NAIST-IS-DD0561034

Doctoral Dissertation

Efficient Task-independent Reinforcement Learning Methods based on Policy Gradient

Tetsuro Morimura

March , 2008

Department of Bioinformatics and Genomics
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Tetsuro Morimura

Thesis Committee:

Professor Shin Ishii	(Supervisor)
Professor Kenji Sugimoto	(Co-supervisor)
Professor Kenji Doya	(Co-supervisor)
Associate Professor Junichiro Yoshimoto	(Co-supervisor)
Professor Tsukasa Ogasawara	(Co-supervisor)

Efficient Task-independent Reinforcement Learning Methods based on Policy Gradient*

Tetsuro Morimura

Abstract

This dissertation presents research results about decision making rules in an uncertain environment, called *reinforcement learning* (RL). We focus on RL methods based on gradient descent, so-called *policy gradient reinforcement learning* (PGRL), and give efficient task-independent algorithms through mathematical studies and numerical experiments.

PGRL attempts to find the policy as the decision-making rule that locally maximize the objective function such as the average or temporal discounted reward. It is performed by estimating the gradient of the objective function with respect to the policy parameter from the experienced system trajectories of states, actions, and rewards, and improving the policy parameter on the basis of gradient descent. As long as the policy is parameterized appropriately, PGRL can be instantly implemented to Markov decision process (MDP) without the explicit knowledge about the environment and the learning agent. Moreover, since it is possible to treat the parameter controlling the randomness of the policy as the policy parameters, PGRL can obtain the appropriate stochastic policy and be applied to partially observable MDP (POMDP). Therefore, PGRL is expected to be applied to various fields and draws much attention. However, there are three difficulties at least for PGRL to come into practice use:

1) tendency of learning times to be huge amounts,

*Doctoral Dissertation, Department of Bioinformatics and Genomics, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561034, March , 2008.

2) hardship in setting of hand-tuning parameters as meta-parameters,

3) hardship in parameterizing of appropriate policies.

Although there are many studies for the above problems, most these studies suppose some specific tasks and use the prior knowledge about the tasks. It indicates that the methods of these studies could lack versatility. Therefore, it requires such improvements of the PGRL algorithm as keeping intact about the standard framework of RL, i.e., task-independent modifications rather than task-dependent. In this thesis, in order to resolve the above problems we probe efficient task-independent PGRL algorithms.

For the problem 1), we focus on the structure of the learning (policy) parameter space, in order to keep away plateau phenomenon where the learning curve is almost flat in a long period, and study the natural gradient proposed by Amari. It takes into consideration the sensitivity of each element of the policy parameter and the correlation between the elements, to probability distribution of MDP. Firstly, we develop the natural policy gradient (NPG) method with the Riemannian metric matrix proposed by Kakade, to an efficient algorithm without a matrix inversion. Next, new NPGs based on valid Riemannian metrics are proposed by utilizing the state-stationary distribution. These gradients take into account the changes in the state-action joint distributions for improving the policy parameter, while kakade's NPG takes into account only changes in the action distribution and omits changes in the state distribution.

For the problem 2), we focus on the meta-parameter that controls the temporal discounting for the cumulative rewards, so-called *forgetting* or *discounting* factor γ, since the usefull methods have not been proposed for this parameter so far. In ordinary PG methods (Kimura and Kobayashi, 1998; Baxter and Bartlett, 2001), the forgetting factor γ controls the bias-variance trade-off of the estimation for the average reward gradient with respect to the policy parameter. This is because the ordinary PG methods omit a term regarding the derivative of the state-stationary distribution, in order to estimate the gradients. By deriving a method to estimate the derivative of the stationary distribution, we develop γ-free PGRL algorithms.

For the problem 3), a criterion is derived, in order to judge whether or not the current parameterization of the policy is sufficient for the achievement of task objective. If the criterion converges to zero, the policy parameterization is

sufficient.

Keywords:

Reinforcement learning, Markov decision process, Policy gradient method, Natural gradient method, derivative of state-stationary distribution.

方策勾配に基づく効率の良い課題非依存な強化学習法*

森村 哲郎

内容梗概

方策勾配強化学習法は，エージェントが環境と相互作用する際に得られる報酬の平均値を目的関数とし，この目的関数を局所最大化する方策（行動則）の獲得を目指した方策探索法で，方策パラメータを目的関数の勾配により逐次更新することで実現される．方策さえ適切にパラメータ化すればエージェントや環境に関する知識を必要とせず直ちにマルコフ決定過程（Markov Decision Process; MDP）に実装可能であり，またランダム性を制御するパラメータを方策パラメータに含めることで確率的な方策の獲得も可能なため部分観測マルコフ決定過程にも適用可能である．そのため方策勾配強化学習法は様々な分野への応用が期待され，近年注目を集めている．しかしながら，実用化に向けて解決すべき問題が少なくとも 3 つ挙げられる；

1) 学習所要時間が膨大になり易い,

2) 実験者が事前に与えるパラメータ（メタパラメータ）の設定が困難,

3) 適切な方策のパラメータ化が困難.

これらに対する先行研究は多々あるが，そのほとんどは特定の課題を想定しており，課題の事前知識を利用したものであったため，汎用性に欠けていた．よって標準的な強化学習の枠組みに手を加えない，つまり課題に依存しないような方策勾配アルゴリズムの改良が望まれる．そこで，本研究では上記問題の解決を目指して，効率の良い方策勾配強化学習アルゴリズムを数理的に探った．

問題 1) に対しては，特にプラトー（学習の停滞期間）に注目して，MDP の確率分布に対して各方策パラメータの敏感さの相違やその相関を考慮した自然勾配法の研究を行った．そこでは初めに，Kakade の提案した自然方策勾配 (NPG) の逆行列演算を必要としない適応的な方法で推定する自然時間差分学習法（NTD アルゴリズム）を提案した．これは状態行動を条件とする状態価値関数の時間的

*奈良先端科学技術大学院大学 情報科学研究科 情報生命科学専攻 博士論文, NAIST-IS-DD0561034, 2008 年 3 月 日.

差分（TD）の期待値がアドバンテージ関数と一致する事実を利用して，一般に推定が困難であるアドバンテージ関数の代わりにTDを最小二乗近似することでNPGを推定し，方策を更新する勾配法である．振り子振上げ課題等に適用した数値実験により提案法の有効性を示した．次に，NTDアルゴリズムに従って推定されるNPGの分散に関して理論的に解析し，その分散の上限を最小にするようにベースライン関数を補正する補助関数を導入した拡張型NTDアルゴリズムを提案した．数値実験により従来法に比べ効率よくNPGの推定が可能であることを確認した．さらに自然（方策）勾配で必要とされるリーマン計量行列についても解析し，最適な方策への収束を遅くしている理由を学習すべきパラメータ空間の構造の性質から考察して新しいNPGを導出した．従来用いられてきたKakadeのリーマン計量行列は方策のパラメータ摂動による行動の確率分布変化だけを考慮した計量行列であったのに対して，提案するNPGで用いるリーマン計量行列は行動の分布同様に方策の影響を受ける状態の分布までもを考慮したものになっている．そして数値実験より，特に状態数が多い場合でもプラトーに陥らず有効に働くことを示した．

問題2) に対しては、メタパラメータの中でもこれまで有効な調節法が提案されていない積算報酬の割引率に関する研究を行った．一般の方策勾配法により推定される方策パラメータに関する平均報酬の偏微分値は，状態の定常分布の偏微分の計算が困難であったため，その偏微分に関する項を無視したものであった．この影響（推定値の偏り）は割引率を1に近づければ減少するが，一方で分散は大きくなってしまう．つまり，割引率に関して偏り・分散のトレードオフ問題があった．そこで本研究では，逆方向マルコフ連鎖の性質を利用して定常分布の偏微分を推定する方法を導出し，割引率に依存しない新しい方策勾配法を提案した．割引率の設定が困難なMDPに適用した数値実験により提案法の有用性を示した．

問題3) に対しては、上記の拡張型NTDアルゴリズムにおける理論的解析結果「適切にパラメータ化された方策であれば補助関数が0に収束する」ことを利用した方策の自動パラメタライズ法を考案した．一例として方策が三層パーセプトロンにより表現される場合に，その隠れ層の素子の数を課題に応じて調節できるかを検証した．数値実験により適切な隠れ素子の数を持つパーセプトロンが獲得されることを確認した．

キーワード

強化学習，マルコフ決定過程, 方策勾配法, 自然勾配法，状態定常分布の勾配.

Contents

List of Figures

List of Tables

Acknowledgements

情報科学とは程遠い分野で大学院生活を送っていたある時，本当に学び研究したいことはこれだという思いにさせてくださり，実際にこの世界に飛び込ませてくださった川人光男教授および銅谷賢治教授に深く感謝いたします．研究を進めるにあたり，熱心にご指導くださり，様々なアイデアを提案して頂いた銅谷教授および沖縄大学院大学先行的事業計算神経プロジェクトの内部英治博士に深く感謝いたします．内部博士には昼夜土日を問わず，様々なことに相談にのって頂き，沖縄という慣れない地で右も左もわからない私にとって大変心強い存在でした．ここに感謝の意を表します．また研究輪講や機械学習輪講等で様々なご意見を頂き，研究に関しても鋭い助言を頂いた吉本潤一郎准教授および沖縄大学院大学先行的事業計算神経プロジェクトの方々に大変感謝いたします．大学院生である自分ひとりの力だけでは，決して研究を行うことはできませんでした．研究室にいくと，私の机にはパソコンがあり，その先には，大規模計算機実験を可能にしてくれた研究室専用のクラスターがありました．それらIT環境を安心して利用することができたのも，同プロジェクトの稲嶺安洋さんと井上智文さんのおかげです．ここに改めてお礼申し上げます．また，日々の大学院生活を安心して過ごすことが出来たのも，事務手続き等のサポートを実施していただいた安里恵美子さんと永野いずみさん，上原智佳子さんのおかげです．大変お世話になりました．また，日々，頭を使って疲れてしまった体をリフレッシュさせるためにはスポーツが大切だということを教えてくださった伊藤真博士，宮崎勝彦博士，宮崎佳代子博士，佐藤尚博士に心よりお礼申し上げます．バドミントンやゴルフ，トランポリンを通じて，粘り強く，そしてあきらめずにコツコツと楽しく研究をする力を学ぶことができたと思います．また，スポーツだけでなく音楽好きの自分にとって，Stefan Elfwing博士と新貝敦さんと音楽のお話をできたことは大変楽しい時間でありました．音楽に力をもらい，投げ出さずに研究に取り組むことができたのも，お二人のおかげです．ありがとうございました．また，将来への不安や社会人がどのようなものかもわからずに就職活動していた自分に，たくさんのアドバイスをくださった星野真人博士に心より御礼申し上げます．そして論理生命学講座の方々，ATR脳情報研究所の方々に感謝いたします．特に松原崇充君や藤原祐介君，柴田和久君，林卓治君，上岡拓未君，大塚誠君，中野高志君，吉田岳彦君，Alan Rodorigues君，木村慎志君は研究仲間としてだけでなく，良き友人としても楽しい時間を過ごさせて頂きました．ここに改めて感謝いたします．また博士課程への進学を認め，いかなる時も支えとなってくれた両親や家族また友人

達にここに改めて深く感謝いたします．最後に，論文審査を引き受けて下さった石井信教授，杉本謙二教授，銅谷賢治教授，吉本潤一郎准教授，小笠原司教授に改めてお礼申し上げます．

Chapter 1

Introduction

This dissertation presents research results about learning of decision-making rules in an uncertain environment, called *reinforcement learning* (RL). We focus on and study RL based on gradient descent, and give efficient task-independent algorithms through mathematical studies and numerical experiments.

1.1 Overview of Reinforcement Learning

Reinforcement learning (RL) is a theoretical scheme for learning the decision making rule, so-called a "*policy*", by which an agent or a system decides and executes an action corresponding to an observed state (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998). The criterion of learning is an average or a temporal discounted cumulation of observed immediate rewards. In an RL framework, namely, the agent attempts to maximize a cumulative reward by interacting with environment described as a Markov decision process (MDP). Therefore, if "what the agent should achieve" is induced to the (immediate) reward, the agent learns "how" autonomously in the RL framework. This would be one of the great advantages of RL because it is often intractable for engineers to design the "how" corresponding to all possible situations in engineering fields, e.g., the game agents for backgammon (Tesauro, 1995), tetris (Bertsekas and Tsitsiklis, 1996), etc., and robot controllers for RoboCup soccer (Stone and Veloso, 1999), helicopter flight (Abbeel et al., 2007), etc.

There are various methods for RL. These are categorized to two classes: (I) model-based RL and (II) model-free RL, where the model is the prior knowledge of environment, i.e., the functions about state transitions by the agent taking an action and the immediate rewards. Model-based RL utilizes the model to maximize a cumulative reward , implemented by dynamic programming (Bertsekas, 1995), Dyna-Q (Sutton and Barto, 1998), and so on. Accordingly, in the model-based RL framework, the agent has also to learn the model from data when the model is unknown. While model-based RL is tackled by various ways (Dearden et al., 1999; Brafman and Tennenholtz, 2003; Strehl and Littman, 2005; Poupart et al., 2006) and would be an important field in recent years, it is not included in this thesis.

Model-free RL does not use the model as the prior knowledge of the environment and can be roughly categorized to two classes: (i) the value update based RL and (ii) the direct policy-optimization based RL. In the value-update based RL, the policy is not explicitly represented by adjustable policy parameters, but it is implicitly represented by value functions that approximate expectations of (discounted) cumulative rewards from a state or a state-action pair. The methods of this type of RL attempt to find a good policy through the update of the value function, e.g., Q-learning, SARSA learning (Sutton and Barto, 1998). On the other hand, the policy is explicitly represented by adjustable policy parameters in the direct policy-optimization based RL. This RL optimizes directly the policy parameter to maximize the objective function as the cumulative reward. Most methods of this RL are based on gradient descent scheme and are called *Policy Gradient Reinforcement Learning* (PGRL) or merely *policy gradient* (PG) methods, implemented by REINFORCE (Williams, 1992), GPOMDP (Baxter and Bartlett, 2001), or (natural) Actor-Critic algorithms (Sutton and Barto, 1998; Kimura and Kobayashi, 1998; Baird and Moore, 1999; Sutton et al., 2000; Konda and Tsitsiklis, 2003; Kakade, 2002; Peters et al., 2003). Comparing PGRL with value based RL, PGRL has advantages such that PGRL could be easily applied to the cases of a continuous state-action environment and optimize stochastic policy, while these are often hard for value-based RL. Consequently, PGRL is drawing much attention in recent years. However, PGRL has weekness that it often takes more time-steps to find the good policy than value based RL.

In this thesis, we focus on and study PGRL since it has potential for many engineering applications and is still a developing research topic as described above.

1.2 Motivation

While PGRL is expected to be applied various fields and draws much attention as described above, there are three difficulties at least for PGRL to come into practice use:

1) tendency of learning times to be huge amounts,

2) hardship in parameterizing of appropriate policies, and

3) hardship in setting of hand-tuning parameters as meta-parameters.

Although there are many studies to overcome the above problems, most of these studies suppose some certain tasks and use the prior knowledge about tasks (Ng et al., 1999; Ronsenstein and Barto, 2004; Bagnell et al., 2004). It indicates that the methods of these studies could lack versatility. Therefore, it requires such modifications of the PGRL algorithm as keeping intact about the standard framework of RL, i.e., task-independent modifications. In this thesis, in order to resolve the above problems we probe efficient task-independent PGRL algorithms.

For the first problem 1), there is a limitation of standard gradient descent algorithms to consume huge learning time, that the ordinary gradient of a fucntion does not necessarily indicate its steepest direction, because the parameters might not be expressed in orthonormal coordinates. In order to overcome this problem, Amari (1998) proposed the concept of natural gradient, and Kakade (2002) introduced it in policy gradient RL and proposed the "natural policy gradient" method (NPG). However, the drawbacks of their algorithms require the computation of the inverse of a matrix and the Riemannian metric matrix having effect of the NPG direction was heuristic. We present a new algorithm based on Kakade's NPG, *Natural policy gradient utilizing Temporal Differences* (NTD) algorithm, which estimates the natural policy gradient in an online manner without matrix inversion (Morimura et al., 2005), and also propose a new NPG based on a valid Riemannian metric matrix by utilizing the state-stationary distribution (Morimura et al., 2007b).

For the second problem 2), we derive a criterion to judge whether or not the current parameterization of the policy is sufficient for the achievement of task objective. When the criterion converges to zero, the policy parameterization is sufficient. We develop the auto-adjustmenting algorithm for the number of hidden units of a multi-layer perceptron used as the policy.

For the final problem 3), we focus on the meta-parameter that controls the temporal discounting for the cumulative rewards, so-called *forgetting* or *discounting* factor γ, since the usefull methods have not been proposed for this parameter so far. In ordinary PG methods (Kimura and Kobayashi, 1998; Baxter and Bartlett, 2001), the forgetting factor γ controls the bias-variance trade-off of the estimation for the average reward gradient with respect to the policy parameter. This is because the ordinary PG methods omit a term regarding the derivative of the state-stationary distribution, in order to estimate the gradients. By deriving a method to estimate the derivative of the stationary distribution, we develop γ-free PGRL algorithms.

1.3 Contents of dissertation

This dissertation is organized as follows. In chapter 2, we explain the basic framework of PGRL and the natural gradient as preliminaries. The following chapters are divided into two main branches.

Studies in the first branch do not utilize the derivative and tackles the problem 1) regarding the learning times by utilizing the Kakade's NPG and deriving a baseline adjustment function for variance reduction, and also tackles the problem 2) regarding the parameterization of the policy. These topics are included in chapter 3.

Studies in the second branch utilize the derivative of the stationary distribution. We first derive the method estimating the derivative and develop γ-free PGRL algorithms for the problem 3) regarding the forgetting factor γ as the meta-parameter in chapter 4. Next, we derive a new NPG based on a valid Riemannian metric matrix by utilizing the derivative in chapter 5.

In chapter 6, we conclude this dissertation.

Chapter 2

Preliminaries

2.1 Policy gradient reinforcement learning (PGRL)

We review the conventional reinforcement learning (RL) methods based on policy gradient (PG)—PGRL. PGRL is modeled on a discrete time Markov decision process (MDP) (Bertsekas, 1995; Sutton and Barto, 1998). It is defined by the quintuplet $(\mathcal{S}, \mathcal{A}, p, r, \pi)$, where $\mathcal{S} \ni s$, $\mathcal{A} \ni a$ are finite sets of states and actions, respectively. $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a state transition probability function of a current state $s_t \in \mathcal{S}$, a current action $a_t \in \mathcal{A}$ and a following state $s_{t+1} \in \mathcal{S}$ from a time step $t\,(\geq 0)$ to $t+1$, i.e., $p(s_{t+1}|s_t, a_t) \equiv \Pr(s_{t+1}|s_t, a_t)$, satisfying $\sum_{s_{t+1}\in\mathcal{S}} p(s_{t+1}|s_t, a_t) = 1$. $r : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [\mathcal{R}_{\min}, \mathcal{R}_{\max}]$ is a reward function of s_t, a_t, and s_{t+1} and is bounded below by $\mathcal{R}_{\min}$ and above by $\mathcal{R}_{\max}$, which defines an immediate reward r_{t+1} observed by a learning agent [1]. $\pi : \mathcal{S} \times \mathcal{A} \times \mathcal{R}^d \to [0, 1]$ is a function for an action probability given a state and a policy parameter $\boldsymbol{\theta} \in \mathcal{R}^d$, so-called a stochastic policy, i.e., $\pi(a_t|s_t; \boldsymbol{\theta}) \equiv \Pr(a_t|s_t, \boldsymbol{\theta})$, which defines the decision-making of a learning agent and is adjustable by learning of the policy parameter $\boldsymbol{\theta}$.

We assume that the policy $\pi(a|s; \boldsymbol{\theta})$ is differentiable with $\boldsymbol{\theta}$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ [2], and would notate $\pi_\theta(a|s)$ as $\pi(a|s; \boldsymbol{\theta})$ for simplicity. We also posit the following assumption:

[1] Even if $r(s_t, a_t, s_{t+1})$ is a random variable, all results of this thesis can be applied directly by replacing $r(s_t, a_t, s_{t+1})$ with $\mathbb{E}\{r(s_t, a_t, s_{t+1})|s_t, a_t, s_{t+1}\}$.

[2] $\|\nabla_\theta \ln \pi_\theta(a|s)\| < \infty$.

