Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Mixing-Time Regularized Policy Gradient

Tetsuro Morimura
IBM Research - Tokyo
5-6-52 Toyosu, Koto-ku
Tokyo, Japan
tetsuro@jp.ibm.com

Takayuki Osogami
IBM Research - Tokyo
5-6-52 Toyosu, Koto-ku
Tokyo, Japan
osogami@jp.ibm.com

Tomoyuki Shirai
Kyushu University
744 Motooka, Nishi-ku
Fukuoka, Japan
shirai@imi.kyushu-u.ac.jp

Abstract

Policy gradient reinforcement learning (PGRL) has been receiving substantial attention as a mean for seeking stochastic policies that maximize cumulative reward. However, the learning speed of PGRL is known to decrease substantially when PGRL explores the policies that give the Markov chains having long mixing time. We study a new approach of regularizing how the PGRL explores the policies by the use of the hitting time of the Markov chains. The hitting time gives an upper bound on the mixing time, and the proposed approach improves the learning efficiency by keeping the mixing time of the Markov chains short. In particular, we propose a method of temporal-difference learning for estimating the gradient of the hitting time. Numerical experiments show that the proposed method outperforms conventional methods of PGRL.

1 Introduction

Policy Gradient Reinforcement Learning (PGRL) attempts to find a policy that maximizes the average reward, based on gradient ascent in the policy parameter space (Gullapalli 1990; Williams 1992; Baxter and Bartlett 2001). Since PGRLs can optimize the parameters controlling the measure of randomness of the policy, PGRLs, as compared with value-function-based approaches (Sutton and Barto 1998), can find appropriate stochastic policies. Meanwhile, PGRL methods often require an excessively large number of learning steps to construct a good stochastic policy. The number of learning steps depends on the mixing time of the Markov chains that are given by intermediate policies that the PGRL explores (Bartlett and Baxter 2000; Baxter and Bartlett 2000; 2001; Kakade 2003). Roughly speaking the mixing time of a Markov chain represents the number of steps needed for the Markov chain to approach sufficiently close to its stationary distribution. In this paper, we give a new PGRL method that regularizes the hitting time as a bound of a mixing time, where hitting-time regressions based on temporal-difference learning are proposed. This will keep the Markov chain compact and can improve the learning efficiency.

The organization and the contributions of this paper are summarized as follows. In Section 2, we briefly review the PGRL and also present a motivation and outline of our mixing-time regularization. The relation among a bias and variance of an estimator for an arbitrary linear sum of the stationary distribution on a Markov chain, a mixing time, and a hitting time is described in Section 3. It is proved as our first theoretical contribution that the bias and variance of the estimator are bounded via the Cesàro mixing time, which in turn is bounded via the worst-case expected hitting time. In Section 4, we derive a new framework of PGRL with mixing-time regularization, where, as the second contribution, the sufficient condition for the convergence to a local optimum is also provided in terms of the strength of the regularization term. The estimating method of this regularization term is proposed in Section 5. Numerical experiments in Section 6 show that the proposed method outperforms conventional PGRL methods.

2 Background of Policy Gradient Reinforcement Learning

Problems of PGRL are usually modeled on a Markov decision process (MDP) (Bertsekas 1995; Sutton and Barto 1998). It is defined by the quintuplet $(\mathcal{S}, \mathcal{A}, p_{\rm T}, R, \pi)$, where $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$ and $\mathcal{A} = \{1, \ldots, |\mathcal{A}|\}$ are finite sets of states and actions, respectively. Also, $p_{\rm T} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0, 1]$ is a state transition probability function of a state s_t, an action a_t, and the following state s_{t+1} at time $t \in \mathbb{N}$, i.e.,[1] $p_{\rm T}(s_{t+1}|s_t, a_t) \triangleq \Pr(s_{t+1}|s_t, a_t)$. The $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a bounded reward function of s_t, a_t, and s_{t+1}, which defines an immediate reward $r_{t+1} = R(s_t, a_t, s_{t+1})$ observed by a learning agent at each time step. The action probability function $\pi : \mathcal{A} \times \mathcal{S} \times \mathbb{R}^d \to [0, 1]$ defines the decision-making rule of the learning agent, which is also called a policy, i.e., $\pi(a|s; \boldsymbol{\theta}) \triangleq \Pr(a|s, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a policy parameter. The policy is parametrized by users and is controlled by tuning $\boldsymbol{\theta}$. Here, we make the following usual assumptions for the MDP (Bartlett and Baxter 2000; Baxter and Bartlett 2001; Kakade 2002).

[1] Although it should be $\Pr(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t)$ for the random variables S_{t+1}, S_t, and A_t to be precise, we write $\Pr(s_{t+1}|s_t, a_t)$ for brevity. The same rule is applied to the other distributions.

Assumption 1 *The Markov chain on $\mathcal{S}$ prescribed by $\mathrm{M}(\boldsymbol{\theta}) \triangleq \{\mathcal{S}, \mathcal{A}, p_T, R, \pi, \boldsymbol{\theta}\}$ is always ergodic (irreducible and aperiodic).*

Under Assumption 1, there exists a unique stationary distribution, $d_{\boldsymbol{\theta}}(s)$, which satisfies the balance equations:

$$d_{\boldsymbol{\theta}}(s_{t+1}) = \sum_{s_t\in\mathcal{S}}\sum_{a_t\in\mathcal{A}} p_{\mathrm{T}}(s_{t+1}|s_t,a_t)\pi(a_t|s_t;\boldsymbol{\theta})d_{\boldsymbol{\theta}}(s_t).$$

This stationary distribution is equal to the limiting distribution and independent of the initial state, $d_{\boldsymbol{\theta}}(s') = \lim_{t\to\infty}\Pr(S_t = s'|S_0 = s, \mathrm{M}(\boldsymbol{\theta}))$, $^\forall s \in \mathcal{S}$.

