf{0}$ for $\pi_\theta(x,u) = 0$.

Assumption 1 *The Markov chain $M(\boldsymbol{\theta})$ with a state transition probability $p(x_{+1}|x,u)$ and a stochastic policy $\pi_\theta(x,u)$ is ergodic (irreducible and aperiodic) for all policy parameters. Then there exists a unique stationary state distribution $d^\pi(x) = \lim_{k\to\infty} p(x_{+k} = x|\pi_\theta) > 0$, which is independent of the initial state and satisfies*

$$d^\pi(x) = \sum_{x_{-1},u_{-1}} p_{M(\boldsymbol{\theta})}(x, u_{-1}|x_{-1}) d^\pi(x_{-1}), \tag{1}$$

where $p_{M(\boldsymbol{\theta})}(x,u_{-1}|x_{-1}) \equiv \pi_\theta(x_{-1},u_{-1})p(x|x_{-1},u_{-1})$.

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

$$R(\boldsymbol{\theta}) \equiv \lim_{K\to\infty} \frac{1}{K} \mathbb{E}_{M(\boldsymbol{\theta})}\left\{ \sum_{k=1}^{K} r_{+k} \middle| x \right\},$$

where $\mathbb{E}_{M(\boldsymbol{\theta})}$ denotes the expectation over the Markov chain $M(\boldsymbol{\theta})$. It is noted that, under Assumption 1, the average reward is independent of the initial state x and can be shown to equal [10]:

$$R(\boldsymbol{\theta}) = \sum_{x_{+1},x,u} d^\pi(x)\pi_\theta(x,u)p(x_{+1}|x,u)r(x_{+1},x,u). \tag{2}$$

The policy gradient RL algorithms update the policy parameter $\boldsymbol{\theta}$ in the direction of the gradient of the average reward $R(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

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

which is often referred as the policy gradient (PG) for short. This is given by

$$\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \sum_{x_{+1},x,u} d^\pi(x)\pi_\theta(x,u) \left(\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x,u) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x)\right) p(x_{+1}|x,u) r(x_{+1},x,u). \tag{3}$$

As the derivation of the gradient of the log stationary state distribution $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x)$ is nontrivial, the conventional PG algorithms [1, 2] utilize an alternative representation of the PG

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = &\sum_{x,u} d^\pi(x)\pi_\theta(x,u)\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x,u) Q^\pi_\gamma(x,u) \\ &+ (1-\gamma) \sum_{x} d^\pi(x) \nabla_{\boldsymbol{\theta}} \ln d^\pi(x) V^\pi_\gamma(x), \end{aligned} \tag{4}$$

where $Q^\pi_\gamma(x,u) \equiv \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{\sum_{k=1}^{K} \gamma^{k-1} r_{+k}|x,u\}$ is an action value function and $V^\pi_\gamma(x) \equiv \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{\sum_{k=1}^{K} \gamma^{k-1} r_{+k}|x\}$ is a state value function with discount factor $\gamma \in [0,1)$ [11].

Since the contribution of the second term of Eq.4 becomes smaller as γ approaches 1 [1], the conventional algorithms [1, 2] approximated the PG only from the first term by taking $\gamma \approx 1$. Although the bias introduced by this omission becomes smaller as γ is set close to 1, the variance of the estimate becomes larger.

Here we propose an alternative approach, which estimates the log stationary distribution gradient (LSDG), $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$, and uses Eq.3 for the derivation of the PG. A marked feature is that we do not need to learn the value function, and thus, the algorithm is free from the bias-variance trade-off in the choice of the discount factor γ.

3 Estimation of the Log Stationary Distribution Gradient

In this section, we propose an LSDG estimation algorithm based on least squares, LSDG(λ). For this purpose, we formulate the backwardness of the ergodic Markov chain $M(\boldsymbol{\theta})$, and show that LSDG can be estimated in the temporal difference framework [12, 13, 14].

3.1 Properties of forward and backward Markov chains

According to Bayes' theorem, a backward probability from a current state to a past state-action pair is given by

$$q(x_{-1}, u_{-1}|x) = \frac{p(x|x_{-1}, u_{-1})p(x_{-1}, u_{-1})}{\sum_{x_{-1}, u_{-1}} p(x|x_{-1}, u_{-1})p(x_{-1}, u_{-1})}.$$

The posterior $q(x_{-1}, u_{-1}|x)$ depends upon the prior distribution $p(x_{-1}, u_{-1})$. When the prior distribution follows the stationary distribution and the policy—$p(x_{-1}, u_{-1}) = \pi_{\theta}(x_{-1}, u_{-1})d^{\pi}(x_{-1})$—the posterior is termed as the *stationary* backward probability and the subscript $B(\boldsymbol{\theta})$ is appended to it, where it appears as $q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}|x)$,

$$\begin{aligned} q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}|x) &= \frac{p(x|x_{-1}, u_{-1})\pi_{\theta}(x_{-1}, u_{-1})d^{\pi}(x_{-1})}{d^{\pi}(x)} \\ &= \frac{p_{M(\boldsymbol{\theta})}(x, u_{-1}|x_{-1})d^{\pi}(x_{-1})}{d^{\pi}(x)}. \end{aligned} \tag{5}$$

