JMLR: Workshop and Conference Proceedings 13: 285-298
2nd Asian Conference on Machine Learning (ACML2010), Tokyo, Japan, Nov. 8–10, 2010.

Adaptive Step-size Policy Gradients with Average Reward Metric

Takamitsu Matsubara TAKAM-M@IS.NAIST.JP
Graduate School of Information Science
Nara Institute of Science and Technology, Nara, Japan,
Department of Brain Robot Interface
ATR Computational Neuroscience Laboratories, Kyoto, Japan

Tetsuro Morimura TETSURO@JP.IBM.COM
IBM Research - Tokyo, Kanagawa, Japan

Jun Morimoto XMORIMO@ATR.JP
Department of Brain Robot Interface
ATR Computational Neuroscience Laboratories, Kyoto, Japan

Editor: Masashi Sugiyama and Qiang Yang

Abstract

In this paper, we propose a novel adaptive step-size approach for policy gradient reinforcement learning. A new metric is defined for policy gradients that measures the effect of changes on average reward with respect to the policy parameters. Since the metric directly measures the effects on the average reward, the resulting policy gradient learning employs an adaptive step-size strategy that can effectively avoid falling into a stagnant phase from the complex structure of the average reward function with respect to the policy parameters. Two algorithms are derived with the metric as variants of ordinary and natural policy gradients. Their properties are compared with previously proposed policy gradients through numerical experiments with simple, but non-trivial, 3-state Markov Decision Processes (MDPs). We also show performance improvements over previous methods in on-line learning with more challenging 20-state MDPs.

Keywords: Policy Gradients, Natural Policy Gradients, Average Reward Metric

1. Introduction

Policy gradient reinforcement learning (Williams, 1992; Kimura and Kobayashi, 1998; Sutton et al., 2000; Konda and Tsitsiklis, 2003) has received much attention for several applications (Tedrake et al., 2004; Matsubara et al., 2006; Richter et al., 2006). This approach has a stochastic policy expressed by a function with its own parameters called the *policy parameters*. The policy parameters are updated to increase some performance criteria, (e.g., average reward) towards locally optimal policies using gradient ascent. Compared with global optimization with value-function-based methods (e.g., Q-learning (Watkins and Dayan, 1992) and Sarsa (Sutton and Barto, 1998)), policy gradients have several advantages such as their applicability to Partially Observable Markov Decision Processes (POMDPs) and the convergence proof for using function approximators (Baxter and Bartlett, 2001b; Sutton et al., 2000; Konda and Tsitsiklis, 2003).

One recent theoretical advance is a natural gradient approach for policy gradients called the *natural policy gradients*. This approach, originally proposed in (Kakade, 2002), was inspired by Amari's natural gradient algorithms in supervised learning contexts (Amari, 1998). Following this pioneering work, various algorithms have been proposed (Bagnell and Schneider, 2003; Peters et al., 2003; Richter et al., 2006; Ghavamzadeh and Engel, 2006). The natural policy gradients involve covariant policy searches with some metric measuring the effect on an action probability distribution with respect to the policy parameters (Kakade, 2002). A number of studies have empirically demonstrated that natural policy gradients significantly outperformed ordinary policy gradients in terms of their convergence rates as in (Kakade, 2002; Bagnell and Schneider, 2003; Peters et al., 2003; Richter et al., 2006; Ghavamzadeh and Engel, 2006).

However, the metric does not measure the effect on the performance criteria (e.g., average reward). This means that the updates of the policy parameters, even with the natural gradient, may result in an extremely small (or undetectable) improvement. Therefore, natural policy gradients still lead to a learning process in a stagnant phase from a complex structure of the average reward function because the metric does not reflect the average reward function. This concern motivates us to ask whether we can find a principled metric that takes the average reward in a Markov Decision Process (MDP) into account to more directly control the progress of the learning process.

In this paper, we propose a novel adaptive step-size approach for policy gradient reinforcement learning. A new metric is defined for policy gradients that measures the effect of changes on the average reward with respect to the policy parameters. Since the metric directly measures the effect on the average reward, the resulting policy gradient learning employs an adaptive step-size strategy that can effectively avoid falling into a stagnant phase from the complex structure of the average reward function. Two algorithms are derived with the metric as variants of ordinary and natural policy gradients. Their properties are compared with previously proposed policy gradients by numerical experiments with simple, but non-trivial, 3-state MDPs. We demonstrate performance improvements over previous methods in on-line learning with more challenging 20-state MDPs.

The rest of this paper is organized as follows. Section 2 briefly reviews policy gradients and natural policy gradients. Section 3 shows our adaptive step-size policy gradient approach for the metric of average reward. Section 4 presents algorithms for policy gradients using the metric. Section 5 demonstrates its effectiveness compared to previously proposed policy gradients with 3-state MDPs and more challenging 20-state MDPs. We discuss the conclusions of this paper and future work in Section 6.

2. Policy Gradient to Natural Policy Gradient

In this section, policy gradients and natural policy gradients are briefly reviewed as preliminaries for the next sections.