Assumption 1 *The Markov chain $M(\boldsymbol{\theta}) = \{\mathcal{S}, \mathcal{A}, p, \pi_\theta\}$ is ergodic (irreducible and aperiodic) for all policy parameters $\boldsymbol{\theta}$. Then, there exists a unique stationary state distribution $d^\pi(s) \equiv \Pr(s|M(\boldsymbol{\theta}))$, equated to the limiting distribution, which is independent of the initial state,*

$$d^\pi(s') = \lim_{t\to\infty} \Pr(S_t = s'|S_0 = s, M(\boldsymbol{\theta})), \quad {}^\forall s \in \mathcal{S}. \tag{2.1}$$

The stationary distribution satisfies the following balance equation

$$\begin{aligned} d^\pi(s') &= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} p(s'|s,a)\pi(a|s,\boldsymbol{\theta})d^\pi(s), \qquad (2.2)\\ &\equiv \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} p_{M(\boldsymbol{\theta})}(s',a|s)d^\pi(s), \end{aligned}$$

where $p_{M(\boldsymbol{\theta})}(s',a|s) = p(s'|s,a)\pi(a|s;\boldsymbol{\theta})$. The following equation instantly holds (Bertsekas, 1995),

$$d^\pi(s') = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr(S_t = s'|S_0 = s, M(\boldsymbol{\theta})), \quad {}^\forall s \in \mathcal{S}. \tag{2.3}$$

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

$$R(\boldsymbol{\theta}) \equiv \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{M(\boldsymbol{\theta})} \left\{ \sum_{t=1}^{T} r_t \middle| s_0 \right\}, \tag{2.4}$$

where $\mathbb{E}_{M(\boldsymbol{\theta})}$ denotes the expectation over the Markov chain $M(\boldsymbol{\theta})$. It is noted that, under Assumption 1, the average reward is independent of the initial state s_0 and can be shown to be equal (Bertsekas, 1995):

$$\begin{aligned} R(\boldsymbol{\theta}) &= \mathbb{E}_{M(\boldsymbol{\theta})} \{r(s,a,s')\} \qquad (2.5)\\ &= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s'|s,a)r(s,a,s') \qquad (2.6)\\ &= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\bar{r}(s,a), \end{aligned}$$

where $\bar{r}(s,a) \equiv \sum_{s' \in \mathcal{S}} p(s'|s,a) r(s,a,s')$ does not depend on the policy parameter $\boldsymbol{\theta}$. Accordingly, the derivative of the average reward with respect to the policy parameter $\boldsymbol{\theta}$, which is often referred as the policy gradient (PG) for short,

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

where $\top$ denotes transpose, is calculated to

$$\begin{aligned} \nabla_{\theta} R(\boldsymbol{\theta}) &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \nabla_{\theta} (d^{\pi}(s) \pi_{\theta}(a|s)) \bar{r}(s,a) && (2.7) \\ &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d^{\pi}(s) \pi_{\theta}(a|s) \left(\nabla_{\theta} \ln \pi_{\theta}(a|s) + \nabla_{\theta} \ln d^{\pi}(s) \right) \bar{r}(s,a). && (2.8) \end{aligned}$$

The ordinary policy gradient RL algorithms update the policy parameter $\boldsymbol{\theta}$ in the direction of the ordinary gradient of the average reward, $\nabla_{\theta} R(\boldsymbol{\theta})$, with the sufficient small learning rate α:

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha \nabla_{\theta} R(\boldsymbol{\theta}),$$

where := denotes the the right-to-left substitution. Similarly, the natural policy gradient RL algorithms update $\boldsymbol{\theta}$ in the direction of the natural gradient of the average reward, $\widetilde{\nabla}_{\boldsymbol{\theta}} R(\boldsymbol{\theta})$, which is introduced in the following section 2.2.

As the derivation of the log stationary state distribution $\nabla_{\theta} \ln d^{\pi}(s)$ is non-trivial, the conventional PG algorithms (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998) utilize an alternative representation of the PG (see appendix for this derivation)

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} & d^{\pi}(x) \pi_{\theta}(a|s) \nabla_{\theta} \ln \pi_{\theta}(a|s) Q^{\pi}_{\gamma}(s,a) \\ & + (1-\gamma) \sum_{s \in \mathcal{S}} d^{\pi}(x) \nabla_{\theta} \ln d^{\pi}(s) V^{\pi}_{\gamma}(x), \qquad (2.9) \end{aligned}$$

where

$$Q^{\pi}_{\gamma}(s_t, a_t) \equiv \lim_{K \to \infty} \mathbb{E}_{M(\boldsymbol{\theta})} \left\{ \sum_{k=1}^{K} \gamma^{k-1} r_{t+k} | s_t, a_t \right\}$$

is an action value function and

$$V^{\pi}_{\gamma}(s_t) \equiv \lim_{K \to \infty} \mathbb{E}_{M(\boldsymbol{\theta})} \left\{ \sum_{k=1}^{K} \gamma^{k-1} r_{t+k} | s_t \right\}$$

is a state value function with discount factor $\gamma \in [0, 1)$ (Sutton and Barto, 1998). Since the contribution of the second term of Eq.2.9 becomes smaller as γ approaches 1 (Baxter and Bartlett, 2001), the conventional algorithms (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998) approximate the PG only from the first term by taking $\gamma \approx 1$ as a biased PG, i.e.,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) &\approx \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(x)\pi_{\theta}(a|s)\nabla_{\theta}\ln \pi_{\theta}(a|s) Q^{\pi}_{\gamma}(s,a), \qquad 0 << \gamma < 1 \qquad (2.10)\\ &\equiv \nabla^{\gamma}_{\theta} R(\boldsymbol{\theta}).\end{aligned}$$

The dependence on γ of the biased PG is explained in the following lemma:

Lemma 1 *We define ε by*

$$\varepsilon \equiv \frac{(1-\gamma)\, ||\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(s)\nabla_{\theta}\ln d^{\pi}(s) V^{\pi,\gamma}(s)||}{||\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\nabla_{\theta}\ln \pi_{\theta}(a|s) Q^{\pi,\gamma}(s,a)||}, \qquad (2.11)$$

where $||\boldsymbol{c}||$ is Euclidean norm of vector $\boldsymbol{c}$. Then, an angle between $\nabla^{\gamma}_{\theta} R(\boldsymbol{\theta})$ and the true gradient $\nabla_{\theta} R(\boldsymbol{\theta})$ is bounded by $cos^{-1}(\frac{1-\varepsilon}{1+\varepsilon})$. In the limit $\gamma \to 1$, ε is equal to zero; then, the biased policy gradient becomes the true policy gradient, i.e., $\nabla^{\gamma}_{\theta} R(\boldsymbol{\theta}) = \nabla_{\theta} R(\boldsymbol{\theta})$.

Proof: see the appendix 1.2.
Since $\nabla_{\theta}\ln d^{\pi}(s)$ can be estimated by the method proposed in chapter 4 or Morimura et al. (2007b), ε in Eq.2.11 can also be estimated. Thus, lemma 1 would be useful in order to adapt γ, although Baxter and Bartlett (2001) and Kakade (2001) provide other relations between γ and the biased PG with regard to the second eigenvalue of the state transition matrix.

Although the bias introduced by this omission becomes smaller as γ is close to 1, the variance of the estimate becomes larger. In chapter 3, we discuss PG algorithms computing the derivative of the average reward based on eq.2.10, which ignores the derivative of the stationary distribution. In chapter 4, we propose an alternative approach, which estimates the log stationary distribution derivative (LSD), $\nabla_{\theta}\ln d^{\pi}(s)$, and uses eq.2.8 to compute the derivative of the average reward. A marked feature is that we do not need to learn the value function, and thus, the algorithm is free from the bias-variance trade-off in the choice of the forgetting (or discount) factor γ.

2.2 Natural gradient

As mentioned in chapter 1, the ordinary gradient (derivative) of a fucntion does not necessarily correspond to its steepest direction if its parameters are not expressed in orthonormal coordinates in terms of a manifold defined by the function. Therefore, to solve the problem, we consider the application of the natural gradient (Amari, 1998), which can represent the steepest descent direction in this case. In this section, we introduce the background of the natural gradient (NG) and the natural policy gradient (NPG) as the NG for the PG.

In a Riemannian manifold of a parameter $\boldsymbol{a}$, the steepest descent direction of a function $g(\boldsymbol{a})$ is expressed as

$$\widetilde{\nabla}_{\boldsymbol{G},\boldsymbol{a}} g(\boldsymbol{a}) = \boldsymbol{G}^{-1}(\boldsymbol{a}) \nabla_{a} g(\boldsymbol{a}),$$

where $\boldsymbol{G}(\boldsymbol{a})$ is the Riemannian metric matrix of $\boldsymbol{a}$, which is defined by the Fisher information matrix in the case that the parameter space of $\boldsymbol{a}$ is in a statistical model, and $\widetilde{\nabla}_{a} g(\boldsymbol{a})$ is called the natural gradient. The Fisher information matrix is a unique metric matrix of the second-order Taylor expansion of the KL-divergence on a fixed probability distribution. When a different statistical model or a probability distribution is considered, obviously, the Fisher information matrix varies and the direction of the NG has to vary.

For NPG (the application of the NG to PGRL), it should be discussed what statistical model or probability distribution on MDP is appropriate to the basis of the Riemannian metric matrix. While we provide some answers about above question in chapter 5, we propose efficient NPG algorithm based on the Riemannian metric matrix proposed by Kakade (2002) in chapter 3. In chapter 5, we derive a valid Riemannian metric matrix for PGRL and propose a new NPG, which utilizes LSD.

Chapter 3

A Natural Policy Gradient on Kakade's Riemannian Metric

Since most previous algorithms which implement the natural policy gradient (NPG) on Kakade's Riemannian metric matrix (Kakade, 2002; Peters et al., 2003; Mori et al., 2005), use matrix inversion, they suffer from numerical instability and high computational costs. In section 3.2, we propose a novel NPG estimation method without matrix inversion by regressing the temporal difference (TD) reward prediction errors by using a set of basis functions given by the parameterization of the policy. We also show that the bias in the gradient estimate can be reduced by employing "eligibility traces" in the TD regression. The proposed algorithm, the natural policy gradient utilizing the temporal differences (NTD) algorithm, is applied to a simple Markov decision problem and a more challenging nonlinear pendulum-control problem to demonstrate its effectiveness.

In section 3.3, we discuss the baseline function for the NPG estimate based on NTD algorithm with respect to the variance and show a condition that an optimal baseline function reducing the variance is equivalent to the state value function. Because the state value could be much different from the optimal baseline outside of the condition, an extended version of the NTD algorithm is proposed for such cases. It introduces an auxiliary function 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.

In section 3.4, we discuss the problem what policy parameterizations are appropriate for some tasks, and propose the average absolute value of the auxiliary function to adjust the baseline as a criterion to judge whether or not the current parameterization of the policy is sufficient for the achievement of task objective. An auto-adjustmenting algorithm for the number of hidden units of a multi-layer perceptron used as the policy is developed by the fact that the criterion being zero means the policy parameterization is sufficient.

It must be noted that, in this chapter, because we deal only with the *biased* PGs and the *discounted* value functions, we omit the term *biased*[1] and *discounted*, respectively. For instance, when we discuss about the bias of an estimated PG, we imply the bias from the biased PG to the estimate.

3.1 Definition of Kakade's NPG

Kakade (2002) supposed that the Fisher information matrix of RL is the average of $\boldsymbol{F}_a(s, \boldsymbol{\theta})$ weighted by the stationary state distribution, $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta}) \equiv \sum_{a \in \mathcal{A}} d^\pi(s) \boldsymbol{F}_a(s, \boldsymbol{\theta})$, and then showed (see appendix for the derivation)

$$\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d^\pi(s) \pi_\theta(a|s) \nabla_\theta \ln \pi_\theta(a|s) f(s, a; \boldsymbol{w}) = \boldsymbol{w}, \tag{3.1}$$

where $f(s, a; \boldsymbol{\omega}) \equiv \nabla_\theta \ln \pi_\theta(a|s)^\top \boldsymbol{\omega}$ is termed as the compatible function (Sutton et al., 2000). Peters et al. (2003), and Bagnell and Schneider (2003) independently proved that $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is equivalent to the scaled Fisher information matrix of the probability distribution of the system trajectories, $p(\boldsymbol{\xi}_T|\boldsymbol{\theta})$; $\boldsymbol{\xi}_T = (s_0, a_0, s_1, ..., a_{T-1}, s_T)^\top$, with respect to the policy parameter $\boldsymbol{\theta}$ with the limit $T \to \infty$, i.e., [2]

$$\overline{\boldsymbol{F}}_a(\boldsymbol{\theta}) = \lim_{T \to \infty} \frac{1}{T} \boldsymbol{F}_{\boldsymbol{\xi}_T}(\boldsymbol{\theta}). \tag{3.2}$$

Since the maximization of the average reward can be regarded as the optimization of the integration of rewards over the space of possible system trajectories, the scaled Fisher information matrix of the trajectory distribution could be one of the

[1] The bias from PG to a biased PG is discussed in Baxter and Bartlett (2001)

[2] See chapter 5 for the derivation of 3.2 and detailed discussions about the Riemannian metric and the Fisher information matrices for RL.

reasonable Riemannian metrics for RL [3]. The natural policy gradient as natural gradient of RL on Kakade's Riemannian metrix is

$$\widetilde{\nabla}_{\overline{\boldsymbol{F}}_a,\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1}\nabla_{\theta} R(\boldsymbol{\theta}). \tag{3.3}$$

We simplify $\widetilde{\nabla}_{\overline{\boldsymbol{F}}_a,\boldsymbol{\theta}}$ to $\widetilde{\nabla}_{\theta}$ from here in this chapter, because $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is the only Riemannian matrix used for NPG in this chapter.

However, the algorithms (Kakade, 2002; Peters et al., 2003) that implement the natural policy gradient require the computation of the matrix inversion. With this background, we present the *natural policy gradient utilizing the temporal differences (NTD)* algorithm that estimates the natural policy gradient in an online manner without matrix inversion.

3.2 Utilizing incremental temporal differences for natural actor-critic (NAC)—NTD algorithm

The actor-critic framework for NPG is called the natural actor-critic (NAC) (Peters et al., 2003). The *critic* estimates NPG $\hat{\boldsymbol{\omega}}$ and the *actor* executes the action drawn from the policy $\pi_\theta(a|s)$, 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. In the following sections, we show the original and extended NTD algorithms.

3.2.1 Organization of NAC and the NTD Algorithm

We first introduce the overall architecture of the NTD algorithm and then we explain how each component works. The NTD algorithm comprises three components[4], as shown in Figure 3.1. The first component is the value estimator that

[3] This Riemannian metric matrix takes into account only changes in the action distribution for improving the policy parameter and omits changes in the state distribution, which also depends on the policy in almost all cases. In chapter 5, we propose a new Riemannian metric considering the state distribution as well as the action distribution and derive a new natural policy gradient based on the metric.

[4] If it is regarded as an actor-critic model (Sutton and Barto, 1998), the value and NPG estimators configure the critic and the policy is the same as the actor.

estimates the state value function. This is performed by ordinary TD(λ) (Sutton, 1988) or LSTD(λ) (Bradtke and Barto, 1996; Boyan, 1999) learning. The second is the natural policy gradient (NPG) estimator. It is realized by regressing the temporal differences (TD) given by the first component with a linear function approximator comprising basis functions defined by policy parameterization and the weight vector. The final component is the policy, which is updated toward the direction of the NPG estimate given as the weight vector of the second component. We term this framework *the **N**atural policy gradient utilizing **T**emporal **D**ifferences –the **NTD** Algorithm.*

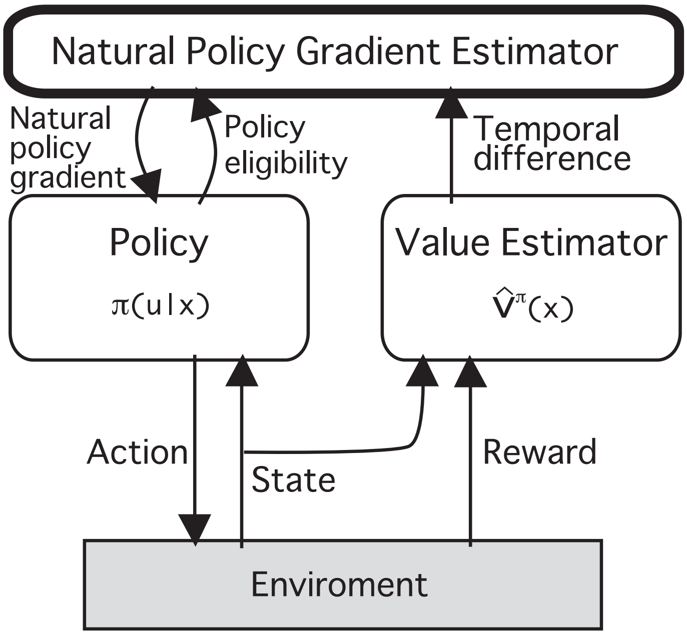


Figure 3.1. Architecture of the NTD algorithm.

In section 3.2.2, we show the following. When the compatible function $f(s, a; \boldsymbol{w})$ with respect to the policy parameterization (Sutton et al., 2000), which is a linear function with the weight $\boldsymbol{w}$ and the policy eligibility $\nabla_\theta \ln \pi_\theta(a|s)$ as the basis function, regresses the temporal difference of the state value function, the weight becomes an estimate of NPG. In section 3.2.3, we show that the weight can be an unbiased estimate of NPG if eligibility traces are applied to the TD regression at an eligibility decay rate of $\lambda = \gamma$ under a fixed policy.

3.2.2 Function Approximation to TD for PG

Function approximation for policy gradient

As Konda and Tsitsiklis (2003) and Sutton et al. (2000) have shown that the *unbiased* PG is expressed by the compatible function regressing a "differential cost function" defined as a solution of the Poisson equation, the compatible function $f(s,a;\boldsymbol{w}) \equiv \boldsymbol{w}^\top \nabla_\theta \ln \pi_\theta(a|s)$ can also be used to represent the PG defined by eq.2.10 as (see the appendix 2.2),

$$\begin{aligned}\nabla_\theta^\gamma R(\boldsymbol{\theta}) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\nabla_\theta \ln \pi_\theta(a|s) f(s,a;\boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)}) \qquad (3.4)\\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\psi}(s,a)\boldsymbol{\psi}(s,a)^\top \boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)},\end{aligned}$$

where $\boldsymbol{\psi}(s,a) \equiv \nabla_\theta \ln \pi_\theta(a|s)$ is termed the policy eligibility and $\boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)}$ is the weight that minimizes the mean square error between $Q^\pi(s,a) - b(s)$ and $f(s,a;\boldsymbol{w})$,

$$\epsilon(\boldsymbol{w}) \equiv \frac{1}{2}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\left\{Q^\pi(s,a) - b(s) - \boldsymbol{\psi}(s,a)^\top \boldsymbol{w}\right\}^2. \qquad (3.5)$$

Hereafter, $\boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)}$ will be abbreviated as $\boldsymbol{w}^*$ for simplicity. In this case,

$$\nabla_w \epsilon(\boldsymbol{w}^*) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\left\{Q^\pi(s,a) - b(s) - \boldsymbol{\psi}(s,a)^\top \boldsymbol{w}^*\right\}\boldsymbol{\psi}(s,a) = 0.$$

It is noted that $\boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)}$ remains unchanged by the choice of the baseline function $b(s)$ because $f(s,a;\boldsymbol{w})$ has zero mean for each state,

$$\sum_{a\in\mathcal{A}} \pi_\theta(a|s) f(s,a;\boldsymbol{w}) = \boldsymbol{w}^\top \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|s) = \boldsymbol{w}^\top \nabla_\theta \sum_{a\in\mathcal{A}} \pi_\theta(a|s) = 0, \qquad {}^\forall s \in \mathcal{S}, \qquad (3.6)$$

then Eq.3.5 is calculated as $\epsilon(\boldsymbol{w}) = 1/2 \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\left\{Q^\pi(s,a) - f(s,a;\boldsymbol{w})\right\}^2$. However, when the number of samples is finite, $b(s)$ affects the variance of the estimate of the regressor $f(s,a;\hat{\boldsymbol{w}}|_{Q^\pi(s,a)-b(s)})$ where $\hat{\boldsymbol{w}}|_{Q^\pi(s,a)-b(s)}$ is a weight regressed to the regressand "$Q^\pi(s,a) - b(s)$" with finite samples. Therefore, in practice, it is important to set $b(s)$ appropriately. Sutton et al. (2000) and Peters et al. (2003) suggest that a value of $b(s)$ satisfying

$$\sum_{a \in \mathcal{A}} \pi_\theta(a|s) \left(Q^\pi(s,a) - b(s)\right) = 0, \tag{3.7}$$

that is, $b(s) = V^\pi(s)$, is a better baseline function than $b(s) = 0$ because of the constraint of the compatible function, Eq.3.6 [5]. That may be supported by the following proposition

Proposition 1 *If the baseline function $b(s)$ is equal to the state value function $V^\pi(s)$, a residual sum of squares*

$$\mathrm{RSS}_f(Q^\pi(s,a) - b(s)) \equiv \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d^\pi(s) \pi_\theta(a|s) \left\{Q^\pi(s,a) - b(s) - f(s,a;\boldsymbol{w})\right\}^2$$

is minimized about $b(s)$ for any $\boldsymbol{w}$.