The goal of PGRL is to find the policy parameter $\boldsymbol{\theta}^*$ that maximizes the average immediate reward in the infinitely long run, so called the *average reward*,

$$\begin{aligned}\eta(\boldsymbol{\theta}) &\triangleq \lim_{T\to\infty}\frac{1}{T}\mathbb{E}_{\boldsymbol{\theta}}\left[\sum_{t=0}^{T-1} R(S_t, A_t, S_{t+1})\right]\\ &= \sum_{s_t\in\mathcal{S}}\sum_{a_t\in\mathcal{A}}\sum_{s_{t+1}\in\mathcal{S}} d_{\boldsymbol{\theta}}(s_t)\pi(a_t|s_t;\boldsymbol{\theta})\\ &\qquad\qquad \times p_{\mathrm{T}}(s_{t+1}|s_t,a_t)R(s_t,a_t,s_{t+1}),\end{aligned}$$

where $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ denotes the expectation operator over random variables in the Markov chain $\mathrm{M}(\boldsymbol{\theta})$, e.g., S_t, A_t, etc.

The derivative of the average reward with respect to the policy parameter,$\boldsymbol{\nabla}\eta(\boldsymbol{\theta}) \triangleq [\partial\eta(\boldsymbol{\theta})/\partial\theta_1, ..., \partial\eta(\boldsymbol{\theta})/\partial\theta_d]^\top$, is referred to as the Policy Gradient (PG). The average reward η is increased by updating the policy parameter $\boldsymbol{\theta}_t$ at time t as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{G}_{\boldsymbol{\theta}_t}\boldsymbol{\nabla}\eta(\boldsymbol{\theta}_t), \tag{1}$$

where α_t is a learning rate. The matrix $\boldsymbol{G}_{\boldsymbol{\theta}} \in \mathbb{R}^{d\times d}$ is an arbitrary uniformly-bounded positive definite matrix, which often consists of the Fisher information matrix of the policy (Kakade 2002) and/or the stationary state distribution (Morimura et al. 2008). This framework is called the PGRL (Baxter and Bartlett 2001). In the normal setting of reinforcement learning, the state transition probability or the reward function is unknown. Thus the PG $\boldsymbol{\nabla}\eta(\boldsymbol{\theta})$ cannot be computed analytically and thus needs to be estimated. However, as is described in the next section, the number of time steps needed to make the estimated $\boldsymbol{\nabla}\eta(\boldsymbol{\theta})$ almost unbiased will increase as the mixing time of the Markov chain $\mathrm{M}(\boldsymbol{\theta})$ gets larger.

To control the magnitude of the mixing time, we consider adding a regularization term, $l(\boldsymbol{\theta})$, into the policy update of Eq. (1) as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{G}_{\boldsymbol{\theta}_t}\boldsymbol{\nabla}\eta(\boldsymbol{\theta}_t) - \lambda_t l(\boldsymbol{\theta}_t), \tag{2}$$

where λ_t is a regularization parameter.

3 Mixing time and hitting time for PGRL

We introduce a mixing time and analyze its effect on biases and variances of estimators required for the policy update, such as $\boldsymbol{\nabla}\eta(\boldsymbol{\theta})$ or $\boldsymbol{G}_{\boldsymbol{\theta}}$ in Eq. (1), in a finite-time Markov chain. Then, as a bound of the mixing time, a hitting time is also introduced.

Cesàro mixing time

In PGRL, a standard mixing time[2], which is the time to be close to the stationary state distribution $d_{\boldsymbol{\theta}}(s)$, is often used for connection with the policy gradient estimator with discount factor (Bartlett and Baxter 2000; Baxter and Bartlett 2000; 2001). Here we consider an alternative formula of a mixing time which will be useful for discussing effects on an estimation problem in a finite-time Markov chain as follows. That is the *Cesàro mixing time* $m(\varepsilon, \boldsymbol{\theta})$ (Lovász and Winkler 1998; Levin, Peres, and Wilmer 2008), which is defined as

$$m(\varepsilon,\boldsymbol{\theta}) \triangleq \min_t\left\{t \;\middle|\; \max_{s_0\in\mathcal{S}}\frac{1}{2}\sum_{s\in\mathcal{S}}|\nu_{\boldsymbol{\theta}}(t,s_0,s) - d_{\boldsymbol{\theta}}(s)| \le \varepsilon\right\}, \tag{3}$$

where $\nu_{\boldsymbol{\theta}}(t, s_0, s)$ is the time-average probability of visiting s within t time-steps given an initial state $S_0 = s_0$,

$$\nu_{\boldsymbol{\theta}}(t,s_0,s) \triangleq \frac{1}{t}\sum_{k=0}^{t-1}\left\{\boldsymbol{e}_{|\mathcal{S}|}(s_0)\,\boldsymbol{P}_{\boldsymbol{\theta}}^k\right\}_s.$$

The matrix $\boldsymbol{P}_{\boldsymbol{\theta}}$ is a matrix representation of the state transition probability given the policy $\pi(a|s;\boldsymbol{\theta})$, i.e., $\{\boldsymbol{P}_{\boldsymbol{\theta}}\}_{s,s'} \triangleq \sum_{a\in\mathcal{A}}\pi(a|s;\boldsymbol{\theta})p_{\mathrm{T}}(s'|s,a)$, where $\{\boldsymbol{X}\}_{i,j}$ or $\{\boldsymbol{x}\}_i$ denotes the (i,j)-th or i-th element of a matrix $\boldsymbol{X}$ or a vector $\boldsymbol{x}$, respectively. The vector $\boldsymbol{e}_n(k)$ denotes the n-dimensional column vector whose k-th element is 1 and otherwise zero.

