The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Natural Actor-Critic with Baseline Adjustment for Variance Reduction

Tetsuro Morimura[†,††] Eiji Uchibe[†,] Kenji Doya[†,††,†††]

[†] Graduate School of Information Science, Nara Institute of Science and Technology
[††] Initial Research Project, Okinawa Institute of Science and Technology
[†††] ATR Computational Neuroscience Laboratories
{morimura, uchibe, doya}@oist.jp

Abstract

In this study, we discuss the baseline function for the estimation of the natural policy gradient with respect to the variance and show a condition that an optimal baseline function reducing the variance is equivalent to the state value function. However, the state value could be much different from the optimal baseline outside of the condition. For such cases, an extended version of the NTD algorithm [1] is proposed, where an auxiliary function is estimated to adjust the baseline, being state value estimates in the original version, to the optimal baseline. The proposed algorithm is applied to simple MDP and a challenging pendulum swing-up problem.

Keywords: policy gradient reinforcement learning, natural gradient, variance reduction.

1 Introduction

Policy gradient (PG) methods for reinforcement learning (RL) attempt to maximize the average (or discounted cumulative) reward by improving the policy parameter on the basis of the stochastic gradient decent with the experienced system trajectories of states, actions, and rewards [2]. Kakade proposed a "*natural policy gradient (NPG)*" as an efficient covariant gradient of the average reward to reduce the effect of plateau phenomenon [3]. The actor-critic framework with the NPG is called Natural Actor-Critic (NAC), i.e. the actor as the policy is updated by the NPG estimated in the critic [4]. In our previous studies [1], the natural policy gradient utilizing temporal differences (NTD) algorithm as an implementation of NAC was proposed, where the NPG is estimated without matrix inversion. While the original NTD algorithm uses the state value function as the baseline function for estimating the NPG, it has not been clarified whether the state value function is a valid baseline function for reduction of the variance of the estimated gradients.

Here, we discuss the baseline function for the above question by using the results of Greensmith et al. [5]. We introduce an optimal baseline function that minimizes an upper bound of the variance of the NPG estimates, and see that the optimal function can be different from the state value function. We then show the condition, under which the optimal baseline function is equivalent to the state value function. Although the state value function is a valid baseline function under the condition, the state value could be invalid baseline function outside of the condition, being much different from the optimal baseline. For such case, we propose an "extended NTD algorithm", which compensates for the differences between the state value function and the optimal baseline function by introducing an auxiliary function. An algorithm to estimate the auxiliary function is also proposed. These proposed algorithms are applied to simple Markov decision problems to confirm their performances. The extended NTD algorithm is also applied to a more challenging nonlinear pendulum swing-up problem to show its effectiveness compared with other PG methods.

2 Background of NTD Algorithm

2.1 Natural policy gradient

While PG is based on gradient decent, one of the major limitations of standard gradient descent algorithms is that the ordinary gradient of a function does not necessarily indicate its steepest direction, because a parameter as an input of the function might not be expressed in orthonormal coordinates in terms of change of an output from the function. In order to overcome this problem, Amari proposed the concept of natural gradient to define the steepest direction based on Riemannian geometry [6]. The natural gradient of an objective function $R(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is

$$\widetilde{\nabla}_\theta R(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta})^{-1} \nabla_\theta R(\boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{G}(\boldsymbol{\theta})$ is a Riemannian metric matrix of $\boldsymbol{\theta}$, which is defined by the Fisher information matrix (FIM) in

the case that $\boldsymbol{\theta}$ is a parameter of a statistical model, and $\nabla_\theta R(\boldsymbol{\theta})$ is derivative of $R(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