If a Markov chain follows $q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}|x)$, we term it as the backward Markov chain $B(\boldsymbol{\theta})$ associated with $M(\boldsymbol{\theta})$ following $p_{M(\boldsymbol{\theta})}(x, u_{-1}|x_{-1})$. 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 $p_{M(\boldsymbol{\theta})}(x|x_{-1}) \equiv \sum_{u_{-1}} p_{M(\boldsymbol{\theta})}(x, u_{-1}|x_{-1})$ be irreducible and ergodic. Then* *the backward Markov chain $B(\boldsymbol{\theta})$ characterized by the backward (stationary) transition probability $q_{B(\boldsymbol{\theta})}(x_{-1}|x) \equiv \sum_{u_{-1}} q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}|x)$ against $p_{M(\boldsymbol{\theta})}$ is also ergodic and has the same unique stationary distribution of $M(\boldsymbol{\theta})$:*

$$d_{M(\boldsymbol{\theta})}(x) = d_{B(\boldsymbol{\theta})}(x), \tag{6}$$

where $d_{M(\boldsymbol{\theta})}(x) \equiv d^{\pi}(x)$ and $d_{B(\boldsymbol{\theta})}(x)$ are the stationary distributions of $M(\boldsymbol{\theta})$ and $B(\boldsymbol{\theta})$, respectively.

Proof: By multipling both sides of Eq.5 by $d^{\pi}(x)$ and summing over all possible $u_{-1} \in \mathbb{U}$, we obtain

$$q_{B(\boldsymbol{\theta})}(x_{-1}|x) d^{\pi}(x) = p_{M(\boldsymbol{\theta})}(x|x_{-1}) d^{\pi}(x_{-1}). \tag{7}$$

Then, $\sum_x q_{B(\boldsymbol{\theta})}(x_{-1}|x) d^{\pi}(x) = d^{\pi}(x_{-1})$ holds by summing both sides of Eq.7 over all possible $x \in \mathbb{X}$, indicating that (i) $B(\boldsymbol{\theta})$ has the same stationary distribution of $M(\boldsymbol{\theta})$ and (ii) $B(\boldsymbol{\theta})$ has the same irreducible property as $M(\boldsymbol{\theta})$. Eq.7 is reformulated by the transition probability, $p_{M(\boldsymbol{\theta})}(x|x_{-1})$ or $q_{B(\boldsymbol{\theta})}(x_{-1}|x)$, assembled to the matrix notation, $\boldsymbol{P}_{M(\boldsymbol{\theta})}$ or $\boldsymbol{Q}_{B(\boldsymbol{\theta})}$, respectively, and the stationary distribution to the vector notation $\boldsymbol{d}^{\pi}$:[3]

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

We can easily see that the diagonal component of $(\boldsymbol{P}_{M(\boldsymbol{\theta})})^n$ is equal to that 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 $M(\boldsymbol{\theta})$. Eq.6 is directly proven by (i)–(iii) [15]. $\square$

Proposition 2 *Let the distribution of x_{-K} follow $d^{\pi}(x)$; then, the expectations of both the directional Markov chains regarding the sum of arbitrary functions $f(x_{-k}, u_{-k})$ over $k \in [0, K]$ are equivalent:*

$$\mathbb{E}_{B(\boldsymbol{\theta})}\Big\{ \sum_{k=0}^{K} f(x_{-k}, u_{-k}) \Big| x \Big\} = \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{ \sum_{k=0}^{K} f(x_{-k}, u_{-k}) \Big| x, d^{\pi}(x_{-K}) \Big\}, \tag{8}$$

where $\mathbb{E}_{B(\boldsymbol{\theta})}$ and $\mathbb{E}_{M(\boldsymbol{\theta})}$ denote the expectations over the forward and backward Markov chains, $B(\boldsymbol{\theta})$ and $M(\boldsymbol{\theta})$, respectively, and $\mathbb{E}\{\cdot | d^{\pi}(x_{-K})\} \equiv \mathbb{E}\{\cdot | p(x_{-K}) = d^{\pi}(x_{-K})\}$. Eq.8 holds even at the limitation, $K \to \infty$.