Proof: see the appendix 2.3.

Function approximation to TD as advantage function

When $b(s) = V^\pi(s)$, the regressand is equal to *the advantage function* $A^\pi(s,a) \equiv Q^\pi(s,a) - V^\pi(a)$ (Baird, 1993)[6]. It is noted that the advantage function cannot be learned by TD learning that uses $f(s,a;\boldsymbol{w})$ exclusively (Peters et al., 2003). Although there are some methods for learning, they are considerably difficult because they require an argmax operator or a matrix inversion computation (Baird, 1993; Dayan and Singh, 1996; Peters et al., 2003). Here, we present lemma 2 for the feasible construction of $f(s,a;\hat{\boldsymbol{w}}|_{A^\pi(s,a)})$, which is the same as that under $b(s) = V^\pi(s)$, by utilizing the TD of the state value function. The TD (also referred to as the TD error) is defined in the Bellman equation (Sutton and Barto, 1998),

$$\delta_t \equiv r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t).$$

[5] The detailed discussions regarding the baseline function are present in Greensmith et al. (2004) and Peters and Schaal (2006), which propose optimal baseline functions minimizing the bounds of variances, and Morimura et al. (2007a), which show that the state value function is equivalent to these optimal baseline functions when the policy parameterization is proper and $f(s,a;\hat{\boldsymbol{w}})$ converges to $f(s,a;\boldsymbol{w}^*)$.

[6] The advantage function provided by Baird (1993) is $A^\pi(s,a) \equiv Q^\pi(s,a) - \mathrm{argmax}_a Q^\pi(s,a)$.

In practice, δ_t is only used to update the value function and is discarded in each trial [7]. However, the basic concept of our algorithm is that the TD δ_t is considered as a function of the state and the action,

$$\delta^\pi(s_t, a_t) \equiv r(s_t, a_t, s_{t+1}) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t),$$

and $\delta^\pi(s, a)$ is considered as target function of the regression function $f(s, a; \boldsymbol{w})$. The TD $\delta^\pi(s, a)$ is a random variable because s_{t+1} is a random variable, except in the case of $p(s_{t+1}|s_t, a) = 1$, and the reward function $r(s_t, a_t, s_{t+1})$ may also be a random variable. It is noted that the expectation of the TD given s and a, $\langle \delta^\pi(s, a) \rangle$, does not necessarily become zero on the stochastic policy; this is applied to the derivation of proposition 2. Of course, the expectation of the TD given s is equal to zero.

Proposition 2 *The expectation of the TD of the state value function in the state-action space is equal to the advantage function,*

$$\mathbb{E}_{M(\boldsymbol{\theta})} \left\{ \delta^\pi(s, a) | s, a \right\} = A^\pi(s, a). \tag{3.8}$$

If $\mathrm{Var}_\pi(\delta^\pi(s, a)) = 0$*, the following equation holds:*

$$\delta^\pi(s, a) = A^\pi(s, a),$$

where the function $\mathrm{Var}_\pi(\delta^\pi(s, a))$ *is the average of the variance of* $\delta^\pi(s, a)$*, based on the state and action distribution,* $\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\mathbb{E}_{M(\boldsymbol{\theta})}\left\{ \left(\delta^\pi(s, a) - A^\pi(s, a)\right)^2 | s, a \right\}$.

Proof: see the appendix 2.4.

Lemma 2 (I) *Let regressions be performed with infinite number of samples from Markov chains* $M(\boldsymbol{\theta})$ *with an appropriate regression method. Then, the following equation holds:* $f(s, a; \hat{\boldsymbol{w}}^*|_{\delta^\pi(s,a)}) = f(s, a; \hat{\boldsymbol{w}}^*|_{A^\pi(s,a)}) = f(s, a; \boldsymbol{w}^*)$ [8].
(II) *Let regressions be performed with a finite number of samples from* $M(\boldsymbol{\theta})$.
(II-1) *If* $\mathrm{Var}_\pi(\delta^\pi(s, a)) = 0$*, the following equation holds:*

$$f(s, a; \hat{\boldsymbol{w}}|_{\delta^\pi(s,a)}) = f(s, a; \hat{\boldsymbol{w}}|_{A^\pi(s,a)}).$$

[7] The TD is also used for the policy updating in actor-critic RL, but it is also discarded in each trial (Kimura and Kobayashi, 1998).

[8] (I) means $f\left(s, a; \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)}\}\right) = f\left(s, a; \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{w}}|_{A^\pi(s,a)}\}\right) = f(s, a; \boldsymbol{w}^*)$

(II-2) *If* $\mathrm{Var}_\pi(\delta^\pi(s,a))$ *is sufficiently smaller than* $\mathrm{RSS}_f(A^\pi(s,a))$*, then,*

$$f(s,a;\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)}) \simeq f(s,a;\hat{\boldsymbol{w}}|_{A^\pi(s,a)}).$$

Proof: (I) is proved in the appendix 2.5, shown as both the regression functions, $f(s,a;\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)})$ and $f(s,a;\hat{\boldsymbol{w}}|_{A^\pi(s,a)})$, converge to $f(s,a;\boldsymbol{w}^*)$ with infinite samples. (II-1) is apparent, since $\delta^\pi(s,a) = A^\pi(s,a)$ holds by proposition 2 when $\mathrm{Var}_\pi(\delta^\pi(s,a)) = 0$. (II-2) is provided by (I), (II-1), and the following two things; First, the regression of the compatible function to $\delta^\pi(s,a)$ uses the state value function $V^\pi(s)$ as the baseline function, as in the case of $f(s,a;\hat{\boldsymbol{w}}|_{A^\pi(s,a)})$, which is apparent from the definition of $\delta^\pi(s,a)$. Second, the residual sum of squares of the compatible function regressed for $\delta^\pi(s,a)$ is larger than that for $A^\pi(s,a)$ only for $\mathrm{Var}_\pi(\delta^\pi(s,a))$,

$$\mathrm{RSS}_f(\delta^\pi(s,a)) = \mathrm{RSS}_f(A^\pi(s,a)) + \mathrm{Var}_\pi(\delta^\pi(s,a)), \tag{3.9}$$

which is derived in the appendix 2.5. That is, if $\mathrm{Var}_\pi(\delta^\pi(s,a))$ is sufficiently small, then $f(s,a;\hat{\boldsymbol{w}}|_{A^\pi(s,a)}) \simeq f(s,a;\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)})$ holds. □

Lemma 2 indicates that it is effective to use $\delta^\pi(s,a)$ as the regressand for the construction of $f(s,a;\boldsymbol{w}^*)$ under a small $\mathrm{Var}_\pi(\delta^\pi(s,a))$, as well as the case to use $A^\pi(s,a)$, which is hard to be estimated. Even if $\mathrm{Var}_\pi(\delta^\pi(s,a))$ is large, where the entropy about the state transition probability $p(s_{t+1}|s_t,a_t)$ is high and/or the the reward function has a large random noise, $\hat{\boldsymbol{w}}|_{\delta s,a}$ remains an unbiased estimate of $\boldsymbol{w}^*$. However, in this case, $A^\pi(s,a)$ is a better regressand than $\delta^\pi(s,a)$ because $\mathrm{RSS}_f(\delta^\pi(s,a))$ is much larger than $\mathrm{RSS}_f(A^\pi(s,a))$.

By applying lemma 1 and lemma 2, we obtain the convergence property of the NPG estimation with the TD with regard to the natural policy gradient.

Theorem 1 *Let* ε *defined in Eq.2.11 and* $\mathrm{Var}_\pi(\delta^\pi(s,a))$ *be sufficiently close to zero. Then, the natural policy gradient* $\widetilde{\nabla}_\theta R(\boldsymbol{\theta})$ *is approximated by the vector* $\hat{\boldsymbol{w}}|_{\delta(s,a)}$ *which is the weight of the compatible function regressed to the TD with finite samples from the Markov chains* $M(\boldsymbol{\theta})$*,*

$$\widetilde{\nabla}_\theta R(\boldsymbol{\theta}) \simeq \boldsymbol{w}|_{\delta(s,a)}.$$

Proof:

$$
\begin{aligned}
\widetilde{\nabla}_{\theta} R(\boldsymbol{\theta}) \simeq \widetilde{\nabla}^{\gamma}_{\theta} R(\boldsymbol{\theta}) &= \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1} \nabla^{\gamma}_{\theta} R(\boldsymbol{\theta}) && \text{[eq.2.10, lemma 1, \& eq.3.3]} \\
&= \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\nabla_{\theta}\ln \pi_{\theta}(a|s) f(s,a;\boldsymbol{w}^*) && \text{[eq.3.4]} \\
&\simeq \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\nabla_{\theta}\ln \pi_{\theta}(a|s) f(s,a;\hat{\boldsymbol{w}}|_{A^{\pi}(s,a)}) \\
& && \text{[proposition 1]} \\
&\simeq \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\nabla_{\theta}\ln \pi_{\theta}(a|s) f(s,a;\hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)}) && \text{[lemma 2]} \\
&= \hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)}, && \text{[eq.3.1]}
\end{aligned}
$$

□

It is noted that, if the number of samples is infinite, the fourth and fifth transformations of the above proof has an equality instead of a near equality $\simeq$, and then $\widetilde{\nabla}_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)}$ holds with an appropriate regression method. For simplicity, henceforth, we notate $\hat{\boldsymbol{w}}$ as the estimate of $\hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)}$.

3.2.3 Eligibility traces with value function estimates

When the state value function is known, the exact TD $\delta^{\pi}(s,a)$ is available and then the estimation of NPG, $\boldsymbol{w}$, on the NTD algorithm is reduced to a general supervised problem as a linear regression of the TD with the basis function $\nabla_{\theta}\ln \pi_{\theta}(a|s)$. Thus, many methods on supervised learning are available, e.g., the least squares and various gradient descent regressions. To compute the exact state value function analytically, it is necessary that the state transition probability and the reward function are known. However,the above situation is rare during actual tasks. In cases other than the above situation, a critical problem for the implementation of the NTD algorithm is to estimate $\boldsymbol{w}$ appropriately with a state value function estimate $\hat{V}^{\pi}(s)$, which would have estimation errors; that is, when a common supervised algorithm for the regression of "$\hat{\delta}(s_t,a_t) \equiv r_t + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)$" is applied, the NPG estimates $\hat{\boldsymbol{w}}$ would be biased. To solve the the problem, we propose regression algorithms using the eligibility trace of the policy.

Two algorithms using eligibility traces for the TD regression with an estimate $\hat{V}(s)$ are proposed: Algorithm 1 is based on a gradient descent algorithm like

TD(λ) (Sutton and Barto, 1998). Algorithm 2, which is shown in appendix, is based on a least squares algorithm like LSTD(λ) (Boyan, 1999). In these algorithms, := denotes the substitution of the back for the front. Both algorithms estimate NPG at time step t, by regarding the eligibility trace as

$$\boldsymbol{z}_t \equiv \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \nabla_\theta \ln \pi_\theta(a_k|s_k),$$

and the immediate error as

$$\varepsilon_t \equiv r_t + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t) - \hat{\boldsymbol{w}}^\top \Big(\nabla_\theta \ln \pi_\theta(a_t|s_t) - \iota \nabla_\theta \ln \pi_\theta(a_{t+1}|s_{t+1}) \Big), \quad (3.10)$$

where the eligibility decay rate $\lambda \in [0,1]$ and the $\iota \in \mathcal{R}$ are meta-parameters which are decided by hand. Although ι should be equal to $\gamma\lambda$, following the ordinary eligibility manner, it is a free parameter because of the property of Eq.3.6, $\sum_{a\in\mathcal{A}} \pi_\theta(a|s) f(s,a;\boldsymbol{w}) = 0$. However, it should set in $[0,\gamma\lambda]$. This is because, when $\iota = 0$, the immediate error ε_t does not have the randomness from the following time step $t+1$ about $f_{t+1} \equiv \hat{\boldsymbol{w}}^\top \nabla_\theta \ln \pi_\theta(a_{t+1}|s_{t+1})$ in Eq.3.10; however, $\Delta\hat{w}_t$ in eq.3.11 or eq.a-11 [9] has the randomness from the f_k of the following time steps $k \in \{t+1, ..., T\}$. When $\iota = \gamma\lambda$, the feature is the opposite of that mentioned above, and $\iota \in (0,\gamma\lambda)$ fills the gap between these limiting cases. We set $\iota = 0$ in all the numerical experiments in this study, because the differences between the numerical results obtained with various values of $\iota \in [0,1]$ are not significant. There are other meta-parameters in algorithm 1, based on the gradient descent: k is the interval for the update of $\hat{\boldsymbol{w}}$ and α, which would change in time steps (Bertsekas and Tsitsiklis, 1996), is the learning rate of $\hat{\boldsymbol{w}}$. The proposed algorithm has a nice property as the following theorem.

[9] Although $\lambda = 1$ in these equations, it is the same in $\lambda \in [0,1]$.

Algorithm 1 Estimation of NPG based on gradient descent

```
Given:
  • a policy $\pi_\theta(a|s)$.
  • the system trajectory and rewards by the policy, $\{s_0, a_0, r_1, ..., r_T, s_T, a_T\}$.
  • an estimated state value function $\hat{V}(s)$.
Initialize: $k$, $\gamma$, $\alpha$, $\lambda$, $\iota$, and $\hat{\boldsymbol{w}}$.
Set: $\Delta\boldsymbol{w} := \mathbf{0}$; $\boldsymbol{z} := \mathbf{0}$;
For $t = 0 : T-1$ do
  $\boldsymbol{z} := \gamma\lambda\boldsymbol{z} + \nabla_\theta \ln \pi_\theta(a_t|s_t)$;
  $\Delta\hat{\boldsymbol{w}} := \Delta\hat{\boldsymbol{w}} + \boldsymbol{z}\{r_{t+1} + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)$
                 $-\hat{\boldsymbol{w}}^\top(\nabla_\theta \ln \pi_\theta(a_t|s_t) - \iota\nabla_\theta \ln \pi_\theta(a_{t+1}|s_{t+1}))\}$;
  If $mod(t,k)^\dagger = 0$
    $\hat{\boldsymbol{w}} := \hat{\boldsymbol{w}} + \alpha\Delta\hat{\boldsymbol{w}}/k$;
    reset: $\Delta\hat{\boldsymbol{w}} := \mathbf{0}$; $\boldsymbol{z} := \mathbf{0}$;
  end
end
Return: $\hat{\boldsymbol{w}}$.
```

† *mod*(t, k) computes modulus of t after division by k.

Theorem 2 *Let the TD regression be conducted with a fixed policy and a state value estimate $\hat{V}(s)$. If the eligibility decay rate λ is equal to one, the NPG estimate is unbiased.*

Proof: We prove the theorem in the gradient descent case, algorithm 1, based on Kimura and Kobayashi (1998), while the proof in the least squares case, algorithm 2, is shown in the appendix 2.6. We denote $\boldsymbol{\psi}_t \equiv \nabla_\theta \ln \pi_\theta(a_t|s_t)$ and $\hat{V}_t \equiv \hat{V}(s_t)$

for simplicity. Then,

$$
\begin{aligned}
\langle \Delta \hat{\boldsymbol{w}} \rangle &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \varepsilon_t \boldsymbol{z}_t \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \Big[r_{t+1} + \gamma \hat{V}_{t+1} - \hat{V}_t - (\boldsymbol{\psi}_t - \iota \boldsymbol{\psi}_{t+1})^\top \hat{\boldsymbol{w}} \Big] \sum_{\tau=1}^{t} \gamma^{t-\tau} \boldsymbol{\psi}_\tau \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}_t \Big[\sum_{\tau=t}^{T-1} \gamma^{\tau-t} \Big\{ r_{\tau+1} + \gamma \hat{V}_{\tau+1} - \hat{V}_\tau - (\boldsymbol{\psi}_\tau - \iota \boldsymbol{\psi}_{\tau+1})^\top \hat{\boldsymbol{w}} \Big\} \Big] \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}_t \Bigg[\sum_{\tau=t}^{T-1} \gamma^{\tau-t} r_{\tau+1} + \gamma^{T-t} \hat{V}_T - \hat{V}_t \\
&\qquad\qquad - \Big(\boldsymbol{\psi}_t + \sum_{\tau=t}^{T-2} \gamma^{\tau-t} (\gamma - \iota) \boldsymbol{\psi}_{\tau+1} - \gamma^{T-1-t} \iota \boldsymbol{\psi}_T \Big)^\top \hat{\boldsymbol{w}} \Bigg] \qquad (3.11) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d^\pi(s) \pi_\theta(a|s) \nabla_\theta \ln \pi_\theta(a|s) \Big[Q^\pi(s,a) - \hat{V}(x) - \nabla_\theta \ln \pi_\theta(a|s)^\top \hat{\boldsymbol{w}} \Big] \qquad (3.12) \\
&= \nabla_w \epsilon(\hat{\boldsymbol{w}}),
\end{aligned}
$$

where Eq.3.12 is obtained from the definition of the state-action value function and the properties of the TD regressor that has a zero mean for each state, Eq.3.6, and is independent between different time steps, and $\epsilon(\boldsymbol{w})$ defined at Eq.3.5 is the mean square error about $\boldsymbol{w}$. □

Therefore, because $\hat{\boldsymbol{w}}$ in the TD approximation could converge to the unbiased natural policy gradient [10] when $\lambda = 1$, the NTD algorithm can have almost the same suitable properties, as shown in Kakade (2002) and Bagnell and Schneider (2003). That is, the policy parameter is unaffected by the correlation of the parameters. When $\lambda = 0$, $\boldsymbol{\theta}$ is updated in the direction of the value function estimate. The eligibility trace by $\lambda \in (0, 1)$ fills the gap between the above two limiting cases. The characteristics of λ are similar to those of the decay rates used in TD(λ) (Sutton, 1988) and the actor-critic architecture proposed by Kimura and Kobayashi (1998).

[10] In fact, eq.2.10 implies that the gradient is also biased about the average reward. However, as mentioned in the previous section, we neglect the bias in this paper.

3.2.4 Implementation of NTD Algorithm

Heretofore, we have focussed on the estimations of NPG. Here, the NTD algorithm is presented as a RL algorithm, where the policy parameter is updated by the estimated NPG $\hat{\boldsymbol{w}}$ with an appropriate learning rate α,

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha \hat{\boldsymbol{w}}.$$

We propose a simple implementation of the NTD algorithm based on the gradient descent NPG estimation as algorithm 1. As mentioned in section 3.2.1, the NTD algorithm comprises three components—the value function estimator, the TD regressor as the NPG estimator, and the policy. Although it is preferable that the policy update waits for the other components to complete the estimations, a heuristic procedure would be effective in training all the components simultaneously. It is that the weight of the TD regressor is forgotten by a rate $\beta \in [0, 1]$ at each time step, $\boldsymbol{w} \leftarrow \beta \boldsymbol{w}$. Thus, the adverse affect derived from the strong variance of the TD estimator during incomplete learning can be avoided because the elements of $\hat{\boldsymbol{w}}$ which couple with rarely experienced state-action pairs decays to zero and then the corresponding elements of the policy parameter vector are not updated. Indeed, if $\beta \neq 1$, $\hat{\boldsymbol{w}}$ will be biased. When $\beta = 0$, the NTD algorithm corresponds to a standard policy gradient algorithm (Kimura and Kobayashi, 1998). Therefore, the forgetting rate, β, fills the gap between the standard policy gradient and the natural policy gradient. Table 3.1 specifies the NTD algorithm with eligibility traces.

3.2.5 Numerical Experiments

In this section, we test the performance of the NTD algorithm in not only a MDP but also in a continuous state problem. In the application of the NTD algorithm to continuous state problems, consider a continuous state problem as a finite state POMDP by function approximation.