Cesàro mixing time for a bound of estimation bias or variance on finite-time Markov chain

Let us consider a prediction problem of a general class of a linear combination of an arbitrary function $f(s) \in [-C, C]$ and the stationary state distribution $d_{\boldsymbol{\theta}}(s)$,

$$g_{\boldsymbol{\theta}} = \sum_{s\in\mathcal{S}} f(s)d_{\boldsymbol{\theta}}(s), \tag{4}$$

which relates to many sufficient statistics computed in REINFORCE (Williams 1992), GPOMDP (Baxter and Bartlett 2001), and other PGRL methods (Kakade 2002; Peters and Schaal 2008; Morimura et al. 2009). A natural unbiased estimator of $g_{\boldsymbol{\theta}}$ on a finite-time Markov chain of a time length t is

$$\hat{g}_t(s_0) = \frac{1}{t}\sum_{k=0}^{t-1} f(S_k). \tag{5}$$

We show a connection between bias/variance of the estimator and the mixing time in the following propositions as part of the main contribution.

Proposition 1 *The Cesàro mixing time $m(\epsilon, \boldsymbol{\theta})$ of Eq.* (3) *is an upper bound of the number of time steps required to*

[2] The only difference between this standard mixing time and the Cesàro mixing time of Eq. (3) is the definition of $\nu_{\boldsymbol{\theta}}(t, s_0, s)$. In the case of the standard mixing time, $\left\{\boldsymbol{e}_{|\mathcal{S}|}(s_0)\,\boldsymbol{P}_{\boldsymbol{\theta}}^k\right\}_{s'}$ is used for $\nu_{\boldsymbol{\theta}}(t, s_0, s)$ in Eq. (3).

decrease the maximum of the absolute expected bias of the estimator $\hat{g}_t(s_0)$ of Eq. (5) *for $g_{\boldsymbol{\theta}}$ of Eq.* (4)*,*

$$\mathrm{Bias}_{\boldsymbol{\theta}}(t) \triangleq \max_{s_0 \in \mathcal{S}} \left| \mathbb{E}_{\boldsymbol{\theta}}[g_{\boldsymbol{\theta}} - \hat{g}_t(s_0)] \right|$$

below $2C\varepsilon$, i.e., the inequality, $\mathrm{Bias}_{\boldsymbol{\theta}}(m(\varepsilon, \boldsymbol{\theta})) \leq 2C\varepsilon$*, holds.*

Proof. Because of $|f(s)| \leq C$, this bias is bounded as

$$\begin{aligned}\mathrm{Bias}_{\boldsymbol{\theta}}(t) &= \max_{s_0 \in \mathcal{S}} \left| g_{\boldsymbol{\theta}} - \mathbb{E}_{\boldsymbol{\theta}}[\hat{g}_t(s_0)] \right| \\ &= \max_{s_0 \in \mathcal{S}} \left| \sum_{s \in \mathcal{S}} f(s)\, d_{\boldsymbol{\theta}}(s) - \sum_{s \in \mathcal{S}} f(s)\, \nu_{\boldsymbol{\theta}}(t, s_0, s) \right| \\ &\leq \max_{s_0 \in \mathcal{S}} \sum_{s \in \mathcal{S}} |f(s)| \left| d_{\boldsymbol{\theta}}(s) - \nu_{\boldsymbol{\theta}}(t, s_0, s) \right| \\ &\leq C \max_{s_0 \in \mathcal{S}} \sum_{s \in \mathcal{S}} \left| d_{\boldsymbol{\theta}}(s) - \nu_{\boldsymbol{\theta}}(t, s_0, s) \right| \qquad (6)\end{aligned}$$

If t^* is equal to or more than $m(\varepsilon, \boldsymbol{\theta})$, the inequality $\sum_{s \in \mathcal{S}} |d_{\boldsymbol{\theta}}(s) - \nu_{\boldsymbol{\theta}}(t^*, s_0, s)| \leq 2\varepsilon$ holds from the definition of the Cesàro mixing time (Eq. (3)) and thus the proposition, $\mathrm{Bias}_{\boldsymbol{\theta}}(t^*) \leq 2C\varepsilon$, is proved. □

Proposition 2 *The Cesàro mixing time $m(\epsilon, \boldsymbol{\theta})$ of Eq.* (3) *is an upper bound of the number of time steps required to decrease the maximum of the expected (pseudo) variance of the estimator $\hat{g}_t(s_0)$ of Eq.* (5) *for $g_{\boldsymbol{\theta}}$ of Eq.* (4)*,*

$$\mathrm{Var}_{\boldsymbol{\theta}}(t) \triangleq \max_{s_0 \in \mathcal{S}} \mathbb{E}_{\boldsymbol{\theta}} \left[\{ g_{\boldsymbol{\theta}} - \hat{g}_t(s_0) \}^2 \right]$$

below $4C^2\varepsilon$, i.e., the inequality, $\mathrm{Var}_{\boldsymbol{\theta}}(m(\varepsilon, \boldsymbol{\theta})) \leq 4C^2\varepsilon$*, holds.*

Proof. Because of $|g_{\boldsymbol{\theta}} - \hat{g}_t(s_0)| \leq 2C$, this variance is bounded as

$$\mathrm{Var}_{\boldsymbol{\theta}}(t) \leq 2C\, \mathrm{Bias}_{\boldsymbol{\theta}}(t).$$

With Proposition 1, if $t^* \geq m(\varepsilon, \boldsymbol{\theta})$, then $\mathrm{Var}_{\boldsymbol{\theta}}(t^*) \leq 4C^2\varepsilon$ holds. □

Propositions 1 and 2 mean

$$\min_t \left\{ t \geq 0 \,\middle|\, \mathrm{Bias}_{\boldsymbol{\theta}}(t) \leq 2C\varepsilon \right\} \leq m(\varepsilon, \boldsymbol{\theta}),$$
$$\min_t \left\{ t \geq 0 \,\middle|\, \mathrm{Var}_{\boldsymbol{\theta}}(t) \leq 4C^2\varepsilon \right\} \leq m(\varepsilon, \boldsymbol{\theta}).$$

and indicate that the magnitude of the Cesàro mixing time $m(\epsilon, \boldsymbol{\theta})$ can have a great effect on bias and variance for estimating a linear combination of the stationary state distribution on a finite-time MDP. The bias and variance can increase as $m(\epsilon, \boldsymbol{\theta})$ gets larger. Thus, in order to learn a policy in PGRL efficiently, it is required to keep $m(\epsilon, \boldsymbol{\theta})$ low.