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

$$\begin{aligned} &q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}, ..., x_{-K}, u_{-K}|x) \\ &\qquad = q_{B(\boldsymbol{\theta})}(x_{-1}, u_{-1}|x) \cdots q_{B(\boldsymbol{\theta})}(x_{-K}, u_{-K}|x_{-K+1}) \\ &\qquad \propto p_{M(\boldsymbol{\theta})}(x, u_{-1}|x_{-1}) \cdots p_{M(\boldsymbol{\theta})}(x_{-K+1}, u_{-K}|x_{-K}) d^{\pi}(x_{-K}). \end{aligned}$$

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

It instantly proves the proposition in the case of the finite K. Since the following equations are derived with Proposition 1, the proposition in the limit case $K \to \infty$ is also instantly proven,

$$\begin{aligned}\lim_{K\to\infty} \mathbb{E}_{B(\boldsymbol{\theta})}\{f(x_{-K}, u_{-K})|x\} &= \lim_{K\to\infty} \mathbb{E}_{M(\boldsymbol{\theta})}\{f(x_{-K}, u_{-K})|x, d^\pi(x_{-K})\}\\ &= \sum_{x,u} \pi_\theta(x,u) d^\pi(x) f(x,u).\end{aligned}$$

□

Propositions 1 and 2 are significant because thay indicate that the samples from the forward Markov chain $M(\boldsymbol{\theta})$ can be used directly for estimations concerning the backward Markov chain $B(\boldsymbol{\theta})$ under the state distribution converging the stationary distribution, and thus can be utilized in the following sections.

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

LSDG, $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x)$, is decomposed using Eq.5 to

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \ln d^\pi(x) &= \frac{1}{d^\pi(x)} \sum_{x_{-1},u_{-1}} p(x|x_{-1},u_{-1})\pi_\theta(x_{-1},u_{-1})d^\pi(x_{-1})\\ &\qquad\qquad \{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-1},u_{-1}) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-1})\}\\ &= \sum_{x_{-1},u_{-1}} q_{B(\boldsymbol{\theta})}(x_{-1},u_{-1}|x)\{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-1},u_{-1}) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-1})\}\\ &= \mathbb{E}_{B(\boldsymbol{\theta})}\{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-1},u_{-1}) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-1})|x\}. \end{aligned} \tag{9}$$

Noting that there exist $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x)$ and $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-1})$ in Eq.9, the recursion of Eq.9 yields

$$\nabla_{\boldsymbol{\theta}} \ln d^\pi(x) = \lim_{K\to\infty} \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-k},u_{-k}) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-K})\Big| x\Big\}. \tag{10}$$

Eq.10 implies that the LSDG of a state x is the infinite-horizon cumulation of the policy eligibility $\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x,u)$ through the backward Markov chain $B(\boldsymbol{\theta})$ from state x. From Eqs.9 and 10, LSDG could be estimated with temporal difference (TD) learning [12] concerning the following backward TD $\boldsymbol{\delta}$ on the backward Markov chain $B(\boldsymbol{\theta})$ rather than $M(\boldsymbol{\theta})$.

$$\boldsymbol{\delta}(x) \equiv \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-1},u_{-1}) + \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-1}) - \nabla_{\boldsymbol{\theta}} \ln d^\pi(x),$$

where the first two terms are regarded as the one-step actual observation of the policy eligibility[4] and the one-step ahead LSDG on $B(\boldsymbol{\theta})$ against the LSDG of current state, which is the last term. While $\boldsymbol{\delta}(x)$ is a random variable,

[4] While the TD for the value functions is well-known and concerns r on $M(\boldsymbol{\theta})$ [11], this TD for LSDG concerns $\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x,u)$ on $B(\boldsymbol{\theta})$.

$\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}(x)|x\} = \mathbf{0}$ holds. It motivates the minimization of the mean squares of the backward TD-error, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}(x)^2\}$ for the estimation of LSDG, where $\hat{\boldsymbol{\delta}}(x)$ is comprised by the LSDG estimate $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d^\pi(x)$ rather than LSDG $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x)$. $\boldsymbol{\delta}(x)^2$ denotes $\boldsymbol{\delta}(x)^\top \boldsymbol{\delta}(x)$ for simplicity.

With an eligibility decay rate $\lambda \in [0, 1]$ and a backtrace time-step $K \in \mathbb{N}$, Eq.10 is generalized:

$$\nabla_{\boldsymbol{\theta}} \ln d^\pi(x) = \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{ \sum_{k=1}^{K} \lambda^{k-1} \{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-k}, u_{-k}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-k})\} \\ + \lambda^K \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-K})|x \Big\}.$$

Along with this modification, the backward TD is modified into the backward TD(λ), $\boldsymbol{\delta}_{\lambda,K}(x)$,

$$\boldsymbol{\delta}_{\lambda,K}(x) \equiv \sum_{k=1}^{K} \lambda^{k-1} \Big\{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_{-k}, u_{-k}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-k})\Big\} \\ + \lambda^K \nabla_{\boldsymbol{\theta}} \ln d^\pi(x_{-K}) - \nabla_{\boldsymbol{\theta}} \ln d^\pi(x),$$

where the unbiased property, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}_{\lambda,K}(x)|x\} = \mathbf{0}$, is still retained. The minimization of $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2\}$ in $\lambda = 1$ and the limit $K \to \infty$ is regarded as the Widrow-Hoff supervised learning procedure. Even in a larger λ and K instead of the above setting, this minimization would be less sensitive to a non-Markovian effect as in the case of the conventional TD(λ) learning for the value functions [16].