Two state MDP (Kakade, 2002)

We first apply the NTD algorithm to a two-state MDP (Kakade, 2002) in order to investigate whether it can avoid plateaus and the property concerning the forgetting rate β. A comparison with the natural actor-critic algorithm (NAC) (Peters et al., 2003) as an alternative NPG is also presented. Each state has self- and cross-transition actions and rewards, as shown in Figure 3.2. The optimal policy is to maintain the execution of the self-transition action in state s_1 and obtains two as the maximum average reward. The policy has a sigmoidal parameterization

$$\begin{cases} \pi(u = \mathrm{self}|s = i) & = \dfrac{1}{1 + \exp(-\theta_i)} \\ \pi(u = \mathrm{cross}|s = i) & = 1 - \pi(u = \mathrm{self}|s = i), \end{cases}$$

and the policy parameter is initialized to corresponds to the following stationary distributions: $d(s = 1) = .8$ and $d(s = 2) = .2$. Under this setting, Kakade (2002) demonstrated that an ordinary policy gradient method was trapped in a plateau as the suboptimal policy in contrast to the natural policy gradient method, where the chance of a self-loop at the state $s = 1$ increases and then the stationary probability of the state $s = 2$ decreases.

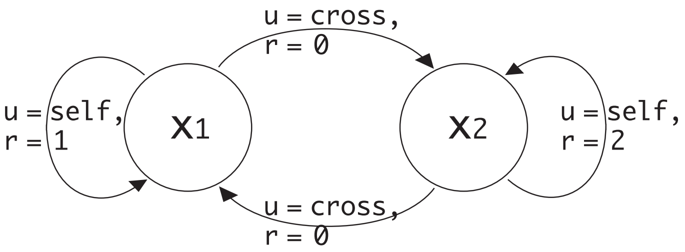


Figure 3.2. The task setting of 2-state MDP.

Performance of NTD algorithm at various β: The NTD algorithm was applied at each forgetting rate $\beta \in \{0, .99, .995, .999, .9995, 1\}$ on $\boldsymbol{w}$, which controls the trade-off between the ordinary and the natural gradient. The other meta-parameters were set appropriately by trial and error, as shown in table 3.2. The policy parameter was initialized as $\boldsymbol{\theta} = [1.4, -2.2]^\top$ to set the stationary distribution as $d(1) = .8$ and $d(2) = .2$. Figure 3.3 (a) shows the average rewards over the time course. Although the agents with larger β could find the optimal policy, the agents with lower β were trapped in the plateau. Figure 3.3 (b) shows the phase plane of the policy parameter $\boldsymbol{\theta} = [\theta_1, \theta_2]^\top$. When θ_1 is small and θ_2 is large, the policy is optimal. The left side of Figure 3.3 (b) shows that the NTD algorithm with β closer to 1 learns along a better trajectory of θ, and the trajectory with $\beta = 1$ is observed to be approximately the same that as in (Kakade, 2002). This indicates that the NTD algorithm with β close to 1 can estimate the natural policy gradient appropriately. The ordinary policy gradient method with the eligibility traces of the policy proposed by Kimura and Kobayashi (1998), Kimura's method, is also applied to each decay rate $\lambda \in \{.0, .5, .9, .99, .999\}$ of the eligibility trace. Kimura's method is similar to the NTD algorithm from the viewpoint of the usage of the TD error for policy updating. The essential difference is that the NTD algorithm stores the TD error based on the eligibility of the policy, while Kimura's method stores the eligibility by itself. The right side of Figure 3.3 (b) shows that Kimura's method was trapped in the plateau and ultimately failed to achieve the optimal policy.

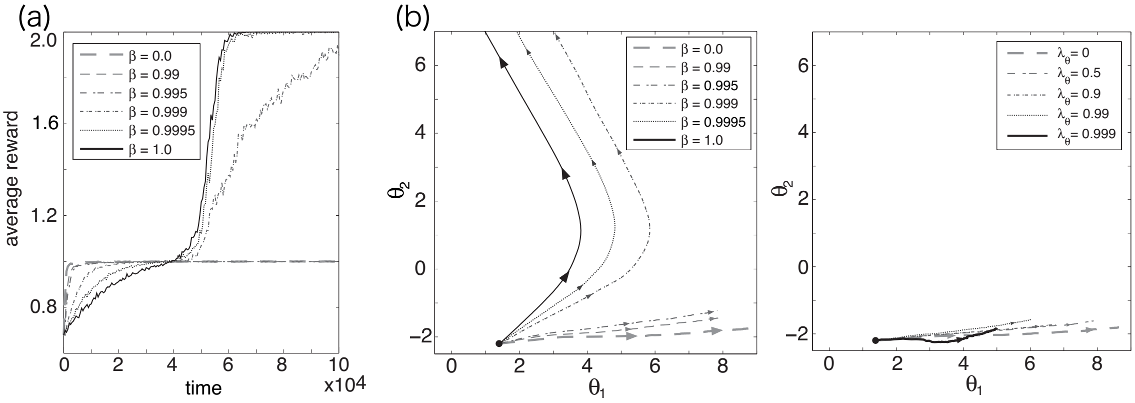


Figure 3.3. Two-state MDP: the averages of ten independent runs. (a) The average rewards over the time course at each value of β. (b) The phase plane –left: the NTD algorithm; right: the ordinary policy gradient method Kimura and Kobayashi (1998).

Comparison with NAC algorithm (Peters et al., 2003): With regard to the actual computational times and trial time steps required for learning, we compared the NTD algorithm with NAC (Peters et al., 2003) as an alternative natural policy gradient algorithm that requires the computation the inverse of a matrix. In the experiments, various values of the dimensional observation state feature vector $\boldsymbol{\psi}(s) \in \mathcal{R}^2$, $\mathcal{R}^5$, $\mathcal{R}^{15}$, $\mathcal{R}^{50}$, or $\mathcal{R}^{100}$, were applied. Because the feature vectors, except for the two-dimensional case, were redundant, we used the Moore-Penrose pseudoinverse of the matrix for the matrix inversion in NAC in all dimensional cases. Each vector was initialized in two steps: first, each the element of temporal vectors $\boldsymbol{\phi}(s_i)$ at $i \in \{1, 2\}$ was decided by uniform distribution $[0, 1]$; and second, each the vector was normalized as $\boldsymbol{\phi}(s_i) := \boldsymbol{\phi}(s_i)/\|\boldsymbol{\phi}(s_i)\|$. The meta-parameters of each algorithm were set to estimate the optimal policy as quick as possible, which are shown in table 3.3. We define an episode as being a "success" when the policy in that the episode reaches the optimal, i.e., $\tilde{\theta}_1 < 0$ and $\tilde{\theta}_2 > 5$, within 50000 time steps, where

$$\tilde{\theta}_i = \frac{\ln \pi_\theta(u = \mathrm{self}|\boldsymbol{\phi}(s_i))}{\ln \pi_\theta(u = \mathrm{cross}|\boldsymbol{\phi}(s_i))}.$$

Otherwise, the episode is called a "failure" and is not used for the results of figure 3.5. Figure 3.4 shows the success rate of each algorithm at each dimensional feature vector and suggests that most of simulation runs on both methods succeeded in learning. It supports that the setting of the meta-parameters was nearly appropriate with regard to the learning speed. Figure 3.5 shows the computational times and the time steps at each dimensional feature vector for learning. The NTD algorithm was faster in the most dimensional cases with regard to the computational time required for learning, although the NTD algorithm needed larger time steps. It indicates that the NTD algorithm is more suitable for actual complex problems, where we need to consider large dimensional feature vectors, while NAC would work better in the case of proper low-dimensional state feature vectors.

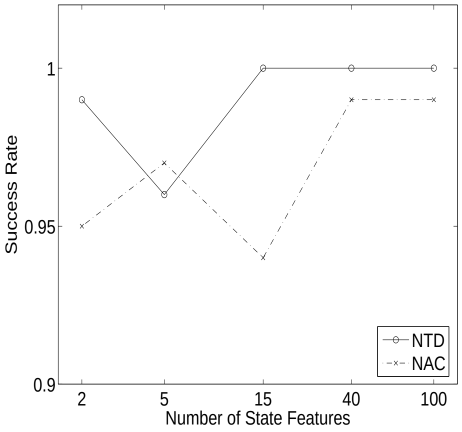


Figure 3.4. Two-state MDP: 100 independent runs. The learning success rate of each algorithm at each dimensional feature vector. NAC is natural actor-critic algorithm Peters et al. (2003).

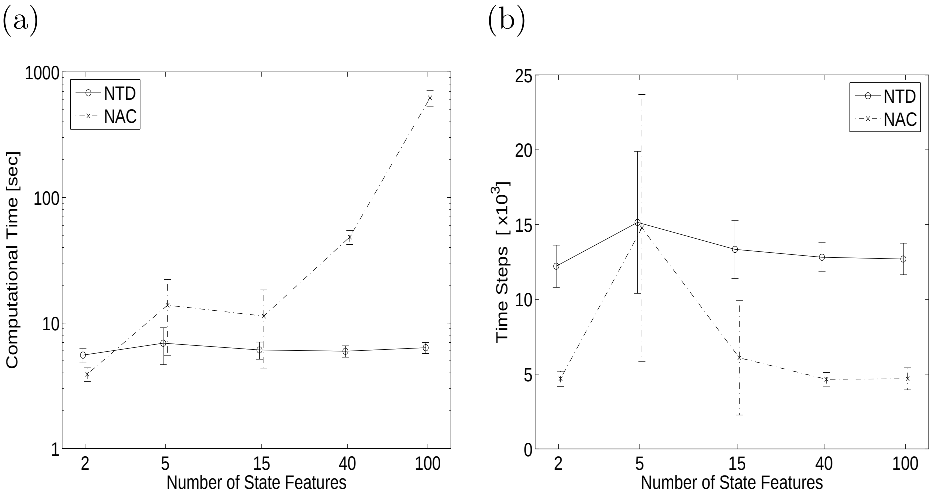


Figure 3.5. Two-state MDP: 100 independent runs. NAC is natural actor-critic algorithm Peters et al. (2003). (a) Actual computational times [s] at each dimensional feature vector for learning. (b) Trial time steps at each dimensional feature vector for learning.

Continuous State Problem

Interpretation of continuous state problems as POMDP: Although policy gradient RL algorithms, including the NTD algorithm, has been developed with finite states and actions, we can apply these algorithms to continuous state problems with function approximation by the following interpretation. When the policy (or the state value estimate) in a continuous problem is represented by the function approximator which has finite basis functions with bounded activation values and finite parameters, the continuous problem can be regarded as a POMDP by regarding the activations biased to non-negative values and normalized as the belief states of finite-state POMDPs (Aberdeen, 2003). Therefore, if the NTD algorithm is applied to continuous state problems with function approximation by using bounded basis functions, the NTD algorithm can estimate a local optimal policy parameter in terms of a POMDP model defined by the function approximator [11].

Pendulum swing-up problem: In this section, we compare the NTD algorithm with other policy gradient methods, NAC (Peters et al., 2003) and Kimura's actor-critic method (Kimura and Kobayashi, 1998), and examine the effect of the eligibility trace for NPG estimation, with regard to the pendulum swing-up problem, which is a continuous state problem. The pendulum swing-up problem with limited torque is a well known benchmark in RL (Doya, 2000). The state $\boldsymbol{\phi}(s) = [x, \dot{x}]^\top$ comprises the angle and the angular speed, as shown in Figure 3.6 (a). The action is a target torque u and is a probability variable following the Gaussian distribution defined by the policy

$$\pi_\theta(u|s) = \frac{1}{\sqrt{2\pi\, \sigma_\theta^2(s)}} \exp\left(-\frac{(u - \mu_\theta(s))^2}{2\,\sigma_\theta^2(s)}\right),$$

where the mean, $\mu_\theta(s)$, and the standard deviation, $\sigma_\theta(s)$, are defined by the policy parameterization and parameter. The pendulum dynamics are given by

$$\ddot{x} = \frac{-\mu\dot{x} + mgl\sin(x) + \tilde{u}}{ml^2},$$

[11] In order to guarantee the above theoretical results, the stochastic process model of this POMDP satisfies the ergodicity condition.

where $\tilde{u}_t$ is the actual torque of the system with the signum function $\mathrm{sign}(u)$,

$$\tilde{u} = \begin{cases} u & |u| \leq u^{\max}, \\ \mathrm{sign}(u)\, u^{\max} & \text{otherwise.} \end{cases}$$

The physical parameters are $m = 1$[kg], $l = 1$[m], $g = 9.8$[m/s^2], $\mu = .01$[N $\cdot$ m], and $u^{\max} = 5$[N $\cdot$ m]. An episode lasts for 20 seconds and the sampling is executed with the time step of 0.02[sec]. In many cases, a heuristic is employed, where an episode ends when the pendulum is over-rotated in order to eliminate the suboptimal policy that keeps the pendulum rotating continuously. In this experiment, instead of introducing the heuristic, we set the reward function as $r_{t+1} = \cos(x_{t+1}) - (\dot{x}_{t+1}/50\pi)^2$, in order to make this problem more challenging.

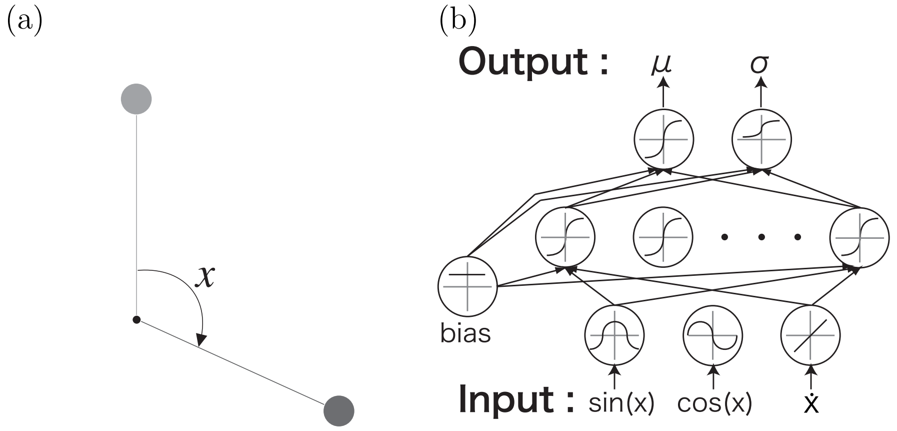


Figure 3.6. Pendulum swing-up task setting. (a) Control of a pendulum with limited torque. (b) Policy setting. The policy is a three-layer neural network based on sigmoidal functions with ten hidden units, the outputs of which correspond with the mean and the standard deviation of the normal distribution.

Here we use a general basis function setting. As shown in Figure 3.6 (b), the mean and the standard deviation of the policy are implemented by a three-layer neural network with ten hidden units, that is, the number of policy parameter elements is 64. Each the element θ_i was initialized by uniform distribution $[-.5,\ .5]$ at each simulation run. The state value function is implemented by normalized radial basis function (RBF) network (Doya, 2000), the parameter of which was initialized as $\mathbf{0}$. We add a typical heuristic to all methods, where the update of the policy is not executed in the first 100 episodes, in order to avoid using the incomplete estimates from the critics for the policy updates. We also add a heuristic operation to the NTD algorithm, where the learning rate of NPG estimator, α_w, is adapted per ten episodes, according to the average of the norm of the basis function $\nabla_\theta \ln \pi_\theta(a|s)$ for ten episodes, $\mathbb{E}_{10\text{episodes}}\{\, \|\nabla_\theta \ln \pi_\theta(a|s)\| \,\}$,

$$\alpha_w = \frac{\tilde{\alpha}_w}{\mathbb{E}_{10\text{episodes}}\{\, \|\nabla_\theta \ln \pi_\theta(a|s)\| \,\}}.$$

That is because $\mathbb{E}\{\, \|\nabla_\theta \ln \pi_\theta(a|s)\| \,\}$ varies during learning for the policy following the Gaussian distribution, since $\|\nabla_\theta \ln \pi_\theta(a|s)\|$ is inversely proportional to $\sigma_\theta(s)$ and $\sigma_\theta(s)$ varies (decreases in many cases) during learning. In NAC, the computation of the matrix inversion and the policy update are executed only at the end of each episode, instead of each time step, in order to suppress computational costs. Despite this, NAC consumed about three times computational costs than other methods in this experiment.

The comparison among the policy gradient algorithms was conducted under a proper setting of the basis function for the state value estimation, which has 15 × 15 RBFs about $x \in (-\pi, \pi]$ and $\dot{x} \in [-15, 15]$. The meta-parameters of each algorithm were set appropriately, as shown in table 3.4. Figure 3.7 (a) and (b) show the average rewards and the average numbers of pendulum rotations over the time course, respectively. Figure 3.7 (a) shows that the NTD algorithm obtained the optimal policy quickly, while NAC and Kimura's method needed considerably more time steps for learning. Figure 3.7 (b) indicates that Kimura's method appeared to be trapped in a plateau, in which the pendulum continued rotating, because this method did not follow the natural policy gradient. Although Peters (2005) shows that NAC can be applied to the pendulum swing-up problem appropriately with elaborate basis functions, the problem of setting the basis function still remains very difficult. In most actual problems, we would not know the elaborate setting for the basis function. Hence, we would use general function approximators such as those used in this experiment.

The comparison among the NTD algorithms at the various eligibility decay rates $\lambda_w \in \{0, .95, .99, 1\}$ was conducted under a rough setting for the state value estimation as 3 × 3 RBFs, which cannot represent the state value function adequately. Figure 3.8 shows the average rewards over the time course, which demonstrates that the performance improves as λ_w is close to one. Therefore, we confirmed that the eligibility trace for the NPG estimator worked effectively when the estimated state value function was poor or rough, consisting with the theoretical result in the section 3.2.3. It is noted and supports the effectiveness of the eligibility trace, that the system at $\lambda_w = 0$, which is without the eligibility trace, was unstable since the learning parameters diverged at a rate of 20% each simulation run, while the systems at the high eligibility decay rates were stable since the divergence rates at $\lambda_w = .95$, .99, and 1 were 3%, 3%, and 0%, respectively. Figure 3.8 does not use the results of the episodes where the parameters diverged.

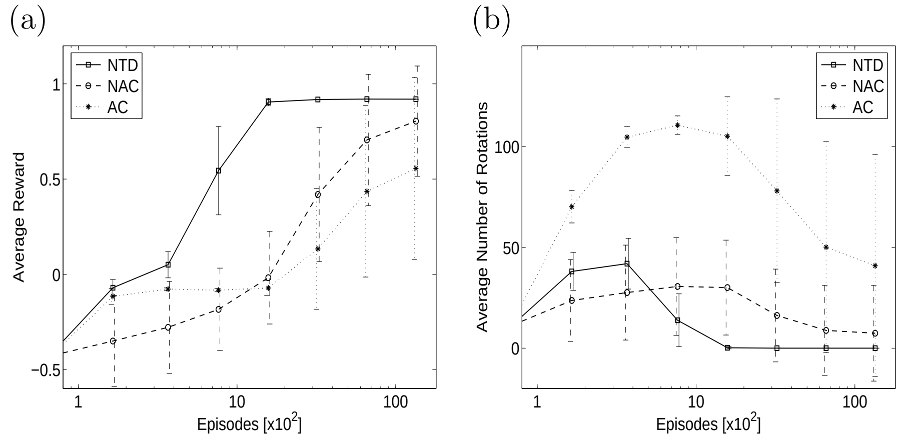


Figure 3.7. Pendulum swing-up problem; the averages over 30 independent runs. Comparison among the policy gradient algorithms under the proper RBF setup, [15 × 15], for the state value estimation, about (a) the average rewards and (b) the average number of rotations in a episode, over the time course. NAC is natural actor-critic (Peters et al., 2003) and AC is Kimura's actor-critic method (Kimura and Kobayashi, 1998).

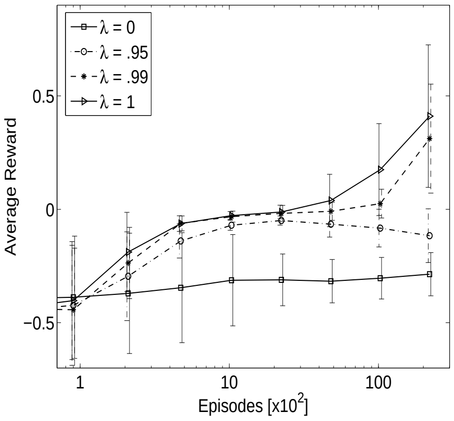


Figure 3.8. Pendulum swing-up problem; the averages over 30 independent runs. Comparison among various λ_w of the NTD algorithm about the average rewards over the time course under the improper (rough) RBF setup, [3 × 3], for the state value estimation.