2.1 Policy gradient approach

We explain policy gradient reinforcement learning for finite-state Markov Decision Processes (MDPs) defined by a system consisting of a finite state set $\mathcal{S}$ and a finite action set $\mathcal{A}$. The state transitions are governed by a state transition probability distribution $p(s'|s,a)$, where

s and $s' \in \mathcal{S}$, $a \in \mathcal{A}$. A stochastic policy that controls the MDP is defined by $\pi(s, a; \boldsymbol{\theta})$ $(= p(a|s; \boldsymbol{\theta}))$ and a reward function is defined as $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The objective is to acquire the (locally) optimal policy $\pi(s, a; \boldsymbol{\theta}^*)$ to maximize the average reward. Assuming that all of the policies $\pi(s, a; \boldsymbol{\theta})$ in the parameter space are *ergodic (irreducible and aperiodic)* (i.e., a unique steady state distribution $d^\pi(s)$ is well-defined for each policy (e.g., (Bertsekas, 1995)), the average reward function with respect to the policy parameter is defined as

$$\begin{aligned} \eta(\boldsymbol{\theta}) &= \lim_{n\to\infty} \frac{1}{n} E\{r_1 + r_2 + \cdots + r_n | \pi_{\boldsymbol{\theta}}\} \\ &= \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s, a; \boldsymbol{\theta}) r(s, a), \end{aligned} \tag{1}$$

where r_n indicates the immediate reward $r(s, a) \in \mathbb{R}$ at time step n. The policy gradients estimate gradient $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ with respect to the policy parameter of the current policy and update the parameter as $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ (Sutton et al., 2000; Konda and Tsitsiklis, 2003; Baxter and Bartlett, 2001b; Kimura and Kobayashi, 1998), where α is a sufficiently small step-size parameter and the notation := denotes the right-to-left substitution. From Eq. (1), the policy gradient $\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ is obtained as

$$\begin{aligned} \nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = &\sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s, a; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln \pi(s, a; \boldsymbol{\theta}) Q_\gamma^\pi(s, a) \\ &+ (1 - \gamma) \sum_{s\in\mathcal{S}} d(s) \nabla_{\boldsymbol{\theta}} \ln d^\pi(s) V_\gamma^\pi, \end{aligned} \tag{2}$$

where

$$\begin{cases} Q_\gamma^\pi(s, a) \equiv \lim_{K\to\infty} E\left\{\sum_{k=1}^K \gamma^k r_k | s, a\right\} \\ V_\gamma^\pi(s) \quad\ \equiv \lim_{K\to\infty} E\left\{\sum_{k=1}^K \gamma^k r_k | s\right\}, \end{cases}$$

$\gamma \in [0, 1)$ is a time-discounting factor, $V_\gamma^\pi(s)$ is a state value function, $Q_\gamma^\pi(s, a)$ is a state-action value function (Baxter and Bartlett, 2001b). In general for reinforcement learning tasks, the gradient cannot be obtained analytically because the steady-state distribution $d^\pi(s)$ and value functions $V_\gamma^\pi(s)$ and $Q_\gamma^\pi(s, a)$ are unknown. Therefore, the gradient must be estimated from the empirically sampled data.

In this paper, we refer to a policy gradient with this approach as an Ordinary Policy Gradient method (OPG).

2.2 Natural policy gradient approach

(Kakade, 2002) introduced the natural gradient learning approach in policy gradient reinforcement learning called a Natural Policy Gradient method (NPG). In (Kakade, 2002), a Riemannian metric is defined to measure the effects of changes on an action probability distribution $\pi(s, a; \boldsymbol{\theta})$ by a small incremental vector $\Delta\boldsymbol{\theta}$ in the current policy $\pi(s, a; \boldsymbol{\theta})$ as

$$D_{p(s,a)}[\boldsymbol{\theta} \| \boldsymbol{\theta} + \Delta\boldsymbol{\theta}] \equiv \Delta\boldsymbol{\theta}^T \mathbf{F}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}, \tag{3}$$

where

$$\mathbf{F}(\boldsymbol{\theta}) \equiv \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s, a; \boldsymbol{\theta}) \left[\nabla_{\boldsymbol{\theta}} \ln \pi(a, s; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln \pi(a, s; \boldsymbol{\theta})^T\right]. \tag{4}$$

In NPG, the update direction of the policy parameter is $\boldsymbol{F}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$ with the constraint $D_{p(s,a)}[\boldsymbol{\theta}\|\boldsymbol{\theta}+\Delta\boldsymbol{\theta}] = \epsilon^2$. The update schema of policy parameters with Kakade's NPG (Kakade, 2002) is

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha\boldsymbol{F}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}),$$

where α is a sufficiently small step-size parameter.

Kakade implemented this method in simulation studies and empirically showed its effectiveness, i.e, faster convergence to locally optimal policies than OPG. Further studies as in (Bagnell and Schneider, 2003; Peters et al., 2003) found that Kakade's work was based on the probability manifold of the path distribution $p(\tau;\boldsymbol{\theta})$ (also called the trajectory or history denoted by $\tau = [s_0, a_0, s_1, a_1, \cdots, s_H, a_H]$, where H is the horizon of the history). They also demonstrated that NPG often outperforms OPG in practice.

One of the main advantages of using NPG comes from the direction of the gradient. A natural gradient will be rotated in a direction that avoids serious plateau phenomena on a curved manifold in policy parameter space during learning (Amari, 1998). Another characteristic comes from the adaptive step-size parameter consistent with the Riemannian metric constraint (Amari, 1998; Peters and Schaal, 2008). This constraint makes the NPG covariant policy search (the performance progress of the learning process) no longer depends on the parametrization of the policy (in the limit of infinitesimally small step-sizes). However, the step-size parameter will be adjusted so that every policy improvement yields the same effect in the action probability distribution (Kakade, 2002). Therefore, its effect on the average reward can be significantly different so that the updates of the policy parameters may result in an extremely small improvement. This may lead the learning process to a stagnant phase caused by the complex structure of the average reward function.