In order to minimize $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2\}$ as the estimation of LSDG, we need to gather many samples drawn from the backward Markov chain $B(\boldsymbol{\theta})$; however, actual samples are drawn from a forward Markov chain $M(\boldsymbol{\theta})$. Fortunately, utilizing Propositions 1 and 2, we can use the following exchangeable property:

$$\begin{aligned} \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2\Big\} &= \sum_x d_{B(\boldsymbol{\theta})}(x)\, \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2|x\Big\} \\ &= \sum_x \ d^\pi(x)\ \ \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2|x, d^\pi(x_{-K})\Big\} \\ &= \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2|d^\pi(x_{-K})\Big\}. \end{aligned} \tag{11}$$

Namely, the actual samples can be reused for minimizing $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2\}$, provided $x_{-K} \sim d^\pi(x)$. In real problems, however, the initial state is rarely drawn from the stationary distribution $d^\pi(x)$. 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 LSDG estimation algorithm: Least squares on backward TD(λ) with constraint

In the previous sections, we introduced the theory that the estimation of LSDG is conducted by the minimization of the mean squares of $\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2$ on $M(\boldsymbol{\theta})$, $\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(x)^2|d^\pi(x_{-K})\}$. However, LSDG also has the following constraint derived from $\sum_x d^\pi(x)=1$:

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{\nabla_{\boldsymbol{\theta}} \ln d^\pi(x)\} = \sum_x d^\pi(x)\nabla_{\boldsymbol{\theta}} \ln d^\pi(x) = \nabla_{\boldsymbol{\theta}} \sum_x d^\pi(x) = \mathbf{0}. \tag{12}$$

In this section, we propose an LSDG estimation algorithm, LSDG(λ), based on least squares [17, 13, 14], which simultaneously attempts to decrease the mean squares and satisfy the constraint. We consider the situation that the LSDG estimate $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d^\pi(x)$ is represented by a linear vector function approximator $\boldsymbol{f}(x;\boldsymbol{\Omega}) \equiv \boldsymbol{\Omega}\boldsymbol{\phi}(x)$, where $\boldsymbol{\phi}(x)\in\mathbb{R}^e$ is a basis function and $\boldsymbol{\Omega}\equiv[\boldsymbol{\omega}_1,...,\boldsymbol{\omega}_d]^\top\in\mathbb{R}^{d\times e}$ is an adjustable parameter matrix, and assume that the optimal parameter $\boldsymbol{\Omega}^*$ satisfies $\nabla_{\boldsymbol{\theta}} \ln d^\pi(x) = \boldsymbol{\Omega}^*\boldsymbol{\phi}(x)$ [5]. For simplicity, we focus our attention only on the i'th element θ_i of the policy parameter $\boldsymbol{\theta}$, notating $f(x;\boldsymbol{\omega}_i)\equiv\boldsymbol{\omega}_i^\top\boldsymbol{\phi}(x)$ and $\nabla_{\theta_i}\ln\pi_\theta(x,u) \equiv \partial\ln\pi_\theta(x,u)/\partial\theta_i$ and $\hat{\delta}_{\lambda,K}(x,\boldsymbol{\omega}_i)$ as the i'th element of $\hat{\boldsymbol{\delta}}_{\lambda,K}(x)$. Accordingly, the objective function to be minimized is

$$\varepsilon(\boldsymbol{\omega}_i) = \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)^2|d^\pi(x_{-K})\} + \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(x;\boldsymbol{\omega}_i)\}^2, \tag{13}$$

where the second term of the right side is for the constraint of Eq.12 [6]. Then, the derivative is

$$\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) = \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)\ \nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)|d^\pi(x_{-K})\} + \frac{1}{2}\nabla_{\boldsymbol{\omega}_i}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(x;\boldsymbol{\omega}_i)\}^2, \tag{14}$$

where

$$\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i) = \sum_{k=1}^{K}\lambda^{k-1}\nabla_{\theta_i}\log\pi_\theta(x_{-k},u_{-k}) + \boldsymbol{\omega}_i^\top\ \nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i),$$

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

Although the conventional least squares aim 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 $\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)$ and its derivative $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)$

[5] If the estimator cannot represent LSDG exactly, LSDG(λ) would behave as suggested in [12, 16], which will be confirmed in a numerical experiment. However, we do not analyze it theoretically.