While the above work deals with the case that a training sample was independent and identically distributed, Kakade extends it to the case of Markov decision process (MDP), which is a model of RL, and propose the natural policy gradient (NPG) as the natural gradient of the average reward, being the object function, of RL with respect to the policy parameter. 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 $p(x_{t+1}|x_t, u_t)$ and a reward function $r_{t+1} = r(x_{t+1}, x_t, u_t)$ at all time steps t [7, 8]. The decision-making follows a stochastic policy $\pi_\theta(u|x) \equiv p(u|x;\boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$. We assume that the every policy π is differentiable with $\boldsymbol{\theta}$ for all $x \in \mathbb{X}$ and $u \in \mathbb{U}$ and makes an ergodic Markov chain, i.e. the stationary state distribution always exists $d^\pi(x') = \sum_{x,u} p(x'|x,u)\pi_\theta(u|x)d^\pi(x)$.

In NPG, $\boldsymbol{G}(\boldsymbol{\theta})$ and $R(\boldsymbol{\theta})$ of eq.1 are the time average of $-\nabla_\theta^2 \ln \pi$ and the average reward, respectively,

$$\boldsymbol{G}(\boldsymbol{\theta}) = \lim_{T\to\infty} \frac{-1}{T} \mathrm{E}^\pi \left\{ \sum_{t=0}^{T-1} \nabla_\theta^2 \ln \pi_\theta(u_t|x_t) \right\} = \sum_x d^\pi(x) \boldsymbol{F}(x, \boldsymbol{\theta}),$$

$$R(\boldsymbol{\theta}) = \lim_{T\to\infty} \frac{1}{T} \mathrm{E}^\pi \left\{ \sum_{t=1}^{T} r_t \right\} = \sum_{x',x,u} p(x'|x,u)\pi_\theta(u|x)d^\pi(x)r(x',x,u),$$

where $\boldsymbol{F}(x,\boldsymbol{\theta}) \equiv -\mathrm{E}^\pi\{\nabla_\theta^2 \ln \pi_\theta(u|x)|x\}$ is the FIM of π given x with respect to $\boldsymbol{\theta}$. It is noted that $G(\boldsymbol{\theta})$ is equivalent to the scaled FIM of the system trajectories [9, 4]. An important property of NPG $\widetilde{\nabla}_\theta R(\boldsymbol{\theta})$ is the relationship with the linear function approximator $f_\omega^\pi(x,u) \equiv \boldsymbol{\omega}^\top \nabla_\theta \ln \pi_\theta(u|x)$, where $\boldsymbol{\omega}$ is called the weight, for the action value function $Q^\pi(x,u) \equiv \lim_{T\to\infty} \mathrm{E}^\pi\left\{\sum_{k=1}^T \gamma^{k-1} r_{t+k} | x_t = x, u_t = u\right\}$ with the discounted rate $\gamma \in [0,1)$, i.e.,[1]

$$\widetilde{\nabla}_\theta R(\boldsymbol{\theta}) \simeq \boldsymbol{\omega}^*. \quad (2)$$

$\boldsymbol{\omega}^*$ is the parameter minimizing the approximate error $\mathrm{E}^\pi\{(Q^\pi - f_\omega^\pi)^2\}$[3]. Here, we also notate the state value function $V^\pi(x) \equiv \mathrm{E}^\pi\{Q^\pi(x,u)|x\}$ and show a following proposition for the policy parameterization:

Proposition 1 *Let X and U_i denote the number of states and available actions at state x_i, respectively. Let $\boldsymbol{\Psi}(\boldsymbol{\theta})$ denote the subspace spanned by $\nabla_\theta \ln \pi_\theta(u|x)$ over states and actions. If the rank of $\boldsymbol{\Psi}(\boldsymbol{\theta})$ is equal to (or greater than) $\sum_{i=1}^X (U_i - 1)$, the policy parameterization is nondegenerate for the task:*

$$f_{\omega^*}^\pi(x,u) \equiv \boldsymbol{w}^{*\top} \nabla_\theta \ln \pi_\theta(u|x) = Q^\pi(x,u) - V(x)^\pi. \quad (3)$$

Proof sketch: It comes from the fact that there is a constraint $\sum_{\boldsymbol{u}} Q^\pi(x,\boldsymbol{u}) - V^\pi(x) = 0$ for each state.