These considerations motivate us to ask whether we can define a principled metric that considers the average reward in the MDP to directly control the learning process. In the next section, we derive a novel policy gradient approach for this purpose.

3. A Policy Gradient Approach with Average Reward Metric

In this section, we present a policy gradient approach with a metric for the average reward. The basic method of the learning is discussed in Section 3.1. In Section 3.2, we derive a Riemannian metric that measures the effect on the average reward of the policy improvement, and its properties are described in Section 3.3.

3.1 Average reward metric policy gradient method

Let us consider a metric for policy gradient algorithms that measures the effect of a change in policy parameter $\boldsymbol{\theta}$ on the average reward. Assuming that the effect is measured by a Riemannian metric similar to a NPG, we can define a metric D_η as

$$D_\eta[\boldsymbol{\theta}\|\boldsymbol{\theta}+\Delta\boldsymbol{\theta}] \equiv \Delta\boldsymbol{\theta}^T\boldsymbol{R}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}, \tag{5}$$

where $\Delta\boldsymbol{\theta}$ is a small incremental vector and $\boldsymbol{R}(\boldsymbol{\theta}) \in \mathbb{R}^{d\times d}$ is a Riemannian metric matrix that defines the properties of the metric. We call D_η the average reward metric and $\boldsymbol{R}(\boldsymbol{\theta})$ the

average reward metric matrix. The average reward metric D_η measures a kind of distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \boldsymbol{\Delta\theta}$ in the average reward.

We assume that $\boldsymbol{R}(\boldsymbol{\theta})$ is a positive definite matrix. Following the same approach as (Amari, 1998), the steepest ascent learning scheme under the constraint $D_\eta = \epsilon^2$ can be derived as

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \epsilon\alpha(\boldsymbol{\theta})\mathbf{R}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}), \tag{6}$$

where

$$\alpha(\boldsymbol{\theta}) \equiv \frac{1}{\sqrt{\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})^T\mathbf{R}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})}}.$$

We call this learning scheme the Average reward metric Policy Gradient method (APG).

Since APG learning has a constraint that the average reward metric $D_\eta[\boldsymbol{\theta}\|\boldsymbol{\theta} + \boldsymbol{\Delta\theta}]$ is constant, the derivative of the change in the average reward at each policy improvement with this gradient is also constant. Therefore, each policy improvement only causes a fixed size change in the average reward, i.e., ϵ^2 for the constraint $D_\eta = \epsilon^2$. That is, the learning process can be directly controlled by changing ϵ which can also be used to avoid falling into a stagnant phase caused by the complex structure of the average reward function in MDPs.

3.2 Average reward metric in a Riemannian metric form

As an example of the average reward metric D_η, we can simply consider the squared length between the average rewards with policy parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \boldsymbol{\Delta\theta}$ as

$$D_\eta[\boldsymbol{\theta}\|\boldsymbol{\theta} + \boldsymbol{\Delta\theta}] := \{\eta(\boldsymbol{\theta}) - \eta(\boldsymbol{\theta} + \boldsymbol{\Delta\theta})\}^2.$$

By taking the Taylor series expansion to the squared length, this metric can be approximately represented in a Riemannian metric form as

$$\{\eta(\boldsymbol{\theta}) - \eta(\boldsymbol{\theta} + \boldsymbol{\Delta\theta})\}^2 = \boldsymbol{\Delta\theta}^T\mathbf{G}(\boldsymbol{\theta})\boldsymbol{\Delta\theta} + O(\|\boldsymbol{\Delta\theta}\|^3) \tag{7}$$

where the metric matrix is

$$\boldsymbol{G}(\boldsymbol{\theta}) \equiv \nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})^T \tag{8}$$

and $\|\boldsymbol{a}\|$ denotes the Euclidean norm of the vector $\boldsymbol{a}$. We ignore the final term on the right-hand side of Eq. (7) by assuming $\|\boldsymbol{\Delta\theta}\| \ll 1$. Then we can use this metric matrix $\mathbf{G}(\boldsymbol{\theta})$ as an instance of the average reward metric D_η . In the rest of this paper, we consider the case $\mathbf{R}(\boldsymbol{\theta}) := \mathbf{G}(\boldsymbol{\theta})$. Note that the Reimannian metric matrix $\mathbf{R}(\boldsymbol{\theta})$ will no longer be a positive definite matrix.