Hitting time for a bound of Cesàro mixing time

The hitting time is the first time at which a Markov chain $\mathrm{M}(\boldsymbol{\theta})$ hits a state $s' \in \mathcal{S}$ from an initial state $s \in \mathcal{S}$, which is defined as

$$\tau_{\boldsymbol{\theta}}(s, s') \triangleq \min \{ t \geq 0 \,|\, S_0 = s,\, S_t = s',\, \mathrm{M}(\boldsymbol{\theta}) \}.$$

The expected hitting time is given as

$$h_{\boldsymbol{\theta}}(s, s') \triangleq \mathbb{E}[\tau_{\boldsymbol{\theta}}(s, s')],$$

where $\mathbb{E}[\cdot]$ is the expectation with respect to the distribution of $\tau_{\boldsymbol{\theta}}(s_0, s)$. Also the worst-case (expected) hitting time is

$$h^*(\boldsymbol{\theta}) \triangleq \max_{s, s' \in \mathcal{S}} \{ h_{\boldsymbol{\theta}}(s, s') \}. \qquad (7)$$

We use the following proposition for a bound of Cesàro mixing time with the hitting time.

Proposition 3 (Levin, Peres, and Wilmer 2008) *The Cesàro mixing time is bounded by using the worst-case hitting time,*

$$m(\varepsilon, \boldsymbol{\theta}) \leq \frac{1}{\varepsilon} h^*(\boldsymbol{\theta}) + 1.$$

4 Mixing-time regularized policy gradient

We derive a framework of policy gradient with mixing-time-bound regularization. The results in the previous section indicate that, in order to compute some statistics for the policy update efficiently, a learner should keep magnitude of the Cesàro mixing time low[3]. Thus we want to directly control the Cesàro mixing time during in learning process. However, according to the best of our knowledge, there is no practical method in RL to estimate and control it. On the other hand, as shown in Proposition 3, the worst-case hitting time is an upper bound of the Cesàro mixing time, and its derivative with respect to the policy parameter can be easily estimated as described in Section 5. Accordingly, we consider a natural approach for keeping the Cesàro mixing time low, in which the derivative of the worst-case hitting time is used for the regularization term $l(\boldsymbol{\theta})$ in Eq. (2), such as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{G}_{\boldsymbol{\theta}_t} \boldsymbol{\nabla} \eta(\boldsymbol{\theta}_t) - \lambda_t \boldsymbol{\nabla} h^*(\boldsymbol{\theta}_t), \qquad (8)$$

or $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{G}_{\boldsymbol{\theta}_t} \boldsymbol{\nabla} \eta(\boldsymbol{\theta}_t) - \lambda_t \boldsymbol{G}_{\boldsymbol{\theta}_t} \boldsymbol{\nabla} h^*(\boldsymbol{\theta}_t)$. The update of the policy tries to increase the average reward and also to decrease the worst-case hitting time or an upper bound of the Cesàro mixing time. However, the intended objective of the proposed approach is not to decrease excessively long Cesàro mixing time but rather tries to prevent the Cesàro mixing time from increasing. This suppression of the Cesàro mixing time can be understood with an analogy to the natural gradient (Amari 1998) or natural policy gradient (Kakade 2002), which tries to prevent the learning parameter from falling into the region that causes learning plateau. One cannot expect the natural gradient to perform well in the regions with heavy learning plateau.

In practice, it is important to balance the effects of the last two terms in Eq. (8). Conceivably, the mixing time might be huge or diverge with the optimal policy. Thus it would be needed to decrease the regularization parameter depending on time t, such as $\lambda_t := x/(y + t^2)$, where $x > 0$ and $y \geq 0$ are constants. We give a proposition for setting of α_t and λ_t to guarantee a convergence to a local optimum.

[3] Note that it is also known that keeping magnitude of a mixing time low encourages the exploration (Kearns and Singh 2002; Kakade 2003).

Proposition 4 *Let a Lipschitz continuity condition on $\boldsymbol{\nabla}\eta$ hold, i.e., there is some constant $L > 0$ satisfying $\|\boldsymbol{\nabla}\eta(\boldsymbol{\theta}) - \boldsymbol{\nabla}\eta(\boldsymbol{\theta}')\| \leq L\,\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$, ${}^{\forall}\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, where $\|\cdot\|$ denotes the Euclidean norm, and $\boldsymbol{\nabla}h^*$ be uniformly bounded, i.e., there is a constant M satisfying $\|\boldsymbol{\nabla}h^*(\boldsymbol{\theta})\| < M$, ${}^{\forall}\boldsymbol{\theta} \in \mathbb{R}^d$. Assume that the learning rate $\alpha_t \geq 0$ and the regularization parameter $\lambda_t \geq 0$ satisfy*

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \qquad \sum_{t=0}^{\infty} \alpha_t^2 < \infty,$$

and, for some positive constant c,

$$\lambda_t \leq c\alpha_t^2,$$

respectively. Then, with the update manner of Eq. (8)*, the $\eta(\boldsymbol{\theta}_t)$ converges to a finite value and $\lim_{t\to\infty} \boldsymbol{\nabla}\eta(\boldsymbol{\theta}_t) = 0$. Furthermore, every limit point of $\boldsymbol{\theta}_t$ is a stationary point of η.*

Proof. See the associated technical report (Morimura, Osogami, and Shirai 2014).