[1] "$\simeq$" in eq.2 is replaced "$=$", in the limit $\gamma \to \infty$, using an undiscounted value function instead of Q^π, or using a discounted average reward instead of $R(\boldsymbol{\theta})$ [3, 9, 4].

2.2 NAC and NTD algorithm

The actor-critic framework for NPG is called the natural actor-critic (NAC) [4]. The *critic* estimates NPG $\hat{\boldsymbol{\omega}}$ and the *actor* executes the action drawn from the policy $\pi_\theta(u|x)$, which is updated by the critic's estimate: $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha\hat{\boldsymbol{\omega}}$, where ":=" denotes the substitution of the right to the left and α is learning rate.

NTD algorithm in our previous work [1] is an implementation of NAC without matrix inversion, comprising the repetition of following three procedures. The first procedure updates the state value estimate $\hat{V}(x)$ by TD(λ) learning [8]. The second updates the NPG estimate $\hat{\boldsymbol{\omega}}$ through the regression with the linear function $f_{\hat{\omega}}^\pi(x_t,u_t) = \hat{\boldsymbol{\omega}}^\top \nabla_\theta \ln \pi_\theta(u_t|x_t)$ to the temporal difference (TD) given from the first,

$$\delta(x_t,u_t) = r_{t+1} + \gamma\hat{V}^\pi(x_{t+1}) - \hat{V}^\pi(x_t).$$

That is, the update direction of NPG estimate $\hat{\boldsymbol{\omega}}$ is [2]

$$\Delta\hat{\boldsymbol{\omega}} = \frac{1}{T}\sum_{t=0}^{T-1} \left(\delta(x_t,u_t) - f_{\hat{\omega}}^\pi(x_t,u_t)\right) \nabla_\theta \ln \pi_\theta(u_t|x_t). \quad (4)$$

The third updates the policy parameter $\boldsymbol{\theta}$ is updated by the weight $\hat{\boldsymbol{\omega}}$ of $f_{\hat{\omega}}^\pi$ in the second.

Since $f_\omega^\pi(x,u)$ has the property for an arbitrary function $a(x)$, due to $\sum_u \nabla\pi = \boldsymbol{0}$,

$$\mathrm{E}^\pi\{a(x)\nabla_\theta \ln \pi(u|x)|x\} = \boldsymbol{0},$$

the expectation of $\Delta\hat{\boldsymbol{\omega}}$ at a time-step t (eq.4) does not depend on the value of $\hat{V}(x_t)$. Therefore, the NTD algorithm uses the state value estimate at the current time-step as the baseline function $b(x)$ for estimating the NPG. However it has not been clarified whether the state value function is a valid baseline function for the variance reduction of $\hat{\boldsymbol{\omega}}$.

3 Variance Reduction for Natural Policy Gradient Estimates

3.1 Optimal baseline function $b^*(x,\hat{\boldsymbol{\omega}})$

Consider a trace of the covariance matrix of the NPG estimates $\hat{\boldsymbol{w}}$ as the variance of $\hat{\boldsymbol{\omega}}$, [3]

$$\mathrm{Var}^\pi(\hat{\boldsymbol{\omega}}) = \mathrm{E}^\pi\left\{(\hat{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}}^*)^2\right\},$$

where $\boldsymbol{a}^2$ denotes $\boldsymbol{a}^\top\boldsymbol{a}$ for an arbitrary vector $\boldsymbol{a}$, and $\hat{\boldsymbol{\omega}}^* \equiv \mathrm{E}^\pi\{\hat{\boldsymbol{\omega}}\}$ has to be equal to $\boldsymbol{w}^*$ for the unbiased regression. In gradient decent regressions, however, it is difficult to treat directly with the variance of $\hat{\boldsymbol{w}}$. Instead we consider $\mathrm{Var}^\pi(\Delta\hat{\boldsymbol{w}})$, the variance of the

[2] While the NTD algorithm uses the eligibility trace in this procedure, here is the decay rate $\lambda = 0$. We omit the cases of arbitrary $\lambda \in [0,\gamma]$, though results in this report are applicable.