Table 3.1. The NTD algorithm

```
Input:
  • Initial parameters; $\boldsymbol{\theta}$, $\boldsymbol{w}$ and $\boldsymbol{v}$ define the policy $\pi_\theta(a|s)$,
    the NPG estimator $\hat{\delta}(s,a) \equiv \boldsymbol{w}^\top \nabla_\theta \ln \pi_\theta(a|s)$ and
    the value estimator $\hat{V}(s)$, respectively.
  • Metaparameters; $\gamma$ is the discouted rate of the value function,
    $\alpha_\theta$, $\alpha_w$ and $\alpha_v$ are the learning rates of $\boldsymbol{\theta}$, $\boldsymbol{w}$ and $\boldsymbol{v}$,
    $\lambda_w$ and $\lambda_v$ are the eligibility decay rates of $\boldsymbol{w}$ and $\boldsymbol{v}$,
    $\beta$ is the forgetting rate of $\boldsymbol{w}$, and $\iota$ is a free parameter.
Initialization:
  • Eligibility traces; $\boldsymbol{z}_w := \boldsymbol{0}$;  $\boldsymbol{z}_v := \boldsymbol{0}$;.
  • Initial condition; $s_0 \sim p(s_0)$,   $a_0 \sim \pi_\theta(a_0|s_0)$.
For $t = 0, 1, 2 \cdots$ do
  a. Sampling
       Execute action $a_t$, observe next state $s_{t+1}$ and reward $r_{t+1}$,
       and decide next action $a_{t+1} \sim \pi(a_{t+1}|s_{t+1})$.
  b. Critic update
     ○ Forget NPG estimator parameter
         $\boldsymbol{w} := \beta \boldsymbol{w}$;
     ○ Compute TD-errors
         $\delta_v = r_{t+1} + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)$
         $\delta_w = \delta_v - \hat{\delta}(s_t, a_t) + \iota \hat{\delta}(s_{t+1}, a_{t+1})$;
     ○ Update eligibility traces
         $\boldsymbol{z}_w := \gamma \lambda_w \boldsymbol{w}_w + \nabla_\theta \ln \pi_\theta(a|s)$;
         $\boldsymbol{z}_v := \gamma \lambda_v \boldsymbol{z}_v + \nabla_v \hat{V}(s_t)$;
     ○ Update value function parameter
         $\boldsymbol{v} := \boldsymbol{v} + \alpha_v \delta_v \boldsymbol{z}_v$;
     ○ Update NPG estimator parameter
         $\boldsymbol{w} := \boldsymbol{w} + \alpha_w \delta_w \boldsymbol{z}_w$;
  c. Actor update
         $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_\theta \boldsymbol{w}$;
```

Table 3.2. Meta-parameters in the experiment, “Performance of NTD algorithm at various β”.

Algorithm	γ	β	α_θ	α_w	α_v	λ	λ_v
NTD	.9	0	.3	.1	.05	0	0
NTD	.9	.99	.0015	.1	.05	0	0
NTD	.9	.995	7.5×10^{-4}	.1	.05	0	0
NTD	.9	.999	1.5×10^{-4}	.1	.05	0	0
NTD	.9	.9995	7.5×10^{-5}	.1	.05	0	0
NTD	.9	1	4.5×10^{-5}	.1	.05	0	0
AC	.9	-	.03	-	.05	0	0
AC	.9	-	.015	-	.05	.5	0
AC	.9	-	.005	-	.05	.9	0
AC	.9	-	.004	-	.05	.99	0
AC	.9	-	.003	-	.05	.999	0

Table 3.3. Meta-parameters in the experiment, “Comparison with NAC algorithm”.

Algorithm	γ	β	α_θ	α_w	α_v	λ	λ_v	ϵ
NTD	.9	1	.0003	.5	.2	0	0	-
NAC	.9	.999	.001	-	-	0	-	$\pi/180$

Table 3.4. Meta-parameters in the experiment,“Pendulum swing-up problem”.

Algorithm	RBF $[x, \dot{x}]$	γ	β	α_θ	$\tilde{\alpha}_w$	α_v	λ	λ_v	ϵ
NTD	[15×15]	.98	.99997	.001	.005	.05	0	.95	-
NAC	[15×15]	.98	.99995	.005	-	-	0	-	$\pi/18$
AC	[15×15]	.98	-	.0007	-	.05	0	.95	-
NTD	[3×3]	.98	.99999	.0001	.0002	.02	0	.02	-
NTD	[3×3]	.98	.99999	.0001	.0005	.02	.95	.02	-
NTD	[3×3]	.98	.99999	.0001	.001	.02	.99	.02	-
NTD	[3×3]	.98	.99999	.0001	.001	.02	1	.02	-

3.3 Extended NTD algorithm for variance reduction

In previous section and Morimura et al. (2005), we propose the NTD algorithm as an implementation of NAC without matrix inversion, which comprise the repetition of following three procedures. The first procedure updates the state value estimate $\hat{V}(s)$ by TD(λ) learning (Sutton and Barto, 1998). The second updates the NPG estimate $\hat{\boldsymbol{\omega}}$ through the regression with the linear function $f^{\pi}_{\hat{\omega}}(s_t, a_t) = \hat{\boldsymbol{\omega}}^{\top}\nabla_{\theta} \ln \pi_{\theta}(a_t|s_t)$ to the temporal difference (TD) given from the first,

$$\delta(s_t, a_t) = r_{t+1} + \gamma\hat{V}^{\pi}(s_{t+1}) - \hat{V}^{\pi}(s_t).$$

That is, the update direction of NPG estimate $\hat{\boldsymbol{\omega}}$ is [12]

$$\Delta\hat{\boldsymbol{\omega}} = \frac{1}{T}\sum_{t=0}^{T-1} \left(\delta(s_t, a_t) - f^{\pi}_{\hat{\omega}}(s_t, a_t)\right) \nabla_{\theta}\ln \pi_{\theta}(a_t|s_t). \tag{3.13}$$

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

Since $f^{\pi}_{\omega}(s, a)$ has the property for an arbitrary function $g(s)$, due to $\sum_{u} \nabla\pi_{\theta}(a|s) = \mathbf{0}$,

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{g(s)\nabla_{\theta} \ln \pi(a|s)|s\} = \mathbf{0},$$

the expectation of $\Delta\hat{\boldsymbol{\omega}}$ at a time-step t (eq.3.13) does not depend on the value of $\hat{V}(s_t)$. Therefore, the NTD algorithm uses the state value estimate at the current time-step as the baseline function $b(s)$ 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 $\hat{\boldsymbol{\omega}}$.

[12] 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.

3.3.1 Variance Reduction for Natural Policy Gradient Estimates

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 $\hat{\boldsymbol{\omega}}$, [13]

$$\mathrm{Var}^{\pi}(\hat{\boldsymbol{\omega}}) = \mathbb{E}_{M(\boldsymbol{\theta})}\{(\hat{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}}^*)^2\},$$

where $\boldsymbol{a}^2$ denotes $\boldsymbol{a}^\top \boldsymbol{a}$ for an arbitrary vector $\boldsymbol{a}$, and $\hat{\boldsymbol{\omega}}^* \equiv \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{\omega}}\}$ has to be equal to $\boldsymbol{w}^*$ for the unbiased regression. In gradient descent 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 update direction $\Delta\hat{\boldsymbol{w}}$ for $\hat{\boldsymbol{w}}$ (at a fixed policy $\boldsymbol{\theta}$). Although a sequence of samples $[s_1, ..., s_T]$ is not drawn independently in almost cases of RL, where the relationship $\mathrm{Var}^{\pi}(\frac{1}{T}\sum_t f(s_t)) = \frac{1}{T}\mathrm{Var}^{\pi}(f(s))$ does not hold due to correlation between the different time-step samples, Greensmith et al. (2004) 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 Greensmith et al. (2004), the following inequality holds

$$\begin{aligned}&\mathrm{Var}^{\pi}(\Delta\hat{\boldsymbol{w}}) \leq o + \\ &h^{\pi}\left(\frac{1}{T}\mathrm{Var}^{\pi}\left(\left(\hat{Q}(s,a) - b(s) - f^{\pi}_{\hat{\omega}}(s,a)\right)\nabla_{\theta}\ln \pi_{\theta}(a|s)\right)\right),\end{aligned} \tag{3.14}$$

where o is independent with the choice of $b(s)$, and $\hat{Q}(s_t, a_t) = \mathbb{E}_{M(\boldsymbol{\theta})}\left\{r_{t+1} + \gamma\hat{V}(s_{t+1})|s_t, a_t\right\}$ and $b(s) = \hat{V}(s)$.

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

$$\begin{aligned}\sigma^2_{\Delta\hat{w}}(b(s)) &\equiv \mathrm{Var}^{\pi}\left(\left(\hat{Q}(s,a) - b(s) - f^{\pi}_{\hat{\omega}}(s,a)\right)\nabla_{\theta}\ln \pi_{\theta}(a|s)\right) \\ &= \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\left((\hat{Q}(s,a) - b(s) - f^{\pi}_{\hat{\omega}}(s,a))\nabla_{\theta}\ln \pi_{\theta}(a|s) - \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\Delta\hat{\boldsymbol{w}}\right\}\right)^2\right\}.\end{aligned}$$

[13] (Peters and Schaal, 2006) 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.

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

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

it is derived as

$$b^*(s, \hat{\boldsymbol{w}}) = \frac{\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\nabla_\theta \ln \pi_\theta(a|s)^2(\hat{Q}(s,a) - f^\pi_{\hat{\omega}}(s,a))|s\right\}}{\mathbb{E}_{M(\boldsymbol{\theta})}\{\nabla_\theta \ln \pi_\theta(a|s)^2|s\}}. \tag{3.15}$$

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

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

We show the following proposition for the policy parameterization:

Proposition 3 *Let S and A_i denote the numbers of states and available actions at state s_i, respectively. Let the matrix $\boldsymbol{\Psi}(\boldsymbol{\theta})$ denote the subspace spanned by $\nabla_\theta \ln \pi_\theta(a|s)$ over states and actions. If the rank of $\boldsymbol{\Psi}(\boldsymbol{\theta})$ is equal to (or greater than) $\sum_{i=1}^S (A_i - 1)$, the policy parameterization is nondegenerate for the task:*

$$f^\pi_{\omega^*}(s,a) \equiv \boldsymbol{w}^{*\top} \nabla_\theta \ln \pi_\theta(a|s) = Q^\pi(s,a) - V^\pi(s). \tag{3.16}$$

Proof: It comes from the fact that the constraint of $f^\pi_\omega(s,a)$ (eq.3.6) is satisfied, because

$$\sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^\pi(s,a) - V^\pi(s) = 0,$$

for each state. □

From proposition 3 and eq.3.15, it is just under the following case for the state value function to be equal to the optimal baseline function.

Proposition 4 *If the condition of proposition 3 is satisfied,*

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

Proof: It is obvious by substituting eq.3.16, "$\hat{Q}(s,a) - f^\pi_{\hat{\omega}^*}(s,a) = \hat{V}(s)$", to eq.3.15. □

Proposition 4 means that the optimal baseline is equivalent to the state 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.

3.3.2 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 could 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,

$$B(s, \hat{\boldsymbol{\omega}}) = \frac{\mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\nabla_\theta \ln \pi_\theta(a|s)^2(\hat{Q}(s,a) - \hat{V}(s) - f^\pi_{\hat{\omega}}(s,a))|s\Big\}}{\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\nabla_\theta \ln \pi_\theta(a|s)^2|s\right\}}. \tag{3.17}$$

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,

$$\delta(s_t, a_t) - B(s_t, \hat{\boldsymbol{\omega}}) = r_{t+1} + \gamma V(s_{t+1}) - b^*(s_t, \hat{\boldsymbol{\omega}}). \tag{3.18}$$

Although eq.3.18 seems roundabout to apply the optimal baseline, it is useful for an eligibility trace technique with estimated value functions (see fig.3.9). In order to estimate $B(s, \hat{\boldsymbol{w}})$, the gradient of $\sigma^2_{\Delta\hat{\boldsymbol{w}}}(b(s))|_{b(s)=V^\pi(s)+\hat{B}_b(s,\hat{\boldsymbol{w}})}$ with respect to the parameter $\boldsymbol{b}$ of $\hat{B}_b(s)$ is used. Fig.3.9 is one of the complete algorithms.

```
Input:
  • Initial parameters; $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, $\boldsymbol{v}$, [ $\boldsymbol{b}$ ] are the parameters of
    $\pi_\theta(a|s)$, $f^\pi_\omega(s,a)=\boldsymbol{\omega}^\top\nabla_\theta \ln \pi_\theta(a|s)$, $\hat{V}(s)$, [ $\hat{B}(s)$ ].
  • Metaparameters; $\gamma$ is the discouted rate of the value function,
    $\alpha_\theta$, $\alpha_\omega$, $\alpha_v$, [ $\alpha_b$ ] are the learning rates of $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, $\boldsymbol{v}$, [ $\boldsymbol{b}$ ].
    $\lambda_\omega$, $\lambda_v$,[ $\lambda_b$ ] are the eligibility decay rates of $\boldsymbol{\omega}$, $\boldsymbol{v}$, [ $\boldsymbol{b}$ ].
    $\beta$ is the forgetting rate of $\boldsymbol{\omega}$.
For $t = 0, 1, 2 \cdots$ do
  a. Sampling
       Execute action $a_t$, observe next state $s_{t+1}$ and reward
       $r_{t+1}$, and decide next action $a_{t+1} \sim \pi_\theta(a_{t+1}|s_{t+1})$.
  b. Critic update
     ◦ Forget TD estimator parameter
         $\boldsymbol{\omega} := \beta\boldsymbol{\omega}$;
     ◦ Compute TD-errors
         $\delta_v := r_{t+1} + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t)$
         $\delta_\omega := \delta_v - f^\pi_\omega(s_t, a_t)$;
       [ $\delta_b := \delta_\omega - \hat{B}(s_t, \boldsymbol{\omega}) + \gamma\lambda_b\hat{B}(s_{t+1}, \boldsymbol{\omega})$; ]
     ◦ Update critic eligibilities
         $\boldsymbol{z}_v := \gamma\lambda_v\boldsymbol{z}_v + \nabla_v\hat{V}(s_t)$;
         $\boldsymbol{z}_\omega := \gamma\lambda_\omega\boldsymbol{z}_\omega + \nabla_\theta \ln \pi_\theta(a_t|s_t)$;
       [ $\boldsymbol{z}_b := \gamma\lambda_b\boldsymbol{z}_b + \nabla_\theta \ln \pi_\theta(a_t|s_t)^2\nabla_b\hat{B}(s_t, \boldsymbol{\omega})$; ]
     ◦ 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 3.9. 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.

Current State	Action	Following State S_1	 S_2	 S_2
S_1	A_1	1	0	0
S_1	A_2	0	1	0
S_2	A_1	0.2	0.2	0.6
S_2	A_2	0.6	0.2	0.2
S_3	A_1	0.8	0.1	0.1
S_3	A_2	0.1	0.1	0.8

Table 3.5. Transition probabilities on the three-state MDP

$$r(S_1) = 1, \qquad \boldsymbol{\phi}(S_1) = [1,\, 0.1]^\top$$
$$r(S_2) = 0, \qquad \boldsymbol{\phi}(S_2) = [1,\, 1]^\top$$
$$r(S_3) = 2, \qquad \boldsymbol{\phi}(S_3) = [1,\, 10]^\top$$

Table 3.6. The reward function and the feature vector of the state on the three-state MDP

3.3.3 Numerical Experiments

MDP with inadequate policy

We selected the 3-state 2-action MDP in Baxter et al. (2001) where the state-transition probability and the parameterization of policy are modified from original. There are three kinds of states S_1, S_2, S_3 and each state has two kinds of actions A_1, A_2. The state-transtion probability is showed in table 3.5. Each state is obserbed as two-dimentinal vector $\boldsymbol{\phi}(s) = \mathcal{R}^2$ and has the corresponding reward $r(s)$ as table 3.6.

Under this policy parameterization, the condition of proposition 3 cannot 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 4. Fig.3.10 indicates that the extended NTD suppresses the variance of the NPG estimates than the normal NTD.

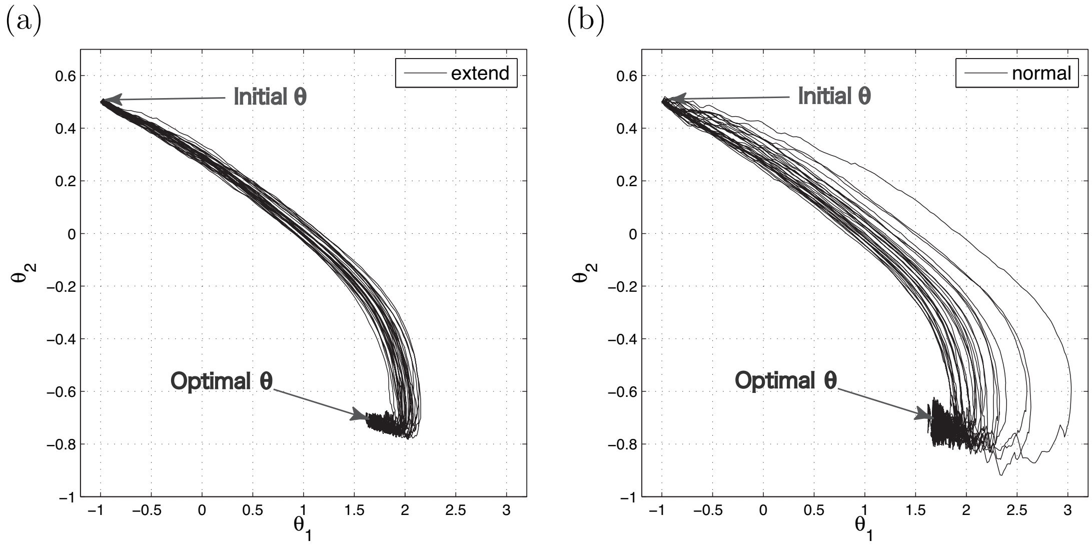


Figure 3.10. MDP; phase plane analyses; policy parameter trajectories (a) the extended NTD, (b) the normal NTD.

Pendulum swing-up problem

This section gives the comparison between NTD algorithms and other policy gradient methods; NAC (Peters et al., 2003), Kimura Actor-Critic (Kimura and Kobayashi, 1998) in the same setting as the pendulum swing-up problem in section 3.2.5.

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

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

$$b_2(s, \hat{\boldsymbol{\omega}}) = \frac{\mathrm{E}^\pi\{\nabla_\theta \ln \pi_\theta(a|s)^2 f^\pi_{\hat{\omega}}(s,a)\}}{\mathrm{E}^\pi\{\nabla_\theta \ln \pi_\theta(a|s)^2\}}.$$

When we use the Gaussian distribution policy in section 3.2.5, while $b_1(s)$ has to be estimated, $b_2(s, \hat{\boldsymbol{\omega}})$ could be solved analytically: $\boldsymbol{b}_\mu(s) \equiv \nabla_\theta \mu_\theta(s)$, $\boldsymbol{b}_\sigma(s) \equiv \nabla_\theta \sigma_\theta(s)$,

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

Fig.3.11 showed that the extended NTD algorithm works better than the other PG algorithms.

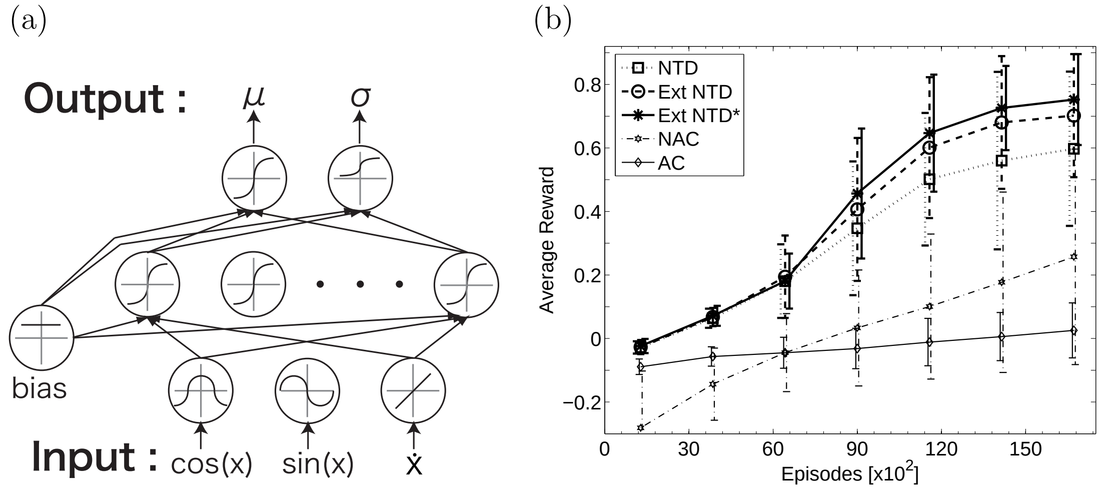


Figure 3.11. Swing-up pendulum problem; (a) The policy is a three-layer neural network with 10 hidden units. (b) The average rewards over 30 independent runs. Comparison among PGs under the improper RBF setup, [5× 5], for the state value estimation. Extended NTD* is the alternative algorithm computing b_1 analytically.

3.4 Utilizing Baseline Adjustment Function for Policy Parameterization

As mentioned in chapter 1, it would be a difficult and important matter how to parameterize an appropriate policy for RL problems. Here, we discuss this problem and focus especially on the problem how to regulate the number of hidden-units of multi-layer perceptron as the policy automatically.

3.4.1 Absolute value of auxiliary function as criterion

From eq.3.17, the auxiliary function of extended NTD algorithm to adjust baseline, $B(s, \hat{\boldsymbol{\omega}})$, represents the differences between the state value function $V^{\pi}(s)$ and the optimal baseline function $b^{*}(s, \hat{\boldsymbol{\omega}})$,

$$B(s, \hat{\boldsymbol{\omega}}) = b^{*}(s, \hat{\boldsymbol{\omega}}) - V^{\pi}(s).$$

When the policy parameterization is nondegenerate for the task (and the NPG estimate converges to the exact NPG), the state value is equal to the optimal baseline by proposition 4,

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

Accordingly, if the policy parameterization is nondegenerate for the task, the absolute value of auxiliary function becomes zero,

$$|B(s, \hat{\boldsymbol{\omega}}^{*})| = 0.$$

Meanwhile, if the policy parameterization is degenerate (or not sufficient),

$$|B(s, \hat{\boldsymbol{\omega}}^{*})| \neq 0.$$

also holds by proposition 3 [14]. Therefore, the absolute value of auxiliary function could be a valid criterion for the policy parameterization,

$$c(\pi) \equiv \sum_{s \in \mathcal{S}} d^{\pi}(s) |B(s, \hat{\boldsymbol{\omega}}^{*})|,$$

where the parameterization of the policy π is better in smaller $c(\pi)$.

[14] Even when the policy parameterization is degenerate, $|B(s, \hat{\boldsymbol{\omega}}^{*})| = 0$ holds under $\pi_{\theta}(a_i|s) = \pi_{\theta}(a_j|s)$ for all $a_i \mathcal{A}$ and $a_j \in \mathcal{A}$. However, such case seldom occurs in RL

3.4.2 Autonomous adjustment of the number of hidden-units of multi-layer perceptron

As one of applications utilizing the criterion $c(\pi)$, an auto-adjustmenting algorithm for the number of hidden units of a multi-layer perceptron (MLP) used as the policy is proposed here. Although the estimation for the exact value of the criterion $c(\pi)$ would be intractable, some properties of the criterion $c(\pi)$ can be evaluated and is enough to adjust the number of hidden-units, e.g.:

- when $c(\pi)$ is increasing, add the hidden-unit,
- when $c(\pi)$ is decreasing, do nothing,
- when $c(\pi)$ does not change and is larger than sufficiently small constant ε, add the hidden-unit,
- when $c(\pi)$ does not change and is smaller than sufficiently small constant ε, do nothing.

In order to evaluate the above properties about $c(\pi)$, we use favor of stochastic process (Osogami and Kato, 2007), especially random walk (figure 3.12). The complete algorithm is shown in Algorithm 3 that adjusts the number of MLP's hidden-units.

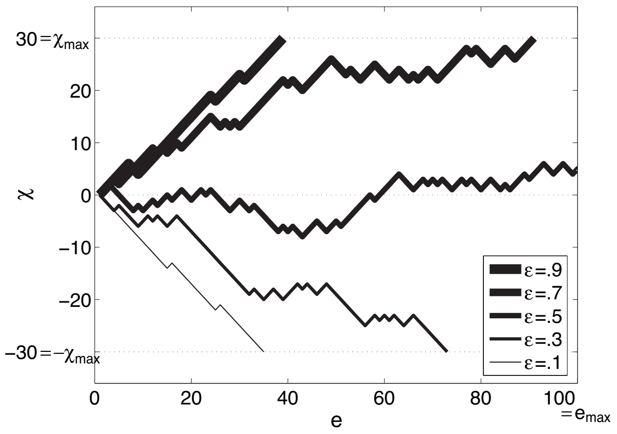


Figure 3.12. Example of random walk; $p(\chi_{e+1} = j+1 | \chi_e = j) = \epsilon$. horizontal and vertical axes represent episodes e and a state χ of the random walk.

Algorithm 3 Adjustment of number of MLP's hidden-units