Alternatively, the regularization term can be decreased in consideration of the achieved objective value. For example, the following heuristic for the update can be used,

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha\boldsymbol{\nabla}\eta(\boldsymbol{\theta}) - \lambda\max(0, \eta^* - \hat{\eta}(\boldsymbol{\theta}))\boldsymbol{\nabla}h^*(\boldsymbol{\theta}), \quad (9)$$

where η^* is a targeted average reward set manually, which does not have to be the maximum average reward. Also, $\hat{\eta}(\boldsymbol{\theta})$ is an estimator of the current average reward, which will be simply updated at time t as

$$\hat{\eta}(\boldsymbol{\theta}) := (1-\alpha)\hat{\eta}(\boldsymbol{\theta}) + \alpha R_{t+1},$$

where $R_{t+1} = R(S_t, A_t, S_{t+1})$ is a random variable of reward. Hereinafter, the update rule of the mixing-time regularized PG with Eq. (8) will be called as Option 1, and also that with Eq. (9) as Option 2.

For the implementation, we need to estimate $\boldsymbol{\nabla}\eta(\boldsymbol{\theta})$ and $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$. Since there are several existing methods for estimating $\boldsymbol{\nabla}\eta(\boldsymbol{\theta})$, it remains to provide a method for estimating $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$, which is described in the following section.

5 Estimation of Mixing-time-Bound Derivative $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$

We derive an estimation method for the derivative of the worst-case hitting time, $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$, as the mixing-time regularization term. Since a direct estimation of $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$ is difficult due to non-linearity of $h^*(\boldsymbol{\theta})$ resulting from its maximizing operation (see Eq. (7)), we have the following stepwise approach. First the expected hitting time and its derivative are estimated as $\hat{h}(s,s')$ and $\widehat{\boldsymbol{\nabla}}h(s,s')$, respectively. Second the state pair (s^*, s'^*) that maximizes $\hat{h}(s,s')$ is searched. Then we compute $\widehat{\boldsymbol{\nabla}}h(s^*, s'^*)$ as an estimate of $\boldsymbol{\nabla}h^*(\boldsymbol{\theta})$. An estimation method for the expected hitting time or its derivative is described in subsequent subsections.

Note that we derive those estimation methods "under a fixed policy," but this is standard in the literature of PGRL. In particular, there is a policy evaluation step that evaluates the performance of a current (fixed) policy. Then the policy is updated based on the results of evaluation. A concrete algorithm for the mixing-time regularized PG is shown in Algorithm 1.

Estimation of hitting time

The hitting time estimation problem of $h_{\boldsymbol{\theta}}(s,s')$ can be reduced to that of the value function (Sutton and Barto 1998) on a finite-time-horizon MDP with an absorbing state s', where the reward is always 1. This observation leads to the following recursive formula, which we call the hitting-time Bellman equation,

$$h_{\boldsymbol{\theta}}(s,s') = \begin{cases} 0 & \text{if } s = s', \\ 1 + \mathbb{E}_{\boldsymbol{\theta}}\big[h_{\boldsymbol{\theta}}(S_{t+1}, s') \,|\, S_t = s\big] & \text{otherwise.} \end{cases}$$

Let us consider an estimator of the expected hitting time, $\hat{h} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$. From the hitting-time Bellman equation, $\hat{h}$ at time t can be updated on the basis of the temporal-difference learning of the value function (Sutton and Barto 1998): if $s_t \neq s_{t+1}$,

$$\hat{h}(s_t, i) := \hat{h}(s_t, i) + \alpha_t \delta_t^{(h)}(i), \;\; i \in \{s \,|\, s \neq s_t, s \in \mathcal{S}\}, \quad (10)$$

where $\delta^{(h)}$ is a temporal-difference function for the hitting time,

$$\delta_t^{(h)}(i) \triangleq 1 + \hat{h}(s_{t+1}, i) - \hat{h}(s_t, i).$$

Throughout, we keep $\hat{h}(s,s) = 0, \forall s \in \mathcal{S}$.

We also derive an alternative approach for estimating $h_{\boldsymbol{\theta}}$ on the basis of the least-square temporal-difference learning (Bradtke and Barto 1996; Boyan 2002). The sufficient statistics for this estimation are defined as $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{S}|}$. The update rule at time t is given as, if $s_t \neq s_{t+1}$,

$$\begin{aligned} \boldsymbol{A} &:= \beta\boldsymbol{A} + \boldsymbol{e}_{|\mathcal{S}|}(s_t)\{\boldsymbol{e}_{|\mathcal{S}|}(s_t) - \boldsymbol{e}_{|\mathcal{S}|}(s_{t+1})\}^\top, \\ \boldsymbol{b} &:= \beta\boldsymbol{b} + \boldsymbol{e}_{|\mathcal{S}|}(s_t), \end{aligned}$$

where $\beta \in [0,1]$ is a forgetting rate. The estimated value with the sufficient statistics is given as

$$\hat{h}(i,j) := \begin{cases} \big\{\boldsymbol{A}_{/j}^{-1}\boldsymbol{b}_{/j}\big\}_i & \text{if } i < j, \\ 0, & \text{if } i = j, \\ \big\{\boldsymbol{A}_{/j}^{-1}\boldsymbol{b}_{/j}\big\}_{i-1}, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{X}_{/y}$ denotes a partitioned matrix in which the y-th column and row are removed from a matrix $\boldsymbol{X}$, and $\boldsymbol{x}_{/y}$ a partitioned vector where the y-th element is removed from a vector $\boldsymbol{x}$. It is empirically shown that the efficiency of the estimation with the least-square temporal-difference learning is higher than that with temporal-difference learning. However, the computational cost with the least-square temporal-difference is also much higher and will be $O(|\mathcal{S}|^3)$ due to the matrix inversion at each time step in our scenario.