[3] Peters *et.al.*[10] consider $\left\langle(\hat{\boldsymbol{w}} - \langle\hat{\boldsymbol{w}}\rangle)^\top \boldsymbol{G}(\boldsymbol{\theta})(\hat{\boldsymbol{w}} - \langle\hat{\boldsymbol{w}}\rangle)\right\rangle$ taking account of the metric of the policy parameters as a proper variance about $\hat{\boldsymbol{w}}$, instead of $\mathrm{Var}^\pi(\hat{\boldsymbol{w}})$. These results of this section can be applied instantly to the case of the above variance.

update direction $\Delta\hat{\boldsymbol{w}}$ for $\hat{\boldsymbol{w}}$ (at a fixed policy $\boldsymbol{\theta}$). Although a sequence of samples $[x_1, ..., x_T]$ is not drawn independently in almost cases of RL, where the relationship $\mathrm{Var}^\pi(\frac{1}{T}\sum_t f(x_t)) = \frac{1}{T}\mathrm{Var}^\pi(f(x))$ does not hold due to correlation between the different time-step samples, Greensmith *et.al.*[5] derive useful results about the variance at a finite ergodic Markov chain. By applying Corollary 5 and Lemma 6 with the increasing function h^π in [5], the following inequality holds

$$\mathrm{Var}^\pi(\Delta\hat{\boldsymbol{w}}) \le o + h^\pi\left(\frac{1}{T}\mathrm{Var}^\pi\left((\hat{Q}(x,u) - b(x) - f^\pi_{\hat{\boldsymbol{\omega}}}(x,u))\nabla_\theta \ln \pi_\theta(u|x)\right)\right), \quad (5)$$

where o is independent with the choice of $b(x)$, and $\hat{Q}(x_t,u_t) = \mathrm{E}^\pi\left\{r_{t+1} + \gamma\hat{V}(x_{t+1})|x_t,u_t\right\}$ and $b(x) = \hat{V}(x)$.

Because we are interested in the choice of the baseline function as $b(x) = \hat{V}(x)$, the following looks for the optimal baseline function $b^*(x,\hat{\boldsymbol{\omega}})$ that minimizes the upper bound of $\mathrm{Var}^\pi(\Delta\hat{\boldsymbol{w}})$ with respect to $b(x)$ and also minimizes the part of the argument of the function h^π,

$$\sigma^2_{\Delta\hat{w}}(b(x)) \equiv \mathrm{Var}^\pi\left((\hat{Q}(x,u) - b(x) - f^\pi_{\hat{\omega}}(x,u))\nabla_\theta \ln \pi_\theta(u|x)\right)$$
$$= \mathrm{E}^\pi\left\{\left((\hat{Q}(x,u) - b(x) - f^\pi_{\hat{\omega}}(x,u))\nabla_\theta \ln \pi_\theta(u|x) - \mathrm{E}^\pi\{\Delta\hat{\boldsymbol{w}}\}\right)^2\right\}.$$

Accordingly, since the optimal baseline $b^*(x,\hat{\boldsymbol{\omega}})$ holds

$$\left.\frac{\partial\sigma^2_{\Delta\hat{w}}(b(x))}{\partial b(x)}\right|_{b(x)=b^*(x,\hat{\boldsymbol{\omega}})} = 0, \quad {}^\forall x \in \mathbb{X},$$

it is derived as

$$b^*(x,\hat{\boldsymbol{w}}) = \frac{\mathrm{E}^\pi\left\{\nabla_\theta \ln \pi_\theta(u|x)^2(\hat{Q}(x,u) - f^\pi_{\hat{\omega}}(x,u))|x\right\}}{\mathrm{E}^\pi\{\nabla_\theta \ln \pi_\theta(u|x)^2|x\}}. \quad (6)$$

Note that b^* has arguments not only x but also $\hat{\boldsymbol{\omega}}$ due to $f^\pi_{\hat{\omega}}(x,u) = \hat{\boldsymbol{\omega}}^\top \nabla_\theta \ln \pi_\theta(u|x)$.