3.3 APG as a Newton method

APG has another desirable property with respect to the convergence rate around the local optimal policies. Assuming a (locally) optimal average reward $\eta(\boldsymbol{\theta}^*)$ and the average reward for the current policy $\eta(\boldsymbol{\theta})$, we can calculate the error between them as

$$\mathcal{E}(\boldsymbol{\theta}) = \frac{1}{2}\{\eta(\boldsymbol{\theta}^*) - \eta(\boldsymbol{\theta})\}^2.$$

By using a second-order Taylor series expansion and simple analytic calculations, the second-order-convergence gradient ascent, called the "Newton method" is given as

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \nabla^2_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}), \tag{9}$$

where

$$\begin{cases} \nabla_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) = -\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) \\ \nabla^2_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T + \{\eta(\boldsymbol{\theta}^*) - \eta(\boldsymbol{\theta})\} \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^2. \end{cases}$$

When $\eta(\boldsymbol{\theta}) - \eta(\boldsymbol{\theta}^*) \approx 0$, which is true near locally optimal policies, the Hessian matrix $\nabla^2_{\boldsymbol{\theta}} \mathcal{E}$ becomes

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T = \mathbf{R}(\boldsymbol{\theta}).$$

Therefore, the proposed method is roughly equivalent to the Newton method near a (locally) optimal policy. The term *roughly* reflects that, while the Hessian is assumed to be a positive definite matrix without a step-size parameter, $\mathbf{R}(\boldsymbol{\theta})$ is not the positive definite matrix and there is adaptive learning in the APG approach. This is well-known in supervised learning contexts (Bishop, 1995). However, to our knowledge, it has never been exploited in the policy gradient reinforcement learning context.

4. Algorithms for APG approach

In this section, we propose several APG algorithms. Section 4.1 derives two algorithms for APG learning. Section 4.2 presents an on-line implementation of APG learning.

4.1 Derivations of algorithms for APG

Here are some APG algorithms. Before going into the details, we need to note that $\mathbf{R}(\boldsymbol{\theta})$ cannot be of the full-rank because it is a rank-one matrix. Thus, there must be some additional constraint to implement an APG algorithm. Here are two different algorithms with two different optimization forms.

The first algorithm is derived as an optimization form with a Euclidean norm objective function constrained by an average reward metric. The resulting policy gradient is a variant of the ordinary policy gradient with an adaptive step-size strategy. This is Algorithm 1. While this learning scheme should avoid a stagnant phase caused by the complex structure of the average reward, unlike NPG, it does not consider the curvature of the manifold in the policy parameter space. Since we want the advantages of both APG and NPG, we derived a second algorithm as an optimization form with a Riemannian metric objective function constrained by the average reward metric. The resulting policy gradient is a variant of the natural policy gradient with an adaptive step-size strategy. This is Algorithm 2. Here are the detailed derivations.

(Algorithm 1) We consider the following optimization form with a Euclidean norm objective function as

$$\min_{\Delta\boldsymbol{\theta}} \quad \Delta\boldsymbol{\theta}^T \Delta\boldsymbol{\theta}, \tag{10}$$

$$\text{s.t.} \quad \sqrt{\Delta\boldsymbol{\theta}^T \mathbf{R}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T \Delta\boldsymbol{\theta} = \epsilon_\eta. \tag{11}$$

The solution $\Delta\boldsymbol{\theta}^*$ is obtained as

$$\Delta\boldsymbol{\theta}^* = \epsilon_\eta \alpha^*(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}),$$

where

$$\alpha^*(\boldsymbol{\theta}) = \frac{1}{\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})}.$$

The gradient $\Delta\boldsymbol{\theta}^*$ can be regarded as an ordinary policy gradient with an adaptive step-size strategy, so, the policy update becomes $\boldsymbol{\theta} := \boldsymbol{\theta} + \Delta\boldsymbol{\theta}^*$. Note that this gradient is calculated using the ordinary policy gradient $\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})$. Therefore, this method does not require any additional estimation from an ordinary policy gradient at all. We call the gradient ascent by $\Delta\boldsymbol{\theta}^*$ Policy Gradient learning method on the *a*verage reward metric (*a*PG).

(Algorithm 2) We consider the following optimization form with a Riemannian metric objective function as

$$\min_{\Delta\boldsymbol{\theta}} \quad \Delta\boldsymbol{\theta}^T \mathbf{F}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}, \tag{12}$$

$$\text{s.t.} \quad \sqrt{\Delta\boldsymbol{\theta}^T \mathbf{R}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T \Delta\boldsymbol{\theta} = \epsilon_\eta, \tag{13}$$

where $\mathbf{F}(\boldsymbol{\theta}) = \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(s,a;\boldsymbol{\theta}) \left[\nabla_{\boldsymbol{\theta}} \ln \pi(a,s;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \ln \pi(a,s;\boldsymbol{\theta})^T\right]$.

The solution $\Delta\boldsymbol{\theta}^\star$ is

$$\Delta\boldsymbol{\theta}^\star = \epsilon \alpha^\star(\boldsymbol{\theta}) \mathbf{F}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}),$$

where

$$\alpha^\star(\boldsymbol{\theta}) = \frac{1}{\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})^T \mathbf{F}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})}.$$

In Eq. (12), the objective function is $D_{p(s,a)}[\boldsymbol{\theta} \| \boldsymbol{\theta} + \Delta\boldsymbol{\theta}] = \Delta\boldsymbol{\theta}^T \mathbf{F}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}$ that measures the effect on an action probability distribution with respect to the policy parameters (Kakade, 2002). Thus, the gradient $\Delta\boldsymbol{\theta}^\star$ can be regarded as NPG with an adaptive step-size strategy. We call this gradient ascent by $\Delta\boldsymbol{\theta}^\star$ Natural Policy Gradient learning method on the *a*verage reward metric (*a*NPG). The difference between the *a*NPG and the NPG comes from the adaptive step-size parameter derived from the average reward metric. This difference can significantly improve the learning performance. Details about the differences are described in Appendix A. The effectiveness is validated by the numerical experiments in Section 5.