```
Inputs:
    • Inputs for extended NTD algorithm (Fig. 3.9)
    • Metaparameters for random walk; $k$, $c_{\max}$, $\chi_{\max}$, $e_{\max}$and $\beta$
Initialization:
    • Initialization for extended NTD algorithm
    • Parameters of random walk; $\chi := 0$; $e := 0$;.
    • Criterion of policy parameterization; $c_{-1} := c_{\max}$;  $c := 0$.
For $t = 0, 1, 2 \cdots$ do
    Extended NTD algorithm (Fig. 3.9);
    $c := c + \left|\hat{B}(s_t, \hat{\boldsymbol{\omega}})\right|$;
    If $mod(t, k)^{\dagger} = 0$
        $c := c/k$;
        $e := e + 1$;
        If $c_{-1} < c$
            $\chi := \chi + 1$;
            If $\chi_{\max} < \chi$    ($c$ is increasing)
                addHiddenUnit;
                $\chi := 0$;   $e := 0$;
                $c_{-1} := \min(c, 2c_{-1})$;
            end
        else
            $\chi := \chi - 1$;
            If $-\chi < -\chi_{\max}$    ($c$ is decreasing )
                $\chi := 0$;   $e := 0$;
                $c_{-1} := \max(c, c_{-1}/2)$;
            end
        end
    end
        If $e_{\max} < e$ and $c_{\max} <= c_{-1}$ ($c$ does not change and is not be sufficiently small)
            addHiddenUnit;
            $\chi := 0$;   $e := 0$;
        end
end
```

3.4.3 Numerical experiment

We apply the proposed algorithm into the pendulum swing-up problem as explained in section 3.2.5. Figure 3.13 shows the time courses of the average rewards and the estimated criterion $\hat{c}(\pi)$. It indicates that proposed adjustment algorithm works better than or as well as the case of a fixed appropriate number of MLP's hidden-units. The time course of the number of the MLP's hidden-units shown in figure 3.14, where it was confirmed that the number converged to about 10. It suggests that the MLP of 10 hidden-units would be sufficient policy parameterization for pendulum swing-up problem. Figure 3.15 showed detail results about algorithm 3 of one simulation run.

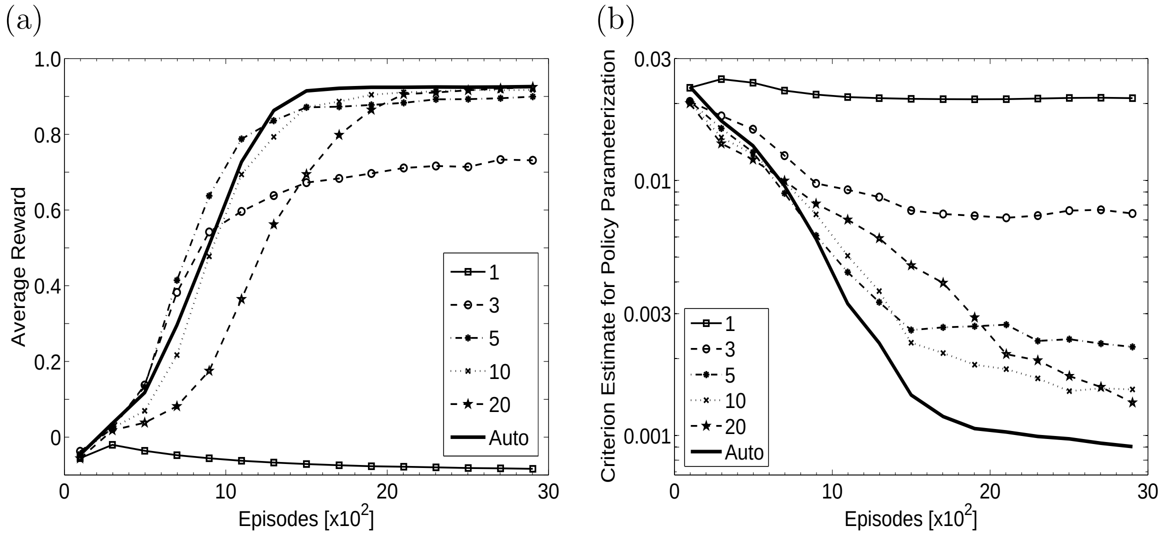


Figure 3.13. Pendulum swing-up problem; the averages rewards and the the criterion estimates $\hat{c}(\pi)$ over 30 independent runs. Comparison between fixing and auto-adjustmenting NTD algorithms about the number of hidden-units of the policy under the improper RBF setup, [5 × 5].

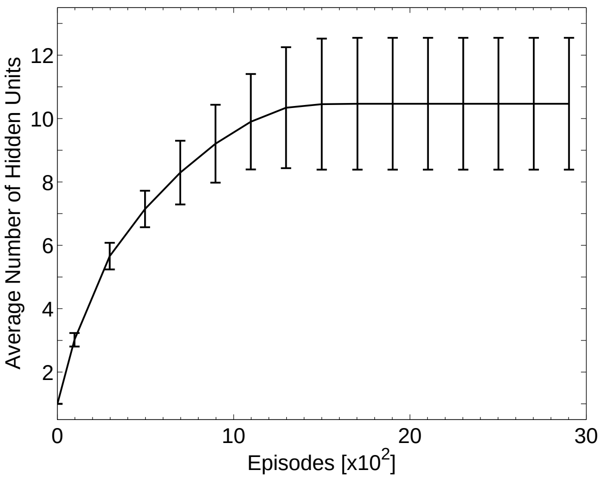


Figure 3.14. Pendulum swing-up problem; the number of the MLP's hidden-units of the policy over 30 independent runs.

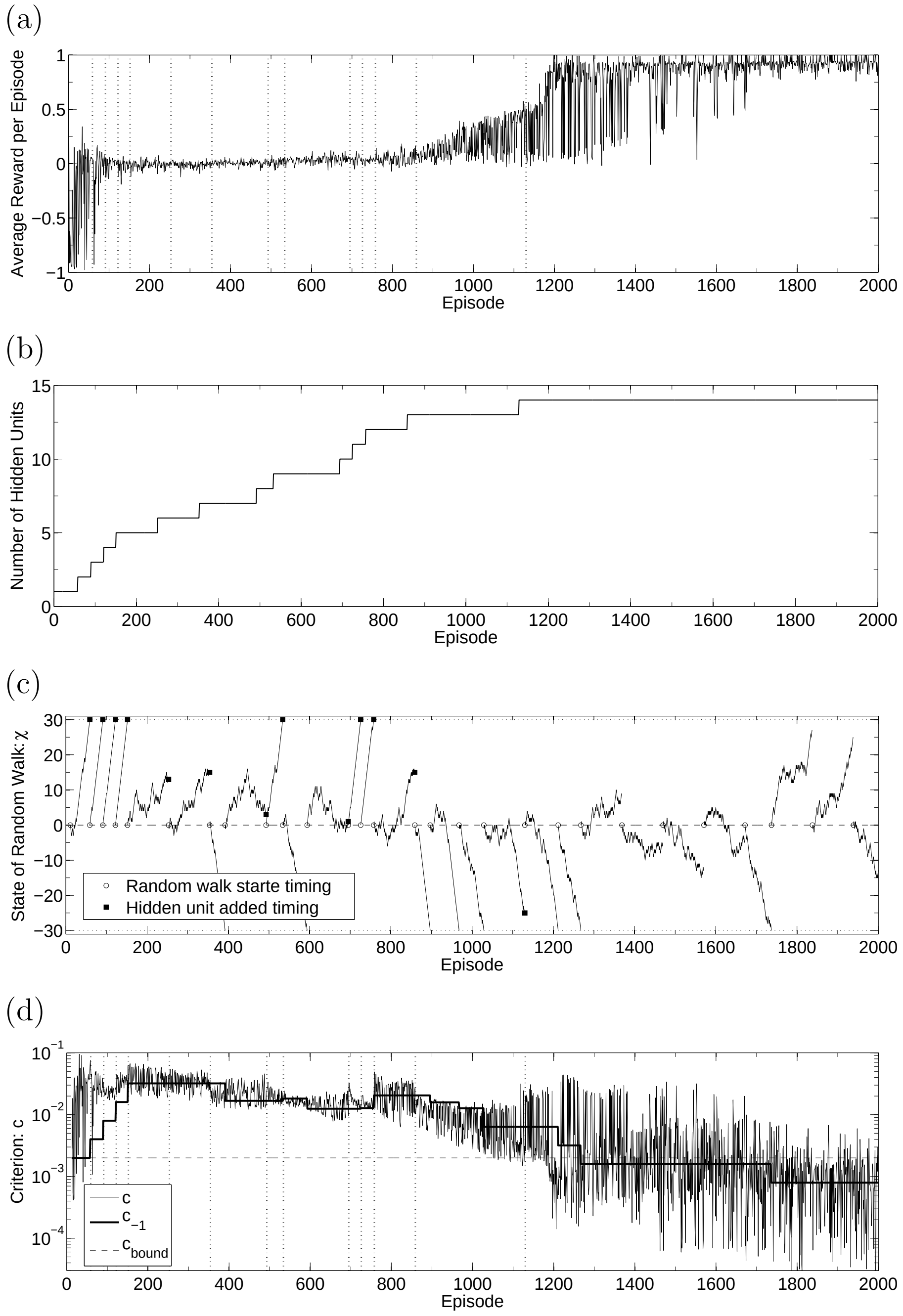


Figure 3.15. Pendulum swing-up problem; the time courses of the parameters in Algorithm 3 by one simulation run.

3.5 Summary and Discussion

This chapter presents the NTD algorithm, in which the regression weights of the TD error with the basis functions defined by the policy parameterization represents the natural policy gradient. If the eligibility decay rate of the NPG estimator is equal to one, the NPG estimate is updated by using the gradient of the actual observed rewards and not those of the estimated state value function; hence, the estimate is unbiased under a fixed policy. The experimental results showed that the NTD algorithm could represent the natural policy gradient and could avoid plateaus, which is consistent with the results of Amari (1998). This is extremely useful because plateaus often occur in RL problems when a suboptimal policy is more easily obtained than an optimal policy, as presented in the pendulum swing-up problem, The experimental results also demonstrated that the NTD algorithm suppresses computational costs than the existing NPG method (Peters et al., 2003) and the eligibility trace for the NPG estimator works efficiently.

This chapter also 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.

For the policy parameterization of the policy, we derived the criterion to judge whether or not the current parameterization of the policy is sufficient for the achievement of task objective, and proposed the algorithm to adjust the number of hidden units of a multi-layer perceptron. Additional theoretical and experimental analyses are necessary to further understand the properties and the effectiveness of the NTD algorithm.

Chapter 4

Policy Gradient with Derivative of Stationary Distribution

As pointed out in previous chapter, policy gradient reinforcement learning (PGRL) is a popular family of algorithms in reinforcement learning (RL) for improving a policy parameter to maximize the average reward by using the average reward gradients with respect to a policy parameter, which are called policy gradients (PGs) (Williams, 1992; Kimura and Kobayashi, 1998; Baird and Moore, 1999; Sutton et al., 2000; Baxter and Bartlett, 2001; Konda and Tsitsiklis, 2003). However, most of conventional PG algorithms for the infinite-horizon problem neglect a term associated with the derivative of the stationary distribution in PGs, since there are no algorithms to estimate this derivative so far (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998). The derivative means the measurement of how the stationary distribution changes due to the changes of the policy parameter. While the biases introduced by this omission can be reduced by taking a forgetting (or discouted) rate γ close to 1, it often increases the variance of the PG estimates and the setting "$\gamma = 1$" cannot be tolerated in these algorithms. This tradeoff makes it difficult to find an appropriate γ in practice. Meanwhile, there is the average reward PG algorithm (Tsitsiklis and Van Roy, 1999; Konda and Tsitsiklis, 2003), which eliminates the use of the forgetting rate by introducing a differential cost function as a solution of Poisson's equation. Because that was one and only PG framework proposed for maximizing the average reward[1], such studies about the average reward optimization are needed and significant.

Here, we propose a new PG framework with estimating the log stationary distribution derivative (LSD) as an alternative and useful form of the derivative of the stationary distribution for estimating PG. It is our main result in this chapter that an method to estimate LSD is derived through backward Markov chain formulation and a temporal difference learning method. The realization of this LSD estimation naturally enables the average reward gradient to be estimated regardless of the value of γ. Especially, in the case of "$\gamma = 0$", the estimation PG does not need to learn value functions. That is, a learning agent estimates LSD instead of value functions in this PG framework. One possible advantage of this framework is that a closed-form solution for an optimal baseline function of the PG can be computed by least squares, while that for the conventional PG framework has not yet been proposed and would be intractable (Greensmith et al., 2004).

The following is the outline: In Section 1, we describe motivation to estimate LSD. In Section 2, we propose an $\mathcal{LS}$LSD(λ) algorithm for the estimation of LSD by a $\mathcal{L}$east $\mathcal{S}$quares temporal difference method based on the backward Markov chain formulation. In Section 3, the $\mathcal{LS}$LSD(λ)-PG algorithm is instantly derived, which is a new γ-free PG algorithm utilizing $\mathcal{LS}$LSD(λ). To verify the performances of the proposed algorithms, the numerical results in simple Markov Decision Processes (MDP) are shown in Section 4. In Section 5, we summarize this chapter and also give other posibility brought by the realization of the LSD estimation, which concerns a natural policy gradient.

4.1 Why log stationary distribution derivative is important for PG estimation

We briefly review the conventional PGRL methods and present the main idea of our new algorithm. The policy gradient RL algorithms update the policy parameter $\boldsymbol{\theta}$ in the direction of the gradient of the average reward $R(\boldsymbol{\theta})$ with

[1] Although there is R-learning for maximizing the average reward, it is the algorithm based on the value function not the PG algorithm (Sutton and Barto, 1998).

respect to $\boldsymbol{\theta}$

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

which is often referred as the policy gradient (PG) for short (see Chapter 2.1 in detail). This is given by

$$\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \sum_{s_{+1} \in \mathcal{S}} d^{\pi}(s) \pi_{\theta}(a|s) \left(\nabla_{\theta} \ln \pi_{\theta}(a|s) + \nabla_{\theta} \ln d^{\pi}(s) \right) p(s_{+1}|s,a) r(s,a,s_{+1}). \tag{4.1}$$

It is noted that, in this chapter, s_{+k}, a_{+k}, and r_{+k} denote a state, an action, and an immediate reward after k time-steps from a state s, an action a, and an immediate reward r, respectively, and vice versa in $-k$. As the derivation of the gradient of the log stationary state distribution $\nabla_{\theta} \ln d^{\pi}(s)$ is nontrivial, the conventional PG algorithms (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998) utilize an alternative representation of the PG

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d^{\pi}(s) \pi_{\theta}(a|s) \nabla_{\theta} \ln \pi_{\theta}(a|s) Q^{\pi}_{\gamma}(s,a) \\ & + (1-\gamma) \sum_{s \in \mathcal{S}} d^{\pi}(s) \nabla_{\theta} \ln d^{\pi}(s) V^{\pi}_{\gamma}(s), \end{aligned} \tag{4.2}$$

where $Q^{\pi}_{\gamma}(x,u) \equiv \lim_{K \to \infty} \mathbb{E}_{M(\boldsymbol{\theta})} \{ \sum_{k=1}^{K} \gamma^{k-1} r_{+k} | s, a \}$ is an action value function and $V^{\pi}_{\gamma}(x) \equiv \lim_{K \to \infty} \mathbb{E}_{M(\boldsymbol{\theta})} \{ \sum_{k=1}^{K} \gamma^{k-1} r_{+k} | s \}$ is a state value function with discouted rate $\gamma \in [0,1)$ (Sutton and Barto, 1998).

Since the contribution of the second term of Eq.4.2 becomes smaller as γ approaches 1 (Baxter and Bartlett, 2001), the conventional algorithms (Baxter and Bartlett, 2001; Kimura and Kobayashi, 1998) approximated the PG only from the first term by taking $\gamma \approx 1$. Although the bias introduced by this omission becomes smaller as γ is set close to 1, the variance of the estimate becomes larger.

Here we propose an alternative approach, which estimates the log stationary distribution derivative (LSD), $\nabla_{\theta} \ln d^{\pi}(s)$, and uses Eq.4.1 for the derivation of the PG. A marked feature is that we do not need to learn the value function, and thus, the algorithm is free from the bias-variance trade-off in the choice of the discouted rate γ.

We should note that two methods to estimate the gradient of the (stationary) state distribution have already been proposed, although these are different from our proposal and have the following problems. The first is the method in operations research called “the likelihood ratio gradient” or “the score function” (Glynn, 1991; Rubinstein, 1991). However, their applicability is limited to regenerative processes (Baxter and Bartlett, 2001) [2]. Another method proposed by Ng et al. (2000) is not a direct estimation of the gradient of the state distribution and is done via the estimation of the state distribution with density propagation. Therefore, these methods require the knowledge of which state the agent is in, while our method only needs to observe the feature vector of the state.

4.2 Estimation of the Log Stationary Distribution Derivative (LSD)

In this section, we propose an LSD estimation algorithm based on least squares, $\mathcal{LS}$LSD(λ). For this purpose, we formulate the backwardness of the ergodic Markov chain $M(\boldsymbol{\theta})$, and show that LSD can be estimated in the temporal difference framework (Sutton, 1988; Bradtke and Barto, 1996; Boyan, 2002).

4.2.1 Properties of forward and backward Markov chains

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

$$q(s_{-1}, a_{-1}|s) = \frac{p(s|s_{-1}, a_{-1})p(s_{-1}, a_{-1})}{\sum_{s_{-1},a_{-1}} p(s|s_{-1}, a_{-1})p(s_{-1}, a_{-1})}.$$

The posterior $q(s_{-1}, a_{-1}|s)$ depends upon the prior distribution $p(s_{-1}, a_{-1})$. When the prior distribution follows the stationary distribution and the policy—$p(s_{-1}, a_{-1}) = \pi_\theta(a_{-1}|s_{-1})d^\pi(s_{-1})$—the posterior is termed as the *stationary* backward probabil-

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

ity and the subscript $B(\boldsymbol{\theta})$ is appended to it, where it appears as $q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1}|x)$,

$$q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1}|s) = \frac{p(s|s_{-1}, a_{-1})\pi_\theta(a_{-1}|s_{-1})d^\pi(s_{-1})}{d^\pi(s)}$$
$$= \frac{p_{M(\boldsymbol{\theta})}(s, a_{-1}|s_{-1})d^\pi(s_{-1})}{d^\pi(s)}. \tag{4.3}$$

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

Proposition 5 *Let a Markov chain $M(\boldsymbol{\theta})$ characterized by a transition probability $p_{M(\boldsymbol{\theta})}(s|s_{-1}) \equiv \sum_{a_{-1}} p_{M(\boldsymbol{\theta})}(s, a_{-1}|s_{-1})$ be irreducible and ergodic. Then the backward Markov chain $B(\boldsymbol{\theta})$ characterized by the backward (stationary) transition probability $q_{B(\boldsymbol{\theta})}(s_{-1}|s) \equiv \sum_{a_{-1}} q_{B(\boldsymbol{\theta})}(s_{-1}, a_{-1}|s)$ against $p_{M(\boldsymbol{\theta})}$ is also ergodic and has the same unique stationary distribution of $M(\boldsymbol{\theta})$:*

$$d_{M(\boldsymbol{\theta})}(s) = d_{B(\boldsymbol{\theta})}(s), \tag{4.4}$$

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

Proof: By multiplying both sides of Eq.4.3 by $d^\pi(s)$ and summing over all possible $a_{-1} \in \mathcal{A}$, we obtain

$$q_{B(\boldsymbol{\theta})}(s_{-1}|s)d^\pi(s) = p_{M(\boldsymbol{\theta})}(s|s_{-1})d^\pi(s_{-1}). \tag{4.5}$$

Then, $\sum_{s\in\mathcal{S}} q_{B(\boldsymbol{\theta})}(s_{-1}|s)d^\pi(s) = d^\pi(s_{-1})$ holds by summing both sides of Eq.4.5 over all possible $s \in \mathcal{S}$, indicating that (i) $B(\boldsymbol{\theta})$ has the same stationary distribution of $M(\boldsymbol{\theta})$ and (ii) $B(\boldsymbol{\theta})$ has the same irreducible property as $M(\boldsymbol{\theta})$. Eq.4.5 is reformulated by the transition probability, $p_{M(\boldsymbol{\theta})}(s|s_{-1})$ or $q_{B(\boldsymbol{\theta})}(s_{-1}|s)$, assembled to the matrix notation, $\boldsymbol{P}_{M(\boldsymbol{\theta})}$ or $\boldsymbol{Q}_{B(\boldsymbol{\theta})}$, respectively [3], and the stationary distribution to the vector notation $\boldsymbol{d}^\pi$: [4]

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

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

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

We can easily see that the diagonal components of $(\boldsymbol{P}_{M(\boldsymbol{\theta})})^n$ are equal to those of $(\boldsymbol{Q}_{B(\boldsymbol{\theta})})^n$ for any natural number n. This implies that (iii) $B(\boldsymbol{\theta})$ has the same aperiodic property as $M(\boldsymbol{\theta})$. Eq.4.4 is directly proven by (i)–(iii) (Schinazi, 1999). □

Proposition 6 *Let the distribution of s_{-K} follow $d^{\pi}(s)$; then, the expectations of both the directional Markov chains regarding the sum of arbitrary functions $f(s_{-k}, a_{-k})$ over $k \in [0, K]$ are equivalent:*

$$\mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\sum_{k=0}^{K} f(s_{-k}, a_{-k})\Big| s\Big\} = \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\sum_{k=0}^{K} f(s_{-k}, a_{-k})\Big| s, d^{\pi}(s_{-K})\Big\}, \tag{4.6}$$

where $\mathbb{E}_{B(\boldsymbol{\theta})}$ and $\mathbb{E}_{M(\boldsymbol{\theta})}$ denote the expectations over the forward and backward Markov chains, $B(\boldsymbol{\theta})$ and $M(\boldsymbol{\theta})$, respectively, and $\mathbb{E}\{\cdot|d^{\pi}(s_{-K})\} \equiv \mathbb{E}\{\cdot|p(s_{-K}) = d^{\pi}(s_{-K})\}$. Eq.4.6 holds even at the limitation, $K \to \infty$.

Proof: By utilizing the Markov property and substituting Eq.4.3, we have the following relationship:

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

It instantly proves the proposition in the case of the finite K. Since the following equations are derived with Proposition 5, the proposition in the limit case $K \to \infty$ is also instantly proven,

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

□

Propositions 5 and 6 are significant because they indicate that the samples from the forward Markov chain $M(\boldsymbol{\theta})$ can be used directly for estimations concerning the backward Markov chain $B(\boldsymbol{\theta})$ under the state distribution converging the stationary distribution, and thus can be utilized in the following sections.

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

LSD, $\nabla_\theta \ln d^\pi(s)$, is decomposed using Eq.4.3 to

$$\begin{aligned}\nabla_\theta \ln d^\pi(s) &= \frac{1}{d^\pi(s)} \sum_{s_{-1}\in\mathcal{S}} \sum_{a_{-1}\in\mathcal{A}} p(s|s_{-1},a_{-1})\pi_\theta(a_{-1}|s_{-1})d^\pi(s_{-1}) \\ &\qquad\qquad \{\nabla_\theta \ln \pi_\theta(a_{-1}|s_{-1}) + \nabla_\theta \ln d^\pi(s_{-1})\} \\ &= \sum_{s_{-1}\in\mathcal{S}} \sum_{a_{-1}\in\mathcal{A}} q_{B(\boldsymbol{\theta})}(s_{-1},a_{-1}|s)\{\nabla_\theta \ln \pi_\theta(a_{-1}|s_{-1}) + \nabla_\theta \ln d^\pi(s_{-1})\} \\ &= \mathbb{E}_{B(\boldsymbol{\theta})}\{\nabla_\theta \ln \pi_\theta(a_{-1}|s_{-1}) + \nabla_\theta \ln d^\pi(s_{-1})|s\}. \qquad (4.7)\end{aligned}$$

Noting that there exist $\nabla_\theta \ln d^\pi(s)$ and $\nabla_\theta \ln d^\pi(s_{-1})$ in Eq.4.7, the recursion of Eq.4.7 yields

$$\nabla_\theta \ln d^\pi(s) = \lim_{K\to\infty} \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{ \sum_{k=1}^{K} \nabla_\theta \ln \pi_\theta(a_{-k}|s_{-k}) + \nabla_\theta \ln d^\pi(s_{-K}) \Big| s \Big\}. \qquad (4.8)$$

Eq.4.8 implies that the LSD of a state s is the infinite-horizon cumulation of the policy eligibility $\nabla_\theta \ln \pi_\theta(a|s)$ through the backward Markov chain $B(\boldsymbol{\theta})$ from state s. From Eqs.4.7 and 4.8, LSD could be estimated with temporal difference (TD) learning (Sutton, 1988) concerning the following backward TD $\boldsymbol{\delta}$ on the backward Markov chain $B(\boldsymbol{\theta})$ rather than $M(\boldsymbol{\theta})$.

$$\boldsymbol{\delta}(s) \equiv \nabla_\theta \ln \pi_\theta(a_{-1}|s_{-1}) + \nabla_\theta \ln d^\pi(s_{-1}) - \nabla_\theta \ln d^\pi(s),$$

where the first two terms are regarded as the one-step actual observation of the policy eligibility and the one-step ahead LSD on $B(\boldsymbol{\theta})$ against the LSD of current state, which is the last term [5].. While $\boldsymbol{\delta}(s)$ is a random variable, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}(s)|s\} = \mathbf{0}$ holds. It motivates the minimization of the mean squares of the backward TD-error, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}(s)^2\}$ for the estimation of LSD, where $\hat{\boldsymbol{\delta}}(s)$ is comprised by the LSD estimate $\widehat{\nabla}_\theta \ln d^\pi(s)$ rather than LSD $\nabla_\theta \ln d^\pi(s)$. $\boldsymbol{\delta}(s)^2$ denotes $\boldsymbol{\delta}(s)^\top \boldsymbol{\delta}(s)$ for simplicity.

[5] While the TD for the value functions is well-known and concerns r on $M(\boldsymbol{\theta})$ (Sutton and Barto, 1998), this TD for LSD concerns $\nabla_\theta \ln \pi_\theta(a|s)$ on $B(\boldsymbol{\theta})$.

With an eligibility decay rate $\lambda \in [0,1]$ and a backtrace time-step $K \in \mathcal{N}$, Eq.4.8 is generalized, where $\mathcal{N}$ denotes the set of natural numbers:

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

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

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

where the unbiased property, $\mathbb{E}_{B(\boldsymbol{\theta})}\{\boldsymbol{\delta}_{\lambda,K}(s)|s\} = \mathbf{0}$, is still retained. The minimization of $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2\}$ in $\lambda = 1$ and the limit $K \to \infty$ is regarded as the Widrow-Hoff supervised learning procedure. Even in a larger λ and K instead of the above setting, this minimization would be less sensitive to a non-Markovian effect as in the case of the conventional TD(λ) learning for the value functions (Peng and Williams, 1996).

In order to minimize $\mathbb{E}_{B(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2\}$ as the estimation of LSD, we need to gather many samples drawn from the backward Markov chain $B(\boldsymbol{\theta})$; however, actual samples are drawn from a forward Markov chain $M(\boldsymbol{\theta})$. Fortunately, by utilizing Propositions 5 and 6, we can use the following exchangeable property:

$$\begin{aligned}\mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2\Big\} &= \sum_{s\in\mathcal{S}} d_{B(\boldsymbol{\theta})}(s)\, \mathbb{E}_{B(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2|s\Big\} \\ &= \sum_{s\in\mathcal{S}} \;\; d^\pi(s) \;\; \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2|x, d^\pi(s_{-K})\Big\} \\ &= \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\hat{\boldsymbol{\delta}}_{\lambda,K}(s)^2|d^\pi(s_{-K})\Big\}. \end{aligned} \tag{4.9}$$

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

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

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

$$\mathbb{E}_{M(\boldsymbol{\theta})}\{\nabla_\theta \ln d^\pi(s)\} = \sum_{s\in\mathcal{S}} d^\pi(s)\nabla_\theta \ln d^\pi(s) = \nabla_\theta \sum_{s\in\mathcal{S}} d^\pi(s) = \mathbf{0}. \tag{4.10}$$

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

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

where $\boldsymbol{\phi}(s) \in \mathcal{R}^e$ is a basis function and $\boldsymbol{\Omega} \equiv [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_d]^\top \in \mathcal{R}^{d\times e}$ is an adjustable parameter matrix, and assume that the optimal parameter $\boldsymbol{\Omega}^*$ satisfies $\nabla_\theta \ln d^\pi(s) = \boldsymbol{\Omega}^*\boldsymbol{\phi}(s)$ [6]. For simplicity, we focus our attention only on the i'th element θ_i of the policy parameter $\boldsymbol{\theta}$, notating $f(s;\boldsymbol{\omega}_i) \equiv \boldsymbol{\omega}_i^\top \boldsymbol{\phi}(s)$ and $\nabla_{\theta_i} \ln \pi_\theta(a|s) \equiv \partial \ln \pi_\theta(a|s)/\partial\theta_i$ and $\hat{\delta}_{\lambda,K}(s,\boldsymbol{\omega}_i)$ as the i'th element of $\hat{\boldsymbol{\delta}}_{\lambda,K}(s)$. Accordingly, the objective function to be minimized is

$$\varepsilon(\boldsymbol{\omega}_i) = \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)^2 | d^\pi(s_{-K})\} + \frac{1}{2}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(s;\boldsymbol{\omega}_i)\}^2, \tag{4.11}$$

[6] If the estimator cannot represent LSD exactly, $\mathcal{LS}$LSD(λ) would behave as suggested by Sutton (1988); Peng and Williams (1996), which will be confirmed in a numerical experiment. However, we do not analyze it theoretically.

where the second term of the right side is for the constraint of Eq.4.10 [7]. Then, the derivative is

$$\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) = \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)\ \nabla_{\omega_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)|d^{\pi}(s_{-K})\} + \frac{1}{2}\nabla_{\omega_i}\mathbb{E}_{M(\boldsymbol{\theta})}\{f(s;\boldsymbol{\omega}_i)\}^2, \tag{4.12}$$

where

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

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

Although the conventional least squares method aims to find the parameter satisfying $\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i) = \mathbf{0}$ as the true parameter $\boldsymbol{\omega}_i^*$, it induces estimation bias if a correlation exists between the error $\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ and its derivative $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ concerning the first term of the right-hand side in Eq.4.11. That is, if

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

then $\nabla_{\boldsymbol{\omega}_i}\varepsilon(\boldsymbol{\omega}_i^*) \neq \mathbf{0}$. Since this correlation exists in general RL problems, we apply the instrumental variable method to eliminate the bias (Young, 1984; Bradtke and Barto, 1996). It requires that $\nabla_{\boldsymbol{\omega}_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)$ is replaced by the instrumental variables $\boldsymbol{\iota}(s)$ that has a correlation with $\nabla_{\omega_i}\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$ but not $\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i^*)$. This condition is obviously satisfied when $\boldsymbol{\iota}(s) = \boldsymbol{\phi}(s)$ as well as LSTD(λ) (Bradtke and Barto, 1996; Boyan, 2002). Instead of Eq.4.12, we aim to find the parameter making the equation

$$\widetilde{\nabla}_{\omega_i}\varepsilon(\omega_i) \equiv \mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\delta}_{\lambda,K}(s;\boldsymbol{\omega}_i)\boldsymbol{\phi}(s)|d^{\pi}(s_{-K})\} + \mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(s)\}\mathbb{E}_{M(\boldsymbol{\theta})}\{\boldsymbol{\phi}(s)\}^{\top}\boldsymbol{\omega}_i \tag{4.13}$$

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

From here, we change the notation to s_t denoting the state at time-step t on the actual Markov chain $M(\boldsymbol{\theta})$. The proposed LSD estimation algorithm,

[7] As $\mathcal{LS}$LSD(λ) consider the two objectives in equal measure, we can instantly extend it for the problem minimizing $\mathbb{E}_{M(\boldsymbol{\theta})}\{\hat{\boldsymbol{\delta}}_{\lambda}^2(x)|d^{\pi}(s_{-K})\}$ subject to the constraint of Eq.4.10 with the Lagrange multiplier method.

$\mathcal{LS}$LSD(λ) sets that the backtrace time-step K is equal to the time-step t of the current state s_t under the eligibility decay rate $\lambda \in [0, 1)$. That is,

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

where $g_{\lambda,i}(s_t) = \sum_{k=0}^{t} \lambda^{t-k} \nabla_{\theta_i} \ln \pi_\theta(a_k|s_k)$ and $\boldsymbol{z}_\lambda(s_t) = (1-\lambda)\sum_{k=1}^{t} \lambda^{t-k}\boldsymbol{\phi}(s_k) + \lambda^t \boldsymbol{\phi}(s_0)$. The expectations in Eq.4.13 are estimated by [8]

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

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

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

Therefore, by substituting these estimators to Eq.4.13, the estimate $\hat{\boldsymbol{\omega}}_i^*$ at time-step T is computed by

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

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

[8] When the limit $T \to \infty$ at $\lambda \in [0, 1)$, these estimators converge to the true values. Although it could be proven based on the results of Bradtke and Barto (1996); Boyan (2002), we omit the proof here.

Algorithm 1
$\mathcal{LS}$LSD(λ): Estimation for $\nabla_\theta \ln d^\pi(s)$