3.2 Consistency of $V^\pi(x)$ and $b^*(x,\hat{\boldsymbol{\omega}})$

Proposition 2 *If the condition of proposition 1 is satisfied,*

$$b^*(x,\hat{\boldsymbol{\omega}}^*) = \hat{V}(x).$$

Proof sketch: It is obvious by substituting eq.3, "$\hat{Q}(x,u) - f^\pi_{\hat{\omega}^*}(x,u) = \hat{V}(x)$", to eq.6. □

Proposition 2 means that the optimal baseline is equivalent to the sate value, if following two conditions are satisfied; (i) the policy parameterization is nondegenerate for the task and (ii) the NPG estimate converges to the exact NPG.

In the NTD algorithm, the condition (ii), $\hat{\boldsymbol{\omega}} \simeq \hat{\boldsymbol{\omega}}^*$, should be realized under appropriate updatings on both the policy parameter as the actor parameter and the NPG estimate in the critic parameter. It indicates that the state value function would not be different from the optimal baseline function so much in cases using "appropriate" policy parameterization. Therefore, the state value function could be a valid baseline function in such cases.

4 Extended NTD algorithm

In this section, we deal with the cases where the condition (i) and/or (ii) could be violated. In these cases, the state value function can be much different from the optimal baseline function. Therefore, we propose an extended NTD algorithm, which compensates for the differences between the state value function and the optimal baseline function by introducing an auxiliary function,

$$B(x,\hat{\boldsymbol{\omega}}) = \frac{\mathrm{E}^\pi\left\{\nabla_\theta \ln \pi_\theta(u|x)^2(\hat{Q}(x,u) - \hat{V}(x) - f^\pi_{\hat{\omega}}(x,u))|x\right\}}{\mathrm{E}^\pi\{\nabla_\theta \ln \pi_\theta(u|x)^2|x\}}.$$

The extended NTD algorithm is the same as the original one, except that the auxiliary function is subtracted from TD as the regressand for the NPG estimation,

$$\delta(x_t,u_t) - B(x_t,\hat{\boldsymbol{\omega}}) = r_{t+1} + \gamma V(x_{t+1}) - b^*(x_t,\hat{\boldsymbol{\omega}}). \quad (7)$$

Although eq.7 seems roundabout to apply the optimal baseline, it is useful for an eligibility trace technique with estimated value functions (see fig.1). In order to estimate $B(x,\hat{\boldsymbol{w}})$, the gradient of $\sigma^2_{\Delta\hat{\boldsymbol{w}}}(b(x) = V^\pi(x) + \hat{B}_b(x,\hat{\boldsymbol{w}}))$ with respect to the parameter $\boldsymbol{b}$ of $\hat{B}_b(x)$ is used. Fig.1 is one of the complete algorithms.

Input:
- Initial parameters; $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, $\boldsymbol{v}$, [$\boldsymbol{b}$] are the parameters of $\pi(u|x)$, $f^\pi_{\omega}(x,u) = \boldsymbol{\omega}^\top\nabla_\theta \ln \pi_\theta(u|x)$, $\hat{V}(x)$, [$\hat{B}(x)$].
- Metaparameters; γ is the discouted rate of the value function, α_θ, α_ω, α_v, [α_b] are the learning rates of $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, $\boldsymbol{v}$, [$\boldsymbol{b}$]. λ_ω, λ_v,[λ_b] are the eligibility decay rates of $\boldsymbol{\omega}$, $\boldsymbol{v}$, [$\boldsymbol{b}$]. β is the forgetting rate of $\boldsymbol{\omega}$.