4.2 On-line learning implementation of Algorithms (1) and (2)

The proposed APG Algorithm (1) only requires $\nabla_{\boldsymbol{\theta}} \eta$ and Algorithm (2) additionally requires an estimate of $\mathbf{F}(\boldsymbol{\theta})$. Here we give a straightforward way to calculate these variables when implementing the APG algorithms. To estimate $\nabla_{\boldsymbol{\theta}} \eta$, we simply use the on-line version of the GPOMDP algorithm in (Baxter and Bartlett, 2001b). The Fisher information matrix $\mathbf{F}(\boldsymbol{\theta})$ is estimated as the mean of the exponential recency-weighted average (Sutton and Barto, 1998) as $\hat{\mathbf{F}}_{t+1} := \hat{\mathbf{F}}_t + \lambda \left(\nabla_{\boldsymbol{\theta}} \ln \pi_{t+1} \nabla_{\boldsymbol{\theta}} \ln \pi_{t+1}{}^T - \hat{\mathbf{F}}_t \right)$, where λ is a forgetting parameter. These are shown in Algorithms 1 and 2. To reduce the variance of the estimator, the constant baseline, as proposed by (Weaver and Tao, 2001), is estimated as $b_{t+1} = b_t + (r(s_{t+1}, a_{t+1}) - b_t)/(t+1)$, which is used in both algorithms.

Algorithm 1 On-line aPG algorithm

Given: an ergodic (PO)MDP, parametrized by $\boldsymbol{\theta}$
Initialize $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, $\epsilon, \beta, \lambda \in [0, 1)$, $\mathbf{z} = 0$, $\mathbf{g} = 0$
repeat
  state transition from s_t with action a_t to s_{t+1}
  $\mathbf{z}_{t+1} = \beta \mathbf{z}_t + \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t)$
  $\mathbf{g}_{t+1} = \mathbf{g}_t + \lambda \left[\left(r(s_{t+1}, a_{t+1}) - b(s_{t+1}) \right) \mathbf{z}_{t+1} - \mathbf{g}_t \right]$
  $\gamma = \left\{ \mathbf{g}_{t+1}^T \mathbf{g}_{t+1} \right\}$, $\tilde{\mathbf{g}}_{t+1} = \mathbf{g}_{t+1} / \gamma$
  $\boldsymbol{\theta} := \boldsymbol{\theta} + \epsilon \tilde{\mathbf{g}}_{t+1}$
until $\sqrt{\mathbf{g}_t^T \mathbf{g}_t} \approx 0$

Algorithm 2 On-line aNPG algorithm

Given: an ergodic (PO)MDP, parametrized by $\boldsymbol{\theta}$
Initialize $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, $\epsilon, \beta, \lambda \in [0, 1)$, $\lambda_F \in [0, 1)$, $\mathbf{z} = 0$, $\mathbf{g} = 0$
repeat
  state transition from s_t with action a_t to s_{t+1}
  $\mathbf{z}_{t+1} = \beta \mathbf{z}_t + \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t)$
  $\mathbf{g}_{t+1} = \mathbf{g}_t + \lambda \left[\left(r(s_{t+1}, a_{t+1}) - b(s_{t+1}) \right) \mathbf{z}_{t+1} - \mathbf{g}_t \right]$
  $\hat{\mathbf{F}}_{t+1} := \hat{\mathbf{F}}_t + \lambda_F \left(\nabla_{\boldsymbol{\theta}} \ln \pi_t \nabla_{\boldsymbol{\theta}} \ln \pi_t{}^T - \hat{\mathbf{F}}_t \right)$
  $\gamma = \left\{ \mathbf{g}_{t+1}^T \hat{\mathbf{F}}_{t+1}^{-1} \mathbf{g}_{t+1} \right\}$, $\tilde{\mathbf{g}}_{t+1} = \mathbf{g}_{t+1} / \gamma$
  $\boldsymbol{\theta} := \boldsymbol{\theta} + \epsilon \hat{\mathbf{F}}_{t+1}^{-1} \tilde{\mathbf{g}}_{t+1}$
until $\sqrt{\mathbf{g}_t^T \hat{\mathbf{F}}_t^{-1} \mathbf{g}_t} \approx 0$

5. Numerical Experiments

In this section, the proposed algorithms for APG learning are validated through numerical simulations. In Section 5.1, we investigate the algorithms with analytically computed policy gradients for 3-state MDPs. Then, in Section 5.2, the performance with estimated gradients in an on-line approach is validated for a more challenging task with large (20-state) MDPs followed by comparisons.

5.1 Analytical approach: applications to 3-state MDPs

To investigate our suggested approach for APG learning, we considered the application of the proposed algorithms to MDPs. Here an analytical policy gradient approach, as proposed in (Baxter and Bartlett, 2001b,a), was used to avoid any algorithmic disturbance such as estimation errors of the gradient. By using the reward function and the model of an environment, we can analytically obtain the exact policy gradient $\nabla_{\boldsymbol{\theta}} \eta(\eta)$. We applied our proposed aPG and aNPG to two 3-state MDPs in (Baxter and Bartlett, 2001a) and (Schraudolph et al., 2006)), where the state is defined as $s \in S = \{A, B, C\}$ and the action is $a \in A = \{a_1, a_2\}$. We also applied NPG, OPG, and fNPG (conventional NPG learning with an adaptive step-size strategy as in (Kakade, 2002; Peters and Schaal, 2008), see Appendix