[6] As LSDG(λ) consider the two objectives in equal measure, we can instantly extend it for the problem minimizing $\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_\lambda^2(x)|d^\pi(x_{-K})\}$ subject to the constraint of Eq.12 with the Lagrange multiplier method.

concerning the first term of the right-hand side in Eq.13. That is, if

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)\ \nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)|d^\pi(x_{-K})\} \neq \mathbf{0},$$

then $\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i^*) \neq \mathbf{0}$. Since this correlation exists in general RL problems, we apply the instrumental variable method to eliminate the bias [17, 13]. It requires that $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)$ is replaced by the instrumental variables $\boldsymbol{\iota}(x)$ that has a correlation with $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)$ but not $\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i^*)$. This condition is obviously satisfied when $\boldsymbol{\iota}(x) = \boldsymbol{\phi}(x)$ as well as LSTD(λ) [13, 14]. Instead of Eq.14, we aim to find the parameter making the equation

$$\widetilde{\nabla}_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) \equiv \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)\boldsymbol{\phi}(x)|d^\pi(x_{-K})\} + \mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(x)\}\mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(x)\}^\top\boldsymbol{\omega}_i \tag{15}$$

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

From here, we change the notation to x_t denoting the state at time-step t on the actual Markov chain $M(\boldsymbol{\theta})$. The proposed LSDG estimation algorithm, LSDG(λ) sets that the backtrace time-step K is equal to the time-step t of the current state x_t under the eligibility decay rate $\lambda \in [0, 1)$. That is,

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

where $g_{\lambda,i}(x_t) = \sum_{s=0}^{t}\lambda^{t-s}\nabla_{\theta_i}\ln\pi_\theta(x_s,u_s)$ and $\boldsymbol{z}_\lambda(x_t) = (1-\lambda)\sum_{s=1}^{t}\lambda^{t-s}\boldsymbol{\phi}(x_s) + \lambda^t\boldsymbol{\phi}(x_0)$. The expectations in Eq.15 are estimated by [7]

$$\begin{aligned}\lim_{K\to\infty}\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(x;\boldsymbol{\omega}_i)\boldsymbol{\phi}(x)|d^\pi(x_{-K})\} &\simeq \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\phi}(x_t)(g_{\lambda,i}(x_t) + \boldsymbol{\phi}(x_t) - \boldsymbol{z}_\lambda(x_{t-1}))^\top\boldsymbol{\omega}_i \\ &= \boldsymbol{b}_T + \boldsymbol{A}_T\ \boldsymbol{\omega}_i,\end{aligned}$$

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

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

Therefore, by substituting these estimators to Eq.15, the estimate $\hat{\boldsymbol{\omega}}_i^*$ at time-step T is computed by

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

LSDG(λ) for the case of the matrix parameter $\hat{\boldsymbol{\Omega}}^*$ rather than $\hat{\boldsymbol{\omega}}_i^*$ is shown at Algorithm 1.

[7] When the limit $T \to \infty$ at $\lambda \in [0, 1)$, these estimators converge to the true values. Although it could be proven based on the results of [13, 14], we omit the proof here.

Algorithm 1
LSDG(λ): Estimation for $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$

Given:
- a policy $\pi_\theta(x, u)$ with a fixed $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(x)$.

Initialize: $\lambda \in [0, 1)$.
Set: $\boldsymbol{c} := \mathbf{0}$; $\boldsymbol{z} \ = \mathbf{0}$; $\boldsymbol{g} := \mathbf{0}$; $\boldsymbol{A} := \mathbf{0}$; $\boldsymbol{B} := \mathbf{0}$.
for $t = 0$ **to** $T - 1$ **do**
  if $t = 0$ **then**
    $\boldsymbol{z} := \boldsymbol{\phi}(x_0)$; $\boldsymbol{c} := \boldsymbol{\phi}(x_0)$;
  else
    $\boldsymbol{z} := \lambda \boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(x_t)$;
  end if
  $\boldsymbol{c} := \boldsymbol{c} + \boldsymbol{\phi}(x_{t+1})$;
  $\boldsymbol{g} := \lambda \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_t, u_t)$;
  $\boldsymbol{A} := \boldsymbol{A} + \boldsymbol{\phi}(x_{t+1})(\boldsymbol{\phi}(x_{t+1}) - \boldsymbol{z})^\top$;
  $\boldsymbol{B} := \boldsymbol{B} + [\boldsymbol{\phi}(x_{t+1}), ..., \boldsymbol{\phi}(x_{t+1})]\,\mathrm{diag}(\boldsymbol{g})$;
end for
$\boldsymbol{\Omega} := (\boldsymbol{A} + \boldsymbol{c}\boldsymbol{c}^\top / t)^{-1} \boldsymbol{B}$;
Return: $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d^{\pi}(x) = \boldsymbol{\Omega}\, \boldsymbol{\phi}(x)$.

Algorithm 2
LSDG(λ)-PG: Optimization for the policy

Given:
- a policy $\pi_\theta(x_t, u_t)$ with an adjustable $\boldsymbol{\theta}$,
- a feature vector function of state $\boldsymbol{\phi}(x)$.