For $t = 0, 1, 2\cdots$ **do**

a. **Sampling**
Execute action u_t, observe next state x_{t+1} and reward r_{t+1}, and decide next action $u_{t+1} \sim \pi(u_{t+1}|x_{t+1})$.

b. **Critic update**
- Forget TD estimator parameter
  $\boldsymbol{\omega} := \beta\boldsymbol{\omega}$;
- Compute TD-errors
  $\delta_v := r_{t+1} + \gamma\hat{V}(x_{t+1}) - \hat{V}(x_t)$
  $\delta_\omega := \delta_v - f^\pi_\omega(x_t,u_t)$;
  [$\delta_b := \delta_\omega - \hat{B}(x_t,\boldsymbol{\omega}) + \gamma\lambda_b\hat{B}(x_{t+1},\boldsymbol{\omega})$;]
- Update critic eligibilities
  $\boldsymbol{z}_v := \gamma\lambda_v\boldsymbol{z}_v + \nabla_v\hat{V}(x_t)$;
  $\boldsymbol{z}_\omega := \gamma\lambda_\omega\boldsymbol{z}_\omega + \nabla_\theta \ln \pi_\theta(u_t|x_t)$;
  [$\boldsymbol{z}_b := \gamma\lambda_b\boldsymbol{z}_b + \nabla_\theta \ln \pi_\theta(u_t|x_t)^2\nabla_b\hat{B}(x_t,\boldsymbol{\omega})$;]
- Update value function parameter[s]
  $\boldsymbol{v} := \boldsymbol{v} + \alpha_v\delta_v\boldsymbol{z}_v$;
  [$\boldsymbol{b} := \boldsymbol{b} + \alpha_b\delta_b\boldsymbol{z}_b$;]
- Update NPG estimator parameter
  $\boldsymbol{\omega} := \boldsymbol{\omega} + \alpha_\omega\delta_\omega\boldsymbol{z}_\omega$;

c. **Actor update**
$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_\theta\boldsymbol{\omega}$;

Figure 1: The [extended] NTD algorithm; The normal NTD algorithm is specified by skipping the contents in the square brackets. In the case of the extended NTD algorithm, the square bracket symbols are ignored.

5 Numerical Experiments

5.1 MDP with inadequate policy

The 3-state 2-action MDP is modified from Baxter *et.al.*[11] at the points of a state-transition probability and a parameterization of policy. We omit the detail task setting because of space limitations. Under this policy parameterization, the condition of proposition 1 can not be satisfied. Thus, even when $\hat{\boldsymbol{\omega}}$ is equal to the exact NPG, the state value could not be the optimal baseline function by proposition 2. Fig.2 indicates that the extended NTD suppresses the variance of the NPG estimates than the normal NTD.

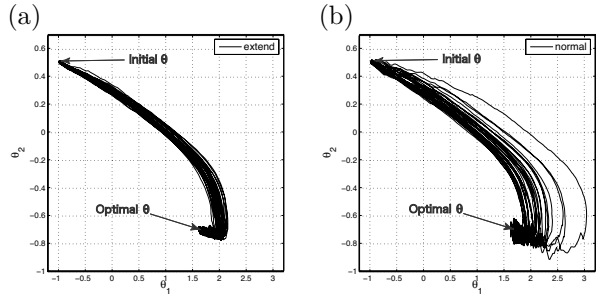

Figure 2: MDP; phase plane analyses; policy parameter trajectories (a) the extended NTD, (b) the normal NTD.

5.2 Pendulum swing-up problem

This section gives the comparison between NTD algorithms and other policy gradient methods; NAC[4], Kimura Actor-Critic[12] in the same setting as [1].