```
Given:
  • a policy $\pi_\theta(a|s)$ with a fixed $\boldsymbol{\theta}$,
  • a feature vector function of state $\boldsymbol{\phi}(s)$.
Initialize: $\lambda \in [0,1)$.
Set: $\boldsymbol{c} := \mathbf{0}$; $\boldsymbol{z} \ = \mathbf{0}$; $\boldsymbol{g} := \mathbf{0}$; $\boldsymbol{A} := \mathbf{0}$; $\boldsymbol{B} := \mathbf{0}$.
  for $t = 0$ to $T-1$ do
    if $t = 0$ then
      $\boldsymbol{z} := \boldsymbol{\phi}(s_0)$;  $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$;
    else
      $\boldsymbol{z} := \lambda\boldsymbol{z} + (1-\lambda)\boldsymbol{\phi}(s_t)$;
    end if
    $\boldsymbol{c} := \boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
    $\boldsymbol{g} := \lambda\boldsymbol{g} + \nabla_\theta \ln \pi_\theta(a_t|s_t)$;
    $\boldsymbol{A} := \boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
    $\boldsymbol{B} := \boldsymbol{B} + \boldsymbol{\phi}(s_{t+1})\boldsymbol{g}^\top$;
  end for
  $\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top/t)^{-1}\boldsymbol{B}$;
Return: $\widehat{\nabla}_{\boldsymbol{\theta}} \ln d^\pi(s) = \boldsymbol{\Omega}\,\boldsymbol{\phi}(s)$.
```

4.3 Policy update with the LSD estimate

Now let us define the PGRL algorithm based on the above LSD estimate. The realization of the estimation for $\nabla_\theta \ln d^\pi(s)$ by $\mathcal{LS}$LSD(λ) instantly derives the following estimate for the PG (eq.4.1), being independent of the discount factor γ:

$$\nabla_\theta R(\boldsymbol{\theta}) \simeq \frac{1}{T}\sum_{t=0}^{T-1} \left(\nabla_\theta \ln \pi_\theta(a_t|s_t) + \nabla_\theta \ln d^\pi(s_t)\right) r(s_t, a_t, s_{t+1}) \tag{4.14}$$

$$\simeq \frac{1}{T}\sum_{t=0}^{T-1} \Big(\nabla_\theta \ln \pi_\theta(a_t|s_t) + \widehat{\nabla}_\theta \ln d^\pi(s_t)\Big) r(s_t, a_t, s_{t+1}) \tag{4.15}$$

The policy parameter can then be updated through the stochastic gradient method with an appropriate stepsize α (Bertsekas and Tsitsiklis, 1996):[9]

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

where := denotes the substitution of the right to the left and r_{t+1} is the immediate reward defined by the reward function $r(s_t, a_t, s_{t+1})$. $\mathcal{LS}$LSD(λ)-PG without baseline function is shown at Algorithm 2 as one of the simplest realizations on PG algorithm, utilizing $\mathcal{LS}$LSD(λ). In algorithm 2, the forgetting rate parameter $\beta \in [0, 1)$ is introduced to discard the past estimate given by old values of $\boldsymbol{\theta}$.