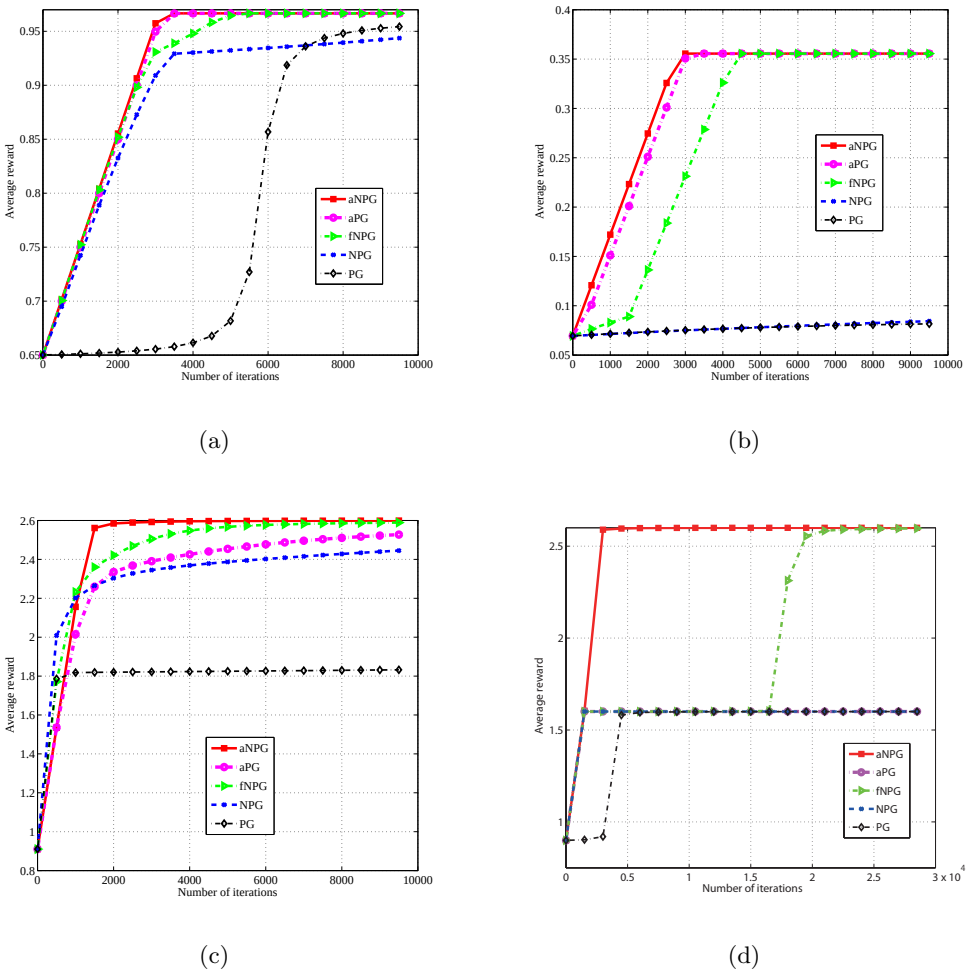

Figure 1: Simulation results obtained by several policy gradient methods in which each gradient is analytically calculated by using models of environment and reward functions. (a) is obtained from MDP1. The step-size parameters for all of the methods are $\alpha_{\mathrm{PG}} = 0.05$, $\alpha_{\mathrm{NPG}} = 0.0005$, $\epsilon_{f\mathrm{NPG}} = 0.09$, $\epsilon_{a\mathrm{PG}} = 0.1$, and $\epsilon_{f\mathrm{NPG}} = 0.1$. The initial policy parameter is set $\boldsymbol{\theta}_0 = [2.5, 2.5]^T$. (b) is with $\boldsymbol{\theta}_0 = [25, -25]^T$ and the same learning rate in (a). These settings are set following (Baxter and Bartlett, 2001b). (c) is obtained by MDP2 with $\alpha_{\mathrm{PG}} = 0.000125$, $\alpha_{\mathrm{NPG}} = 0.0005$, $\epsilon_{f\mathrm{NPG}} = 0.16$, $\epsilon_{a\mathrm{PG}} = 1.25$, $\epsilon_{a\mathrm{NPG}} = 1.25$, and $\boldsymbol{\theta}_0 = [-0.1, 0.1]^T$ following (Schraudolph et al., 2006). (d) is with $\alpha_{\mathrm{PG}} = 0.00025$, $\alpha_{\mathrm{NPG}} = 0.001$, $\epsilon_{f\mathrm{NPG}} = 0.22$, $\epsilon_{a\mathrm{PG}} = 2.5$, $\epsilon_{a\mathrm{NPG}} = 2.5$, and $\boldsymbol{\theta}_0 = [-10.0, 0.5]^T$.

A. for more details) as the baseline algorithms for PG learning. The control policy $\pi(s, a; \boldsymbol{\theta})$ is defined with the policy parameters $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$.

As pointed out in (Schraudolph et al., 2006), Baxter's 3-state MDP has the property that the greedy maximization of the instantaneous reward leads to an optimal policy. In contrast, Schraudolph's 3-state MDP is more challenging by modifying the state transitions, the reward structure, and the state features in Baxter's MDP. In the rest of this section, we refer to Baxter's MDP as MDP1 and Schraudolph's as MDP2. The experimental comparisons are shown in Fig. 1. Two reward functions were used for both MDP1 and MDP2. Each sub-figure in Fig. 1 shows the obtained average rewards against the number of iterations.