Note that the eligibility-trace technique (Sutton and Barto 1998) will be applied to those estimation analogously to TD(λ) in (Sutton and Barto 1998) or LSTD(λ) (Boyan 2002). However, we skip it due to lack of space.

Estimation of hitting-time derivative

We consider an estimator of the hitting-time derivative with respect to the policy parameter $\boldsymbol{\theta}$, $\widehat{\boldsymbol{\nabla}}h : \mathcal{S}\times\mathcal{S} \to \mathbb{R}^d$. By taking partial differentiation of the hitting-time Bellman equation of Eq. (10), the following recursive formula, called the

Algorithm 1: An implementation of the mixing-time regularized policy gradient reinforcement learning

A mixing time bound regularized PGRL algorithm with temporal-difference learning (Option 1)

```
**Given:**
  • a policy $\pi(a|s;\boldsymbol{\theta})$ with an policy parameter $\boldsymbol{\theta} \in \mathbb{R}^d$
  • hyper-parameters: $\alpha_t^{(h)}$, $\alpha_t^{(\nabla)}$, $\alpha_t^{(\pi)}$, $\lambda_t \geq 0$
**Set:**
  • an initial hitting time function:
    $\hat{h} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ s.t. $\hat{h}(s,s) = 0, ^\forall s \in \mathcal{S}$
  • an initial hitting time derivative function:
    $\widehat{\boldsymbol{\nabla}} h : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^d$ s.t. $\widehat{\boldsymbol{\nabla}} h(s,s) = \boldsymbol{0}, ^\forall s \in \mathcal{S}$
  • an initial state: $s_0 \in \{s, \ldots, |\mathcal{S}|\}$ $(\sim \Pr(s_0))$
**For** $t = 0$ **to** $T-1$ **do**
  (*Interaction with environment*)
  • chose and execute action $a_t \sim \pi(a|s;\boldsymbol{\theta})$
  • observe following state $s_{t+1}$ and reward $r_{t+1}$
  (*Update* $\hat{h}$ *and* $\widehat{\boldsymbol{\nabla}} h$ *for all* $i \in \{s \,|\, s \neq s_t, s \in \mathcal{S}\}$)
  • $\hat{h}(s_t,i) := \hat{h}(s_t,i) + \alpha_t^{(h)}\Big(1 + \hat{h}(s_{t+1},i) - \hat{h}(s_t,i)\Big)$
  • $\widehat{\boldsymbol{\nabla}} h(s_t,i) := \widehat{\boldsymbol{\nabla}} h(s_t,i) + \alpha_t^{(\nabla)}\Big\{ - \widehat{\boldsymbol{\nabla}} h(s_t,i)$
        $+ \hat{h}(s_{t+1},i)\boldsymbol{\nabla}\log\pi(a_t|s_t;\boldsymbol{\theta}) + \widehat{\boldsymbol{\nabla}} h(s_{t+1},i)\Big\}$
  (*Compute the worst-case hitting time estimate* $\widehat{\boldsymbol{\nabla}} h^*$
        *as the mixing-time regularization term*)
  • find $(s^*, s'^*) := \arg\max_{s,s' \in \mathcal{S}} \hat{h}(s,s')$
  • compute $\widehat{\boldsymbol{\nabla}} h^* := \widehat{\boldsymbol{\nabla}} h(s^*, s'^*)$
  (*Compute* $\Delta\boldsymbol{\theta}$ *as update direction for the parameter* $\boldsymbol{\theta}$)
  • do an arbitrary PG algorithm, such as GPOMDP
    (Baxter and Bartlett 2001) or NPG (Kakade 2002)
  (*Update the policy parameter* $\boldsymbol{\theta}$)
  • $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t^{(\pi)}\Delta\boldsymbol{\theta} - \lambda_t \widehat{\boldsymbol{\nabla}} h^*$
**End**
**Return:** the policy $\pi(a|s;\boldsymbol{\theta})$.
```

hitting-time-derivative Bellman equation, is derived,

$$\boldsymbol{\nabla} h_{\boldsymbol{\theta}}(s,s') = \begin{cases} \boldsymbol{0}, & \text{if } s = s', \\ \mathbb{E}_{\boldsymbol{\theta}}\big[h_{\boldsymbol{\theta}}(S_{t+1},s')\boldsymbol{\nabla}\log\pi(A_t|s;\boldsymbol{\theta}) \\ \qquad + \boldsymbol{\nabla} h_{\boldsymbol{\theta}}(S_{t+1},s') \mid S_t = s\big], & \text{otherwise.} \end{cases} \tag{11}$$

The lower part comes from the derivative of

$$\mathbb{E}_{\boldsymbol{\theta}}[h_{\boldsymbol{\theta}}(S_{t+1},s')|S_t = s] = \sum_{a_t, s_{t+1}} h_{\boldsymbol{\theta}}(s_{t+1},s') p_{\mathrm{T}}(s_{t+1}|s,a_t)\pi(a_t|s_t;\boldsymbol{\theta}).$$

From the Bellman equation of Eq. (11), an update rule of $\widehat{\boldsymbol{\nabla}} h$, on the basis of the temporal-difference learning (Sutton and Barto 1998), is given at time t as, if $s_t \neq s_{t+1}$,

$$\widehat{\boldsymbol{\nabla}} h(s_t,i) := \widehat{\boldsymbol{\nabla}} h(s_t,i) + \alpha_t \boldsymbol{\delta}_t^{(\nabla)}(i), \qquad i \in \{s \,|\, s \neq s_t, s \in \mathcal{S}\},$$

where $\boldsymbol{\delta}^{(\nabla)} \in \mathbb{R}^d$ is a temporal-difference-vector function for the hitting time derivative,

$$\boldsymbol{\delta}_t^{(\nabla)}(i) \triangleq \hat{h}(s_{t+1},i)\boldsymbol{\nabla}\log\pi(a_t|s_t;\boldsymbol{\theta}) + \widehat{\boldsymbol{\nabla}} h(s_{t+1},i) - \widehat{\boldsymbol{\nabla}} h(s_t,i).$$