Initialize: $\boldsymbol{\theta}$, $\lambda \in [0, 1)$, $\beta \in [0, 1)$, α.
Set: $\boldsymbol{c} := \mathbf{0}$; $\boldsymbol{z} \ = \mathbf{0}$; $\boldsymbol{g} := \mathbf{0}$; $\boldsymbol{A} := \mathbf{0}$; $\boldsymbol{B} := \mathbf{0}$.
for $t = 0$ **to** $T - 1$ **do**
  if $t = 0$ **then**
    $\boldsymbol{z} := \boldsymbol{\phi}(x_0)$; $\boldsymbol{c} := \boldsymbol{\phi}(x_0)$;
  else
    $\boldsymbol{z} := \lambda \boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(x_t)$;
  end if
  $\boldsymbol{c} := \beta \boldsymbol{c} + \boldsymbol{\phi}(x_{t+1})$;
  $\boldsymbol{g} := \beta\lambda \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_t, u_t)$;
  $\boldsymbol{A} := \beta \boldsymbol{A} + \boldsymbol{\phi}(x_{t+1})(\boldsymbol{\phi}(x_{t+1}) - \boldsymbol{z})^\top$;
  $\boldsymbol{B} := \beta \boldsymbol{B} + [\boldsymbol{\phi}(x_{t+1}), ..., \boldsymbol{\phi}(x_{t+1})]\,\mathrm{diag}(\boldsymbol{g})$;
  $\boldsymbol{\Omega} := (\boldsymbol{A} + (1 - \beta)\, \boldsymbol{c}\boldsymbol{c}^\top)^{-1} \boldsymbol{B}$;
  $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha r_{t+1} \{\nabla_{\boldsymbol{\theta}} \ln \pi_\theta(x_t, u_t) + \boldsymbol{\Omega}^\top \boldsymbol{\phi}(x_t)\}$;
end for
Return: $p(u|x; \boldsymbol{\theta}) = \pi_\theta(x, u)$.

4 Policy update by the LSDG estimate

Now let us define the PGRL algorithm based on the above LSDG estimate. The realization of the estimation for $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$ by LSDG(λ) instantly derives the following estimate for the PG (Eq.3), being independent of the discount factor γ:

$$\widehat{\nabla}_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \frac{1}{T} \sum_{t=0}^{T-1} \Big(\nabla_{\boldsymbol{\theta}} \ln \pi_{\theta}(x_{x_t}, u_{x_t}) + \widehat{\nabla}_{\boldsymbol{\theta}} \ln d^{\pi}(x_{x_t}) \Big) r_{t+1}.$$

The policy parameter can then be updated through the stochastic gradient method with an appropriate stepsize α [18]:[8]

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha(\nabla_{\boldsymbol{\theta}} \ln \pi_{\theta}(x_{x_t}, u_{x_t}) + \widehat{\nabla}_{\boldsymbol{\theta}} \ln d^{\pi}(x_{x_t})) r_{t+1},$$

where := denotes the substitution of the right to the left. LSDG(λ)-PG is shown at Algorithm 2 as one of the simplest realizations on PG algorithm, utilizing LSDG(λ), where the decay rate parameter $\beta \in [0, 1)$ is introduced to discard the past estimate given by old values of $\boldsymbol{\theta}$.

This constitutes the other important topic for function estimation: how to set the basis function $\boldsymbol{\phi}(x)$, particularly in the continuous state problems. For the PG estimate, the objective concerning the LSDG estimate is only to approximate $\sum_{x,u} d^{\pi}(x)\pi_{\theta}(x,u)\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x) r(x,u)$, even when the estimate cannot precisely represent LSDG. Therefore, the following proposition would be useful:

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

$$\phi(x) = \sum_{u} \pi_{\theta}(x,u) r(x,u),$$

then the function estimator, $\boldsymbol{f}(x;\boldsymbol{\omega}) = \boldsymbol{\omega} \sum_u \pi_{\theta}(x,u) r(x,u)$, *has the ability to represent the second term of the PG,* $\sum_{x,u} d^{\pi}(x)\pi_{\theta}(x,u)\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x) r(x,u)$, *where the adjustable parameter* $\boldsymbol{\omega}$ *is a vector* $\mathbb{R}^{d\times 1}$*:*

$$\sum_{x,u} d^{\pi}(x)\pi_{\theta}(x,u) r(x,u) \nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x) = \sum_{x,u} d^{\pi}(x)\pi_{\theta}(x,u) r(x,u) \boldsymbol{f}(x;\boldsymbol{\omega}^{\star}),$$

where $\boldsymbol{\omega}^{\star}$ *minimizes the mean error,* $\epsilon(\boldsymbol{\omega}) = \frac{1}{2}\sum_x d^{\pi}(x)\{\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x) - \boldsymbol{f}(x;\boldsymbol{\omega})\}^2$.