PG and NPG encountered serious plateau phenomena during learning, perhaps due to the curvature of the manifold in the policy parameter space. While *f*NPG outperformed PG and NPG, it still saw some unnaturally slow convergence due to the lack of reward information in the metric. The *a*PG method considered the reward information in the metric, which significantly accelerated the learning process in (a) to (c), but, the negative effects from the curvature of the manifold could not be avoided in (d). The *a*NPG method, which considered both the curvature of the parameter manifold and the average reward metric, resulted in the best performance in all of the test cases.

5.2 On-line learning for large MDPs

We tested the PG algorithms (*a*NPG, *a*PG, *f*NPG, NPG, and OPG) while estimating the gradients for random synthesized 20-state MDPs, which are useful for various conditions as in (Morimura et al., 2008; Wang et al., 2008). Each MDP was initialized in each episode. Set of actions initialized as $\mathcal{A}=\{a_1, a_2\}$. The state transition probability function was set using a Dirichlet distribution $\mathtt{Dir}(\boldsymbol{\alpha} \in \mathbb{R}^2)$ and a uniform distribution $\mathtt{U}(20; b)$ generating integers from 1 to 20 except b. We first initialized it so that $p(s'|s,a) := 0$, $^{\forall}(s', s, a)$ and then with $\boldsymbol{q}(s,a) \sim \mathtt{Dir}(\boldsymbol{\alpha}=[.3,.3])$ and $x_{\backslash b} \sim \mathtt{U}(|\mathcal{S}|; b)$ using

$$\begin{cases} p(s+1|s,a_1) := q_1(s,a_1) \\ p(x_{\backslash s+1}|s,a_1) := q_2(s,a_1) \end{cases} \qquad \begin{cases} p(s|s,a_2) := q_1(s,a_2) \\ p(x_{\backslash s}|s,a_2) := q_2(s,a_2) \end{cases}$$

where $s'=1$ and $s'=21$ are identical states. The reward function $r(s,a,s')$ is set for each argument by using a Gaussian distribution $\mathrm{N}(\mu=0, \sigma^2=1)$ and normalized such that $\max_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) = 1$ and $\min_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) = -1$ as

$$r(s,a,s') := \frac{2(r(s,a,s') - \min_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}))}{\max_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) - \min_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta})} - 1.$$

The policy was represented by the sigmoidal function:

$$\begin{cases} \pi(l|i;\boldsymbol{\theta}) &= \dfrac{1}{1+\exp(-\boldsymbol{\theta}^\top \boldsymbol{\phi}(i))} \\ \pi(m|i;\boldsymbol{\theta}) &= 1 - \pi(l|i;\boldsymbol{\theta}). \end{cases}$$

Each ith element of the initial policy parameter $\boldsymbol{\theta}_0 \in \mathbb{R}^{20}$ and the features of state s_j, $\boldsymbol{\phi}(s_j) \in \mathbb{R}^{20}$ were drawn from $\mathrm{N}(0,1)$ and $\mathrm{N}(\delta_{\mathrm{ij}}, 0.5)$, where δ_{ij} is the Kronecker delta. The hyper parameters β, λ, and λ_F (for the NPGs) were set as 0.998, 0.001, and 0.0002. The step-size parameters were initialized to $\alpha_{\mathrm{PG}} = 0.0002$, $\alpha_{\mathrm{NPG}} = 0.00001$, $\epsilon_{f\mathrm{NPG}} = 0.00005$ $\epsilon_{a\mathrm{PG}} = 0.00005$, and $\epsilon_{a\mathrm{NPG}} = 0.00005$. Here we introduced a heuristic where, if the

adaptive step-sizes in aNPG, aPG, and fNPG are larger than $10\alpha_{\mathrm{PG}}$, $10\alpha_{\mathrm{NPG}}$, and $10\alpha_{\mathrm{NPG}}$, respectively, they are reset to $10\alpha_{\mathrm{PG}}$, $10\alpha_{\mathrm{NPG}}$, and $10\alpha_{\mathrm{NPG}}$.

Figure 2 shows the learning performances. The learning process of aNPG was considerably faster than the other methods. We thus confirmed that, just as with the results of the analytical approach in Section 5.1, our aNPG algorithm outperformed the other PG algorithms even in larger MDPs and with the settings for on-line learning.

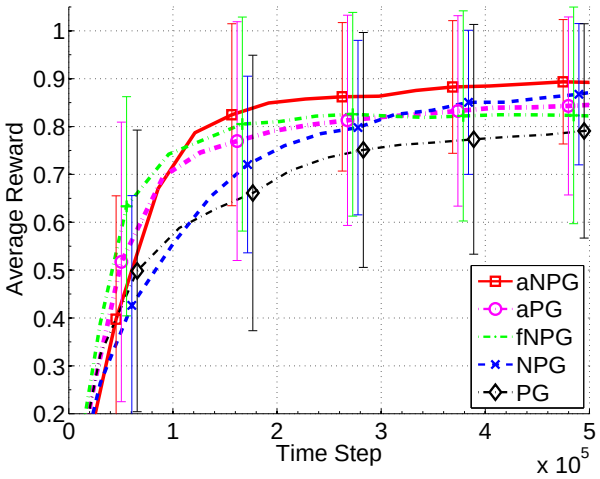

Figure 2: Means and standard deviations over 50 independent episodes. The learning performances (average rewards) are from various PG algorithms on the 20-state MDP.