We also give a least-squares based estimation of $\boldsymbol{\nabla} h_{\boldsymbol{\theta}}$, where the sufficient statistics are defined as $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and $\boldsymbol{C}^{(i)} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, $^\forall i \in \{1, \ldots, |S|\}$. These are updated at time t as, if $s_t \neq s_{t+1}$,

$$\boldsymbol{A} := \beta\boldsymbol{A} + \boldsymbol{e}_{|\mathcal{S}|}(s_t)\{\boldsymbol{e}_{|\mathcal{S}|}(s_t) - \boldsymbol{e}_{|\mathcal{S}|}(s_{t+1})\}^\top,$$
$$\boldsymbol{C} := \beta\boldsymbol{C} + \boldsymbol{\nabla}\log\pi(a_{t-1}|s_{t-1};\boldsymbol{\theta})\hat{\boldsymbol{h}}(s_t,:)^\top,$$

where $\beta \in [0,1]$ is a forgetting rate and $\hat{\boldsymbol{h}}(s,:)$ denotes $[\hat{h}(s,1), \ldots, \hat{h}(s,|\mathcal{S}|)]^\top$.

The estimated value with the sufficient statistics is given as

$$\widehat{\boldsymbol{\nabla}} h(i,j) := \begin{cases} \big\{\boldsymbol{A}_{/j}^{-1}\boldsymbol{C}_{/j}\big\}_{i,:} & \text{if } i < j, \\ \boldsymbol{0} & \text{if } i = j, \\ \big\{\boldsymbol{A}_{/j}^{-1}\boldsymbol{C}_{/j}\big\}_{i-1,:} & \text{otherwise,} \end{cases}$$

where $\{\boldsymbol{X}\}_{i,:}$ is the i-th column of a matrix $\boldsymbol{X}$.

6 Experiments

We look into the effect of the mixing-time-bound regularization through numerical experiments. To simply evaluate this effect, all of the applied methods here use a simple, standard policy gradient method in (Baxter and Bartlett 2001) to estimate $\boldsymbol{\nabla}\eta$. In particular, the baseline methods compared with the proposed methods are the GPOMDPs with the ordinary policy gradient (Baxter and Bartlett 2001) and with the natural policy gradient (Kakade 2002). In the case of our proposed methods of Option 1 and Option 2, GPOMDP for $\boldsymbol{\nabla}\eta$ and the temporal-difference learning for $h_{\boldsymbol{\theta}}$ and $\boldsymbol{\nabla} h_{\boldsymbol{\theta}}$ are used.

The task is a simple two-state MDP in (Kakade 2002), where each state $s \in \{1,2\}$ has self- and cross-transition actions $\mathcal{A} = \{\text{self}, \text{cross}\}$ and each state transition is deterministic. The reward function is set as $R(1,\text{self},1) = 1$, $R(2,\text{self},2) = 2$, and $R(i,\text{cross},j) = 0$ for every feasible pair of (i,j). The policy with $\boldsymbol{\theta} \in \mathbb{R}^2$ is represented by the sigmoidal function: $\pi(\text{self}|s;\boldsymbol{\theta}) = 1/(1+\exp(-\theta_s))$ and $\pi(\text{cross}|s;\boldsymbol{\theta}) = 1 - \pi(\text{self}|s;\boldsymbol{\theta})$. The hyper-parameters of those methods were tuned. The targeted average reward η^*, which is a hyper-parameter in the proposed method of Option 2, was set as $\eta^* := 0.75\max_{\boldsymbol{\theta}\in\mathbb{R}^2}\eta(\boldsymbol{\theta}) = 1.5$.[4] Figure 1 shows performance comparison when the initial policy parameter $\boldsymbol{\theta}_0 = [2.2, -2.2]^\top$, which corresponds to $\pi(\text{self}|1;\boldsymbol{\theta}_0) = \pi(\text{cross}|2;\boldsymbol{\theta}_0) \simeq 0.9$ and severe *plateau* occurs with the baseline methods during the first 10^4 steps due to a large magnitude of the mixing time. From this result, we confirm that the proposed approach of the mixing-time regularization can improve the performance of the conventional PGRL methods.

[4] Although not shown in the paper, we also tested other values for η^*. The results was that the performances differed only little when η^* is in $[0.6\max_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}),\ 0.9\max_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})]$.

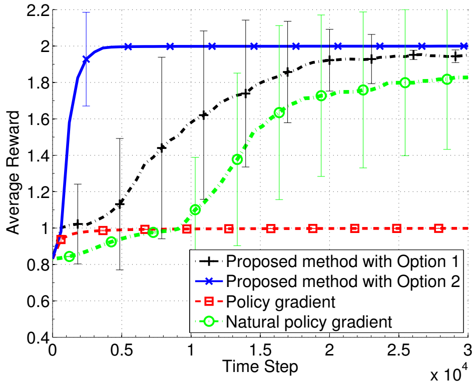


Figure 1: Performance comparison at the initial policy parameter $\boldsymbol{\theta}_0 = [2.2, -2.2]^\top$ on 2-state MDP

Since it is known that performance of the PGRL methods are severely affected by the initialization of the policy, we also looked into the dependence on the initial policy parameter in Figure 2. The results indicate that our approach can largely soften the dependence. The comparison between the proposed methods of Option 1 and 2 indicates that, if a good targeted average reward η^* is set, Option 2 will be a good heuristic.