Proof: It is proven by

$$\nabla_{\boldsymbol{\omega}} \epsilon(\boldsymbol{\omega}^{\star}) = \sum_{x,u} d^{\pi}(x)\pi_{\theta}(x,u) r(x,u) \{\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x) - \boldsymbol{f}(x;\boldsymbol{\omega}^{\star})\} = \mathbf{0}.$$

□

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

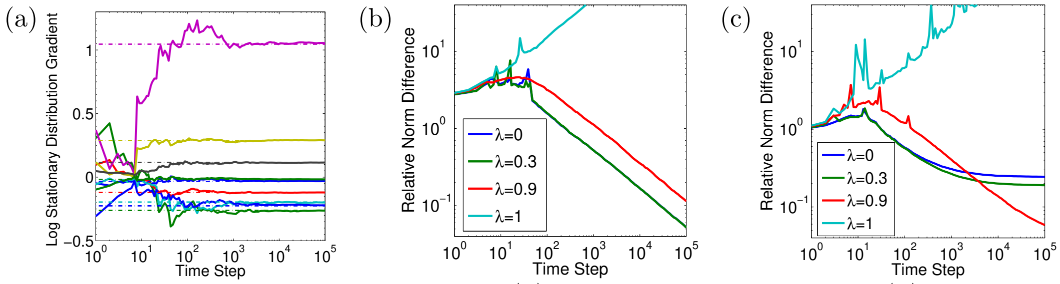

Figure 1: Performances of LSDG(λ) in the estimation of $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$.

5 Numerical Experiments

We verified the performance of our proposed algorithms in a stochastic "one-dimensional torus grid-world" with a finite set of grids $\mathbb{X} = \{1, .., m\}$ and a set of two possible actions $\mathbb{U} = \{L, R\}$. This is a typical m-state MDP task where the state transition probabilities are given by

$$p(x|x_{-1}=i, u_{-1}=L) = \begin{cases} p_i & \text{if } x=i-1 \\ \frac{1-p_i}{2} & \text{if } x=i \text{ or } i+1 \\ 0 & \text{otherwise,} \end{cases}$$

$$p(x|x_{-1}=i, u_{-1}=R) = \begin{cases} p_i & \text{if } x=i+1 \\ \frac{1-p_i}{2} & \text{if } x=i \text{ or } i-1 \\ 0 & \text{otherwise,} \end{cases}$$

where $x = 0$ and $x = m$ ($x = 1$ and $x = m + 1$) are the identical states and $p_i \in [0, 1]$ ($i = 1, \ldots, m$) is a task-dependent constant. In our experiments, a stochastic policy was represented by a sigmoidal function: $\pi_\theta(x, u = L) = 1 - \pi_\theta(x, u = R) = 1/(1 + \exp(\boldsymbol{\theta}^\top \boldsymbol{\phi}(x))$. Here, all elements of state-feature vectors $\boldsymbol{\phi}(1), \ldots, \boldsymbol{\phi}(m) \in \mathbb{R}^m$ were independently drawn from the standard normal distribution $N(0, 1^2)$ for each simulation run. This was for verifying how the parameterization of the stochastic policy affected the performance of our algorithms. The state-feature vector $\boldsymbol{\phi}(x)$ was also used as the basis function of the LSDG estimate. Each simulation run had a single episode consisting of more than 10^5 time steps.

At first, we verified how precisely LSDG(λ) estimates $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$ regardless of problem setting and the policy parameter $\boldsymbol{\theta}$. To this end, task-dependent constants $p_1, \ldots, p_m$ were independently selected from the uniform distribution over the interval of $[0.7, 1]$ and fixed during each simulation run. Policy parameter $\boldsymbol{\theta}$ was also randomly selected according to the normal distribution $N(0, 0.5^2)$ and fixed during each simulation run. Fig.1 (a) shows a typical time course of the LSDG estimate $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$ in case of $m = 3$, where nine different colors indicate different elements of $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$. The solid lines denote the values estimated by LSDG(0), and the dotted lines denote the analytical solution of $\nabla_{\boldsymbol{\theta}} \ln d^{\pi}(x)$. As shown in Fig.1 (a), we confirmed that the estimate

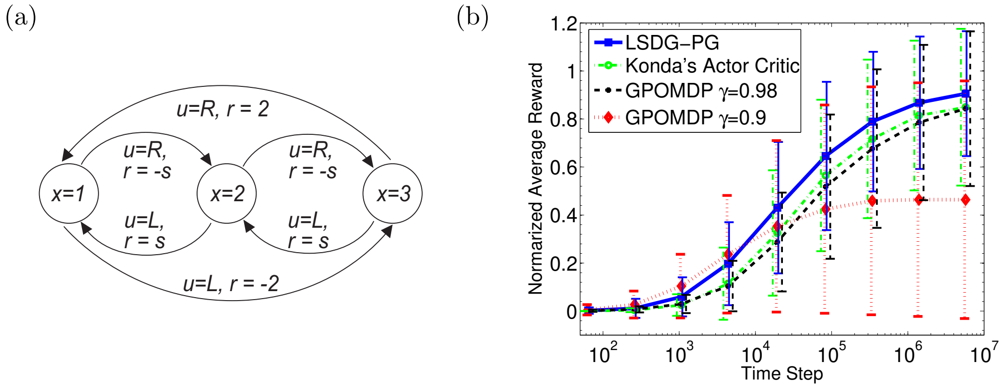

Figure 2: (a) Reward setting of a 3-state task used in our comparative study. (b) Comparison with various PG algorithms. The error bars indicate standard deviation over 100 simulation runs.