The auxiliary function $B(x, \hat{\boldsymbol{\omega}})$ in the extended NTD is decomposed to two terms; $B(x, \hat{\boldsymbol{\omega}}) = b_1(x) - b_2(x, \hat{\boldsymbol{\omega}})$, where

$$b_1(x) = \frac{\mathrm{E}^{\pi}\{\nabla_\theta \ln \pi_\theta(u|x)^2(\hat{Q}(x,u) - \hat{V}(x))\}}{\mathrm{E}^{\pi}\{\nabla_\theta \ln \pi_\theta(u|x)^2\}},$$

$$b_2(x, \hat{\boldsymbol{\omega}}) = \mathrm{E}^{\pi}\{\nabla_\theta \ln \pi_\theta(u|x)^2 f_{\hat{\omega}}^{\pi}(x,u)\} / \mathrm{E}^{\pi}\{\nabla_\theta \ln \pi_\theta(u|x)^2\}.$$

When we use the Gaussian distribution policy [1], while $b_1(x)$ has to be estimated, $b_2(x, \hat{\boldsymbol{\omega}})$ could be solved analytically: $\boldsymbol{b}_\mu(x) \equiv \nabla_\theta \mu_\theta(x)$, $\boldsymbol{b}_\sigma(x) \equiv \nabla_\theta \sigma_\theta(x)$,

$$b_2(x, \hat{\boldsymbol{\omega}}) = \frac{(2\boldsymbol{b}_\mu^\top \boldsymbol{b}_\mu \boldsymbol{b}_\sigma^\top + 4\boldsymbol{b}_\mu^\top \boldsymbol{b}_\sigma \boldsymbol{b}_\mu^\top + 8\boldsymbol{b}_\sigma^\top \boldsymbol{b}_\sigma \boldsymbol{b}_\sigma^\top)\hat{\boldsymbol{\omega}}}{\sigma \boldsymbol{b}_\mu^\top \boldsymbol{b}_\mu + 2\sigma \boldsymbol{b}_\sigma^\top \boldsymbol{b}_\sigma}.$$

Fig.3 showed that the extended NTD algorithm works better than the other PG algorithms.

6 Summary

This paper presented that the state value function could become a valid baseline function with an appropriate policy parameterization for a task. For the case where the state value function diverges from the optimal baseline function, the extended version of the NTD algorithm was proposed, which compensates for the differences between the state value and the optimal baseline by introducing the auxiliary function. Additional theoretical and experimental analyses are necessary to further understand the properties and the effectiveness of the NTD algorithm.

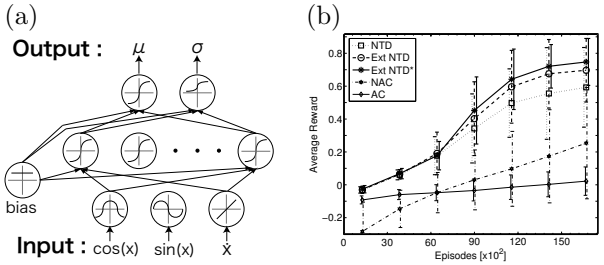

Figure 3: Swing-up pendulum problem; (a) The policy is a three-layer neural network with 10 hidden units. (b) The average rewards over 30 independent runs. Comparison among PGs under the improper RBF setup, [5× 5], for the state value estimation. Extended NTD* is the alternative algorithm computing b_1 analytically.

References

[1] T. Morimura, E. Uchibe, and K. Doya. Utilizing natural gradient in temporal difference reinforcement learning with eligibility traces. In *International Symposium on Information Geometry and its Applications*, 2005.

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

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

[4] J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *IEEE-RAS International Conference on Humanoid Robots*, 2003.

[5] E. Greensmith, P. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

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

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

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

[9] D. Bagnell and J. Schneider. Covariant policy search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, July 2003.

[10] J. Peters and S. Schaal. Policy gradient methods for robotics. In *IEEE International Conference on Intelligent Robots and Systems*, 2006.

[11] J. Baxter, P. Bartlett, and Lex Weaver. Experiments with infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.

[12] 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*, pages 278–286, 1998.