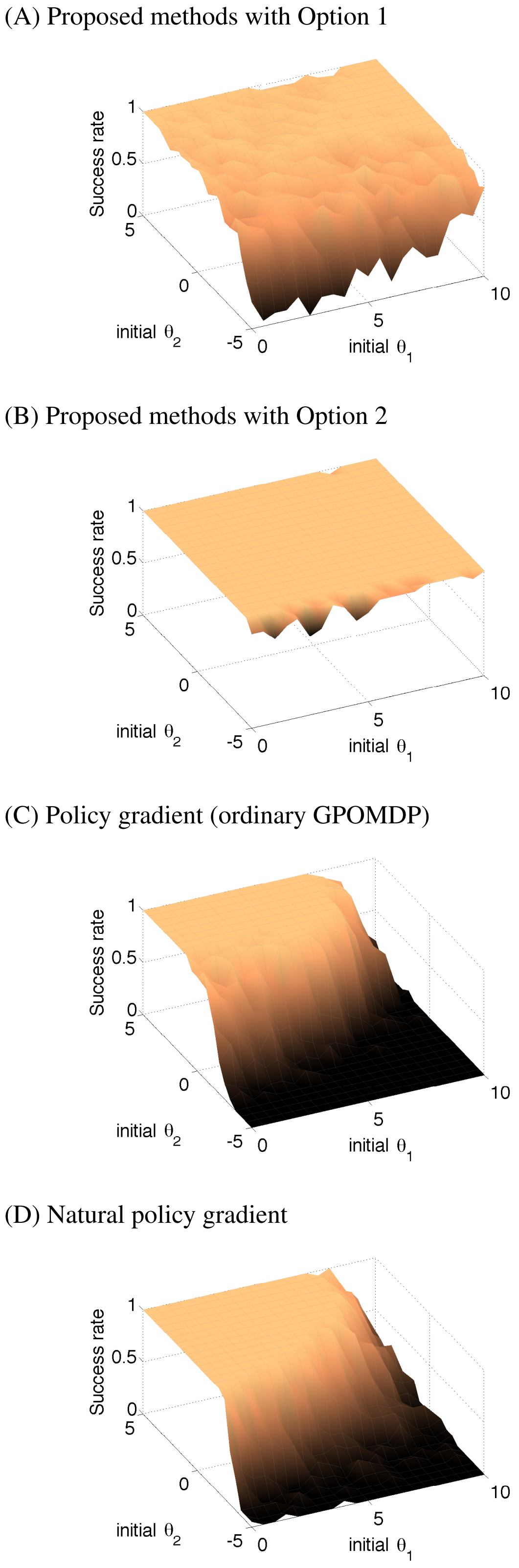
7 Conclusion

In this paper, we proved in Propositions 1 and 2 that the Cesàro mixing time is an upper bound of bias and variance of an estimate in a finite-time Markov chain. Then we proposed an approach for regularizing the Cesàro mixing time via the hitting time on policy gradient reinforcement learning, which keeps the Markov chain compact and improves the learning efficiency. That is to say, the proposed methods for suppressing the hitting time can prevent the mixing time from heavily increasing and thus can avoid large estimation errors of policy-gradient and hitting-time. A sufficient condition of a convergence for the proposed approach was also presented in Proposition 4. For the implementation of this approach, several methods for estimating the hitting time and its derivative were presented based on the temporal-difference learning or the least-squares temporal-difference learning. Finally we demonstrated the effectiveness of the proposed methods through numerical experiments.

Further theoretical analysis, especially for the case that the policy-gradient and regularization terms are estimated with samples rather than known exactly, will be necessary to more deeply understand the properties and efficiency, specifically in term of their convergence and sample complexities. For the scalability in the number of states, while the paper considers all of the state-pairs in the calculation of the hitting time, one can focus only on particular state-pairs, if one knows which states are desirable, undesirable, etc. There are several interesting directions on theoretical analysis as well as algorithmic studies. Also, empirical studies with some more challenging domains is important for our future work.

Acknowledgments

A part of this research was supported by JST, CREST.


(A) Proposed methods with Option 1

(B) Proposed methods with Option 2

(C) Policy gradient (ordinary GPOMDP)

(D) Natural policy gradient

Figure 2: Success rates for achieving a targeted average reward 1.9 $(= 0.95 \max_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}))$ in 10^4 time steps by using the various initial policy parameters $\boldsymbol{\theta}_0$.

References

Amari, S. 1998. Natural gradient works efficiently in learning. *Neural Computation* 10(2):251–276.

Bartlett, P. L., and Baxter, J. 2000. Estimation and approximation bounds for gradient-based reinforcement learning. In *Annual Conference on Computational Learning Theory*, 133–141.

Baxter, J., and Bartlett, P. L. 2000. Reinforcement learning in POMDP's via direct gradient ascent. In *International Conference on Machine Learning*, 41–48.

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

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

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

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

Gullapalli, V. 1990. A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks* 3(6):671–692.

Kakade, S. 2002. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Kakade, S. 2003. *On the Sample Complexity of Reinforcement Learning*. Ph.D. thesis, University College London.

Kearns, M., and Singh, S. 2002. Near-optimal reinforcement learning in polynomial time. *Machine Learning* 49(2-3):209–232.

Levin, D.; Peres, Y.; and Wilmer, E. 2008. *Markov Chains and Mixing Times*. American Mathematical Society.

Lovász, L., and Winkler, P. 1998. Mixing times. In Aldous, D., and Propp, J., eds., *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 85–133.

Morimura, T.; Uchibe, E.; Yoshimoto, J.; and Doya, K. 2008. A new natural policy gradient by stationary distribution metric. 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. In *Advances in Neural Information Processing Systems*, volume 22.

Morimura, T.; Osogami, T.; and Shirai, T. 2014. Mixing-time regularized policy gradient. In *Technical Report*. IBM Research, RT0961.

Peters, J., and Schaal, S. 2008. Natural actor-critic. *Neurocomputing* 71(7-9):1180–1190.

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

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8:229–256.