of LSDG(0) converged to the analytical solution throughout 100 simulation runs in case of $m = 3$. Then, we investigated the effect of the eligibility decay rate λ using 7-state tasks. In order to evaluate the average performance over various settings, we employed a "relative error" criterion that was defined by $\mathbb{E}_{M(\boldsymbol{\theta})}\{(\boldsymbol{f}(x;\boldsymbol{\Omega}^\star) - \boldsymbol{f}(x;\boldsymbol{\Omega}))^2\}/\mathbb{E}_{M(\boldsymbol{\theta})}\{(\boldsymbol{f}(x;\boldsymbol{\Omega}^\star)^2\}$, where $\boldsymbol{\Omega}^\star$ was the optimal parameter defined in Proposition 3 and was analytically calculated. Fig.1 (b) and (c) shows the time courses of relative error averages over 200 simulation runs 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 $\boldsymbol{\phi}(x)$. The feature-vectors $\boldsymbol{\phi}(x) \in \mathbb{R}^7$ used in (b) were appropriate and enough to distinguish all the different states, while the feature-vectors $\boldsymbol{\phi}(x) \in \mathbb{R}^6$ used in (c) were inappropriate. These results were consistent with theoretical prospects. Namely, we could set λ arbitrarily except for $\lambda = 1$ if the basis function was appropriate (Fig.1 (b)), otherwise we would need to set λ at a large value except for $\lambda = 1$ (Fig.1 (c)).

Finally, we compared the LSDG(λ)-PG with the other existing PG methods in a 3-state task, where state transition probabilities were at $p_i = 1$ for all $i \in \{1, 2, 3\}$. Fig.2 (a) shows the reward setting in this task. There are two types of rewards: constants "$r = (\pm)2$" and a variable "$r = (\pm)s$." The variable s was randomized by the uniform distribution over $[0.8, 1)$ for each simulation run. Note that the reward s defines the minimum value of γ to find the optimal policy: $\gamma^2 + \gamma > \frac{2s}{2-s}$. Therefore, the setting of γ is important and difficult in this task. From the performance baselines of the existing PG methods, we adopted two algorithms: GPOMDP [1] and Konda's actor-critic [5]. Fig.2 (b) shows the performance of three methods averaged over 100 simulation runs. Here, the performance was evaluated by $3R(\boldsymbol{\theta})/(2 - 2s)$, that is, the average reward normalized by its upper bound $(2 - 2s)/3$. The result shows that our LSDG(λ)-PG outperformed the other PG methods.

6 Conclusion

We showed that the actual forward and backward Markov chains are closely related and have common properties in the propositions. Utilizing these, we proposed LSDG(λ) as the estimation algorithm of the log stationary distribution gradient (LSDG), and LSDG(λ)-PG as the PG algorithm utilizing the LSDG estimate. The experimental results also demonstrated that LSDG(λ) could work at $\lambda \in [0, 1)$ and LSDG(λ)-PG could learn independent of γ. Additional theoretical and experimental verification and validation are necessary to further understand the properties and efficacy of LSDG algorithms.

References

[1] J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[2] H. Kimura and S. Kobayashi. An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value function. In *International Conference on Machine Learning*, 1998.

[3] S. Kakade. Optimizing average reward using discounted rewards. In *Annual Conference on Computational Learning Theory*, volume 14. MIT Press, 2001.

[4] J. N. Tsitsiklis and B. Van Roy. Average cost temporal-difference learning. *Automatica*, 35(11):1799–1808, 1999.

[5] V. S. Konda and J. N. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166, 2001.

[6] J. N. Tsitsiklis and B. Van Roy. On average versus discounted reward temporal-difference learning. *Machine Learning*, 49(2):179–191, 2002.

[7] P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1991.

[8] R. Y. Rubinstein. How to optimize discrete-event system from a single sample path by the score function method. *Annals of Operations Research*, 27(1):175–212, 1991.

[9] A. Y. Ng, R. Parr, and D. Koller. Policy search via density estimation. In *Advances in Neural Information Processing Systems*. MIT Press, 2000.

[10] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Volumes 1 and 2*. Athena Scientific, 1995.

[11] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[12] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

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

[14] J. A. Boyan. Least-squares temporal difference learning. *Machine Learning*, 49(2-3):233–246, 2002.

[15] R. B. Schinazi. *Classical and Spatial Stochastic Processes*. Birkhauser, 1999.

[16] J. Peng and R. J. Williams. Incremental multi-step Q-learning. *Machine Learning*, 22:283–290, 1996.

[17] P. Young. *Recursive Esimation and Time-series Analysis*. Springer-Verlag, 1984.

[18] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.