6. Conclusions

In this paper, we proposed a novel adaptive step-size policy gradient reinforcement learning approach in an average reward metric space. A new metric was defined for policy gradients to assess the effects of changes in the average reward with with respect to the policy parameters. Since the metric measures effect on the average reward, it effectively avoids falling into a stagnant phase caused by the complex structure of the average reward. The difference between aNPG and Kakade's NPG is the adaptive step-size parameter. Though it may seem to be small difference, it can significantly change the properties of the derived policy gradient. Experimental results verified this with simple, but non-trivial, 3-state MDPs and more challenging 20-state MDPs. Future work includes the development of more sophisticated algorithms for on-line learning implementations with LSPI as proposed by (Lagoudakis and Parr, 2003) and an application of our method to a high-dimensional robotic arm for optimal control. Another direction of the future work would be qualitative analysis on the cause of the stagnant phase for the policy gradient methods.

Acknowledgments

This research was supported by the Strategic Research Program for Brain Science (SRPBS), the Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science (WAKATE-B22700177) and the Promotion program for Reducing global Environmental loaD through ICT innovation (PREDICT) of the Ministry of Internal Affairs and Communications, Japan.

Appendix A. Comparison of *a*NPG with Kakade's NPG

We presented an adaptive step-size parameter for conventional NPG learning in (Kakade, 2002)[1]. Since NPG learning is derived from the constraint $D_{p(s,a)} = \epsilon^2_{p(s,a)}$, it can be regarded as the optimization problem

$$\max_{\Delta\boldsymbol{\theta}} \quad \nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})^T\Delta\boldsymbol{\theta}, \tag{14}$$

$$\text{s.t.} \quad \sqrt{\Delta\boldsymbol{\theta}^T\mathbf{F}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}} = \epsilon_{p(s,a)}. \tag{15}$$

As a solution, an adaptive step-size gradient ascent is obtained as

$$\tilde{\Delta}\boldsymbol{\theta} = \epsilon_{p(s,a)}\tilde{\alpha}(\boldsymbol{\theta})\mathbf{F}^{-1}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})$$

where

$$\tilde{\alpha}(\boldsymbol{\theta}) = \frac{1}{\sqrt{\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})^T\mathbf{F}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta})}}.$$

We call this gradient ascent $\tilde{\Delta}\boldsymbol{\theta}$ Natural Policy Gradient learning on the *F*isher information metric (*f*NPG).

Note that the difference between *a*NPG and *f*NPG comes from these step-size parameters. The step-size parameter $\alpha^\star(\boldsymbol{\theta})$ in *a*NPG is the square of $\tilde{\alpha}(\boldsymbol{\theta})$ in *f*NPG. If $\epsilon_\eta = \epsilon_{p(s,a)}$, then $\alpha^\star$ is larger than $\tilde{\alpha}$ if $\alpha^\star$ is less than one. Otherwise, $\alpha^\star$ is smaller than $\tilde{\alpha}$ if $\alpha^\star$ is more than one. These observations suggest that a more conservative strategy than NPG exists for an adaptive step-size parameter for policy search.

References

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

James Bagnell and Jeff Schneider. Covariant policy search. In *Proceeding of the International Joint Conference on Artifical Intelligence*, pages 1019–1024, 2003.

Jonathan Baxter and Peter L. Bartlett. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001a.

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

1. In (Peters and Schaal, 2008), an adaptive step-size parameter for NPG learning was shown. However, the step-size parameters in their presented algorithms were constant.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control(Volume 1 and 2)*. Athena Scientific, 1995.

Christopher M. Bishop. *Neural Networks for Pattern Regrognition*. Oxford University Press, 1995.

Mohammad Ghavamzadeh and Yaakov Engel. Bayesian policy gradient algorithms. In *Advances in Neural Information Processing Systems*, pages 457–464, 2006.

Sham Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, pages 1531–1538, 2002.

Hajime Kimura and Shigenobu Kobayashi. An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value function. *International Conference on Machine Learning*, pages 278–286, 1998.

Vijay R. Konda and John N. Tsitsiklis. On actor-critic algorithms. *Society for Industrial and Applied Mathematics*, 42(4):1143–1166, 2003.

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

Takamitsu Matsubara, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, and Kenji Doya. Learning CPG-based biped locomotion with a policy gradient method. *Robotics and Autonomous Systems*, (54(11)):911–920, 2006.

Tetsuro Morimura, Eiji Uchibe, Junichiro Yoshimoto, and Kenji Doya. A new natural policy gradient by stationary distribution metric. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 82–97, 2008.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.

Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Reinforcement learning for humanoid robotics. In *Third IEEE-RAS International Conference on Humanoid Robots*, pages 1137–1144, 2003.

Silvia Richter, Douglas Aberdeen, and Jin Yu. Natural actor-critic for road traffic optimisation. In *Advances in Neural Information Processing Systems*, pages 3522–3529, 2006.

Nicol N. Schraudolph, Jin Yu, and Douglas Aberdeen. Fast online policy gradient learning with SMD gain vector adaptation. In *Advances in Neural Information Processing Systems*, pages 1185–1192, 2006.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

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

Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Stable dual dynamic programming. In *Advances in Neural Information Processing Systems*, pages 1569–1576. MIT Press, 2008.

Chris Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventheenth Conference*, pages 538–545, 2001.

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