```
Algorithm 2
  $\mathcal{LS}$LSD($\lambda$)-PG: Optimization for the policy
  without baseline function
Given:
  • a policy $\pi_{\theta}(a_t|s_t)$ with an adjustable $\boldsymbol{\theta}$,
  • a feature vector function of state $\boldsymbol{\phi}(s)$.
Initialize: $\boldsymbol{\theta}$, $\lambda \in [0, 1)$, $\beta \in [0, 1)$, $\alpha_t$.
Set: $\boldsymbol{c} := \mathbf{0}$; $\boldsymbol{z} = \mathbf{0}$; $\boldsymbol{g} := \mathbf{0}$; $\boldsymbol{A} := \mathbf{0}$; $\boldsymbol{B} := \mathbf{0}$.
  for $t = 0$ to $T - 1$ do
    if $t = 0$ then
      $\boldsymbol{z} := \boldsymbol{\phi}(s_0)$;  $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$;
    else
      $\boldsymbol{z} := \lambda \boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(s_t)$;
      $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t\{\nabla_{\theta} \ln \pi_{\theta}(a_t|s_t) + \boldsymbol{\Omega}^{\top}\boldsymbol{\phi}(s_t)\}r_{t+1}$;
    end if
    $\boldsymbol{c} := \beta \boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
    $\boldsymbol{g} := \beta\lambda \boldsymbol{g} + \nabla_{\theta} \ln \pi_{\theta}(a_t|s_t)$;
    $\boldsymbol{A} := \beta \boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^{\top}$;
    $\boldsymbol{B} := \beta \boldsymbol{B} + \boldsymbol{\phi}(s_{t+1})\boldsymbol{g}^{\top}$;
    $\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^{\top}/\|\boldsymbol{c}\|)^{-1}\boldsymbol{B}$;
  end for
Return: $p(a|s; \boldsymbol{\theta}) = \pi_{\theta}(a|s)$.
```

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

There is the other important topic for function approximation: how to set the basis function $\boldsymbol{\phi}(s)$ of approximator, particularly in the continuous state problems. For the PG algorithm, the objective concerning LSD estimate is just to provide the estimate of PG $\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\nabla_\theta \ln d^{\pi}(s)\bar{r}(s,a)$, but not to provide the precise estimate of LSD $\nabla_\theta \ln d^{\pi}(s)$, where $\bar{r}(s,a)\equiv\sum_{s_{+1}\in\mathcal{S}} p(s_{+1}|s,a)r(s,a,s_{+1})$. Therefore, the following proposition would be useful:

Proposition 7 *Let the basis function of the LSD estimator be*

$$\phi(s)=\sum_{a\in\mathcal{A}}\pi_\theta(a|s)\bar{r}(s,a),$$

where $\bar{r}(s,a)\equiv\sum_{s_{+1}\in\mathcal{S}} p(s_{+1}|s,a)r(s,a,s_{+1})$, then the function estimator, $\boldsymbol{f}(s;\boldsymbol{\omega})=\boldsymbol{\omega}\sum_{a\in\mathcal{A}}\pi_\theta(a|s)\bar{r}(s,a)$, has the ability to represent the second term of the PG, $\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\nabla_\theta \ln d^{\pi}(s)\bar{r}(s,a)$, where the adjustable parameter $\boldsymbol{\omega}$ is a d dimensional vector:

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

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

Proof: It is proven by

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

□

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

Since the variance of PG estimates with LSD, eq.4.15, might be huge, we consider the variance reduction by using a baseline function. The following proposition provides what kind of functions can be used as the baseline function for the PG estimation with LSD.

Proposition 8 *With the following function of the state s and the following state s_{+1} on $M(\boldsymbol{\theta})$,*

$$\rho(s, s_{+1}) = c + g(s) - g(s_{+1}), \tag{4.16}$$

where c and $g(S)$ are an arbitrary constant and an arbitrary function of the state, respectively, the derivative of the average reward $R(\boldsymbol{\theta})$ with respect to the policy parameter $\boldsymbol{\theta}$ (eq.4.1), $\nabla_\theta R(\boldsymbol{\theta})$, is transformed to

$$\begin{aligned}\nabla_\theta R(\boldsymbol{\theta}) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s_{+1}|s,a) \\ &\qquad \{\nabla_\theta \ln \pi_\theta(a|s) + \nabla_\theta \ln d^\pi(s)\}\, r(s,a,s_{+1}) && (4.1) \\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s_{+1}|s,a) \\ &\qquad \{\nabla_\theta \ln \pi_\theta(a|s) + \nabla_\theta \ln d^\pi(s)\}\, \{r(s,a,s_{+1}) - \rho(s,s_{+1}).\} && (4.17)\end{aligned}$$

Proof: If the following equation is proved,

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s_{+1}|s,a)\, \{\nabla_\theta \ln \pi_\theta(a|s) + \nabla_\theta \ln d^\pi(s)\}\, \rho(s,s_{+1}) = \mathbf{0} \tag{4.18}$$

the transformation to eq.4.18 obviously holds. Because of eq.4.16 and

$$\begin{cases} \sum_{a\in\mathcal{A}} \pi_\theta(a|s)\nabla_\theta \ln \pi_\theta(a|s)\, c = \nabla_\theta c = \mathbf{0}, \\ \sum_{s\in\mathcal{S}} d^\pi(s)\nabla_\theta \ln d^\pi(s)\, c \qquad = \nabla_\theta c = \mathbf{0}, \end{cases}$$

$$\begin{aligned} &-\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s_{+1}|s,a)\, \{\nabla_\theta \ln \pi_\theta(a|s) + \nabla_\theta \ln d^\pi(s)\}\, \rho(s,s_{+1}) \\ &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s_{+1}\in\mathcal{S}} d^\pi(s)\pi_\theta(a|s)p(s_{+1}|s,a)\, \{\nabla_\theta \ln \pi_\theta(a|s) + \nabla_\theta \ln d^\pi(s)\}\, \{g(s_{+1}) - g(s)\} \end{aligned} \tag{4.19}$$

holds. Since a time average is equivalent to a state-action space average in ergodic Markov chain $M(\boldsymbol{\theta})$ by eq.2.3 and eq.2.2, eq.4.19 is transformed to

$$
\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\{\nabla_\theta\ln\pi_\theta(a_t|s_t)+\nabla_\theta\ln d^\pi(s_t)\}\{g(s_{t+1})-g(s_t)\}\\
&=\sum_{s_{-1}\in\mathcal{S}}\sum_{a_{-1}\in\mathcal{A}}\sum_{s\in\mathcal{S}}d^\pi(s_{-1})\pi_\theta(a_{-1}|s_{-1})p(s|s_{-1},a_{-1})\pi_\theta(a|s)\\
&\qquad\{\nabla_\theta\ln\pi_\theta(a_{-1}|s_{-1})+\nabla_\theta\ln d^\pi(s_{-1})-\nabla_\theta\ln\pi_\theta(a|s)-\nabla_\theta\ln d^\pi(s)\}\,g(s)\\
&=\sum_{s_{-1}\in\mathcal{S}}\sum_{a_{-1}\in\mathcal{A}}\sum_{s\in\mathcal{S}}d^\pi(s_{-1})\pi_\theta(a_{-1}|s_{-1})p(s|s_{-1},a_{-1})\\
&\qquad\{\nabla_\theta\ln\pi_\theta(a_{-1}|s_{-1})+\nabla_\theta\ln d^\pi(s_{-1})-\nabla_\theta\ln d^\pi(s)\}\,g(s)\\
&=\sum_{s\in\mathcal{S}}d^\pi(s)g(s)\sum_{s_{-1}\in\mathcal{S}}\sum_{a_{-1}\in\mathcal{A}}q_{B(\boldsymbol{\theta})}(s_{-1},a_{-1}|s)\{\nabla_\theta\ln\pi_\theta(a_{-1}|s_{-1})+\nabla_\theta\ln d^\pi(s_{-1})\}\\
&\qquad-\sum_{s\in\mathcal{S}}d^\pi(s)g(s)\nabla_\theta\ln d^\pi(s)\\
&=\sum_{s\in\mathcal{S}}d^\pi(s)g(s)\left[\mathbb{E}_{B(\boldsymbol{\theta})}\{\nabla_\theta\ln\pi_\theta(a_{-1}|s_{-1})+\nabla_\theta\ln d^\pi(s_{-1})|s\}-\nabla_\theta\ln d^\pi(s)\right]\\
&=\mathbf{0},
\end{aligned}
$$

where the final transformation is executed by eq.4.7. Therefore, eq.4.18 holds.

□

Proposition 8 means that any $\rho(s,s_{+1})$ defined in eq.4.16 can be used as the baseline function of immediate reward $r_{+1}\equiv r(s,a,s_{+1})$ for the computing the PG, as eq.4.17. Therefore, the PG can be estimated with baseline function $\rho(s,s_{+1})$ with large time-steps T,

$$
\begin{aligned}
\nabla_\theta R(\boldsymbol{\theta}) &\simeq \frac{1}{T}\sum_{t=0}^{T-1}(\nabla_\theta\ln\pi_\theta(a_t|s_t)+\nabla_\theta\ln d^\pi(s_t))\{r(s_t,a_t,s_{t+1})-\rho(s_t,s_{t+1})\}\\
&\equiv \widehat{\nabla}_\theta R(\boldsymbol{\theta}) \qquad (4.20)
\end{aligned}
$$

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

(2004) in chapter 3.3, an upper bound of the variance is derived as

$$\begin{aligned}&\mathrm{Var}_{\pi}\left[\widehat{\nabla}_{\theta}R(\boldsymbol{\theta})\right]\\&\leq \mathbb{E}_{M(\boldsymbol{\theta})}\left\{\|\nabla_{\theta}\ln\pi_{\theta}(a|s)+\nabla_{\theta}\ln d^{\pi}(s)\|^{2}(r(s,a,s_{+1})-\rho(s,s_{+1}))^{2}|s,s_{+1}\right\}+o \quad (4.21)\\&\equiv \sigma^{2}_{\widehat{\nabla}_{\theta}R(\boldsymbol{\theta})}(\rho(s,s_{+1}))+o. \quad (4.22)\end{aligned}$$

where o is independent term of $\rho(s,s_{+1})$. Accordingly, since the optimal baseline function $b^{*}(s,s_{+1})$ satisfies

$$\left.\frac{\partial\sigma^{2}_{\widehat{\nabla}_{\theta}R(\boldsymbol{\theta})}(\rho(s,s_{+1}))}{\partial\rho(s,s_{+1})}\right|_{\rho(s,s_{+1})=b^{*}(s,s_{+1})}=\mathbf{0}, \qquad {}^{\forall}s\in\mathcal{S},\ {}^{\forall}s_{+1}\in\mathcal{S},$$

the optimal baseline function $b^{*}(s,s_{+1})$ is computed as

$$b^{*}(s,s_{+1})=\frac{\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\|\nabla_{\theta}\ln\pi_{\theta}(a|s)+\nabla_{\theta}\ln d^{\pi}(s)\|^{2}r(s,a,s_{+1})\mid s,s_{+1}\right\}}{\mathbb{E}_{M(\boldsymbol{\theta})}\left\{\|\nabla_{\theta}\ln\pi_{\theta}(a|s)+\nabla_{\theta}\ln d^{\pi}(s)\|^{2}\mid s,s_{+1}\right\}}. \quad (4.23)$$

Meanwhile, there is the alternative decent baseline function $b(s,s_{+1})$

$$b(s,s_{+1})=\mathbb{E}_{M(\boldsymbol{\theta})}\left\{r(s,a,s_{+1})|s,s_{+1}\right\}, \quad (4.24)$$

which minimizes the residual sum of squares about $r(s,a,s_{+1})$ and corresponds to the state-value function in the case of the PG estimation with the value function.

When an approximator of the baseline function (eq.4.16) is parameterized as the following linear combination with a feature vector function of state, $\boldsymbol{\phi}(s)$, and a coefficient parameter $\boldsymbol{\upsilon}$,

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

both baseline functions, $b^{*}(s,s_{+1})$ and $b(s,s_{+1})$, are estimated by least squares, though the estimation for b^{*} requires LSD estimates. The $\mathcal{LS}$LSD(λ)-PG algorithm with baseline function is shown in algorithm 3 [10].

[10] Although the technique of eligibility traces is instantly applied for the baseline estimate, we omit it.

Algorithm 3

$\mathcal{LS}$LSD(λ)-PG: Optimization for the policy with "optimal" baseline function

```
Given:
  • a policy $\pi_\theta(a_t|s_t)$ with an adjustable $\boldsymbol{\theta}$,
  • a feature vector function of state $\boldsymbol{\phi}(s)$.
Define: $\boldsymbol{\psi}(s_t, s_{t+1}) \equiv [\boldsymbol{\phi}(s_t)^\top - \boldsymbol{\phi}(s_{t+1})^\top, 1]^\top$
Initialize: $\boldsymbol{\theta}$, $\lambda \in [0, 1)$, $\beta \in [0, 1)$, $\alpha_t$.
Set: $\boldsymbol{c} := \boldsymbol{\phi}(s_0)$; $\boldsymbol{z} := \boldsymbol{\phi}(s_0)/\beta$; $\boldsymbol{g} := \boldsymbol{0}$; $\boldsymbol{A} := \boldsymbol{0}$; $\boldsymbol{B} := \boldsymbol{0}$;
    $w := 1$; $\boldsymbol{X} := \boldsymbol{0}$; $\boldsymbol{y} := \boldsymbol{0}$;
  for $t = 0$ to $T - 1$ do
    if $t \geq 1$ then
      $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_t\{\nabla_\theta \ln \pi_\theta(a_t|s_t) + \boldsymbol{\Omega}^\top \boldsymbol{\phi}(s_t)\}\{r_{t+1} - \boldsymbol{\psi}(s_t, s_{t+1})^\top \boldsymbol{X}^{-1}\boldsymbol{y}\}$;
    end if
    $\boldsymbol{c} := \beta\boldsymbol{c} + \boldsymbol{\phi}(s_{t+1})$;
    $\boldsymbol{z} := \beta\lambda\boldsymbol{z} + (1 - \lambda)\boldsymbol{\phi}(s_t)$;
    $\boldsymbol{g} := \beta\lambda\boldsymbol{g} + \nabla_\theta \ln \pi_\theta(a_t|s_t)$;
    $\boldsymbol{A} := \beta\boldsymbol{A} + \boldsymbol{\phi}(s_{t+1})(\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{z})^\top$;
    $\boldsymbol{B} := \beta\boldsymbol{B} + \boldsymbol{\phi}(s_{t+1})\boldsymbol{g}^\top$;
    $\boldsymbol{\Omega} := (\boldsymbol{A} - \boldsymbol{c}\boldsymbol{c}^\top/\|\boldsymbol{c}\|)^{-1}\boldsymbol{B}$;
    "$w := \|\nabla_\theta \ln \pi_\theta(a_t|s_t) + \boldsymbol{\Omega}^\top \boldsymbol{\phi}(s_t)\|^2$;"
    $\boldsymbol{X} := \beta\boldsymbol{X} + w\boldsymbol{\psi}(s_t, s_{t+1})\boldsymbol{\psi}(s_t, s_{t+1})^\top$;
    $\boldsymbol{y} := \beta\boldsymbol{y} + w\boldsymbol{\psi}(s_t, s_{t+1})r_{t+1}$;
  end for
Return: $p(a|s; \boldsymbol{\theta}) = \pi_\theta(a|s)$.
```

$*$ In the case of the (decent) baseline function $b(s, s')$, instead of $b^*(s, s')$, all the content of "$\cdots$" in the algorithm are omited.

4.4 Numerical Experiments

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

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

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

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

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

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

At first, we verified how precisely $\mathcal{LS}$LSD(λ) algorithm estimates $\nabla_\theta \ln d^\pi(s)$ regardless of the setting of q_s and the policy parameter $\boldsymbol{\theta}$. The each element of $\boldsymbol{\theta}$ and the each task-dependent constant q_s were randomly initialized according to $\mathtt{N}(\mu{=}0,\, \sigma^2{=}0.5^2)$ and $\mathtt{U}(a{=}.7,\, b{=}1)$, respectively, where $\mathtt{U}(a{=}.7,\, b{=}1)$ is the uniform distribution over the interval of $[a, b]$. These were fixed during each episode.

Figure 4.1(A) shows a typical time course of the LSD estimate $\widehat{\nabla}_\theta \ln d^\pi(s)$ in case of $|\mathcal{S}| = 3$-state MDP, where nine different colors indicate all different elements of LSD, respectively. The solid lines denote the values estimated by $\mathcal{LS}$LSD(0), and the dotted lines denote the analytical solution of LSD. This

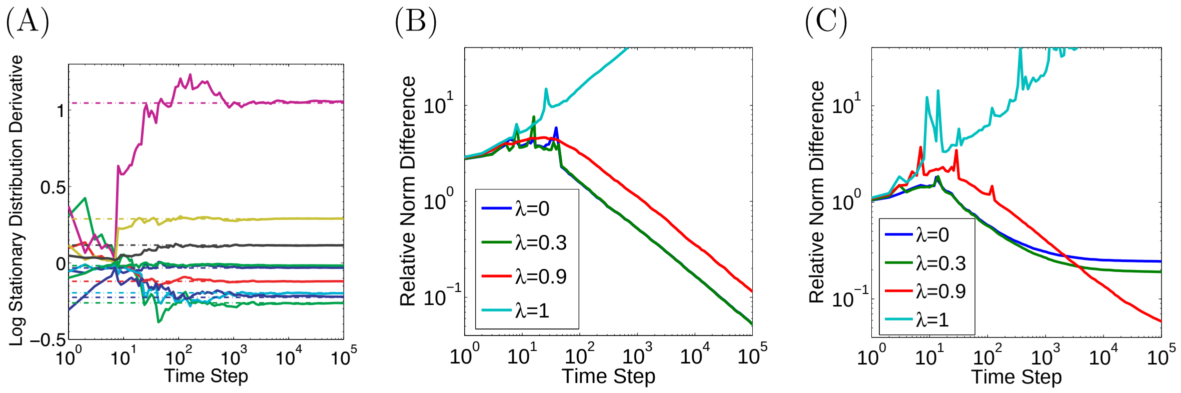


Figure 4.1. Performances of $\mathcal{LS}$LSD(λ) for the estimation of LSD $\nabla_\theta \ln d^\pi(s)$. (A) A typical time course of LSD estimate in a 3-state MDP. (B, C) The relative errors averaged over 200 episodes in 7-state MDPs for various λs; (B) with proper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^7$, (C) with improper basis function $\boldsymbol{\phi}(s) \in \mathcal{R}^6$.

result demonstrates that the proposed LS-LSD algorithm could estimate LSD $\nabla_\theta \ln d^\pi(s)$. We also confirmed that the estimates by $\mathcal{LS}$LSD(0) always converged to the analytical solution at $|\mathcal{S}| = 3$ as the result in Figure 4.1(A), though these are not given here.

Second, we investigated the effect of the eligibility decay rate λ using 7-state MDPs. In order to evaluate the average performance over various settings, we employed a "relative error" criterion that is defined by $\mathbb{E}_{M(\boldsymbol{\theta})}\{(\boldsymbol{f}(x;\boldsymbol{\Omega}^\star) - \boldsymbol{f}(x;\boldsymbol{\Omega}))^2\}/\mathbb{E}_{M(\boldsymbol{\theta})}\{(\boldsymbol{f}(x;\boldsymbol{\Omega}^\star)^2\}$, where $\boldsymbol{\Omega}^\star$ is the optimal parameter defined in Proposition 7. Figure 4.1(B) and (C) show the time courses of relative error averages over 200 episodes for $\lambda = 0$, 0.3, 0.9, and 1. The only difference between these two figures was the number of elements of the feature-vectors $\boldsymbol{\phi}(s)$. The feature-vectors $\boldsymbol{\phi}(s) \in \mathcal{R}^7$ used in (B) were appropriate and enough to distinguish all the different states, while the feature-vectors $\boldsymbol{\phi}(s) \in \mathcal{R}^6$ used in (C) were inappropriate and deficient. These results were consistent with theoretical prospects. Namely, we could set λ arbitrarily in $[0, 1)$ if the basis function was appropriate (Figure 4.1 (B)), otherwise we would need to set λ close to 1 except for $\lambda = 1$ (Figure 4.1 (C)).

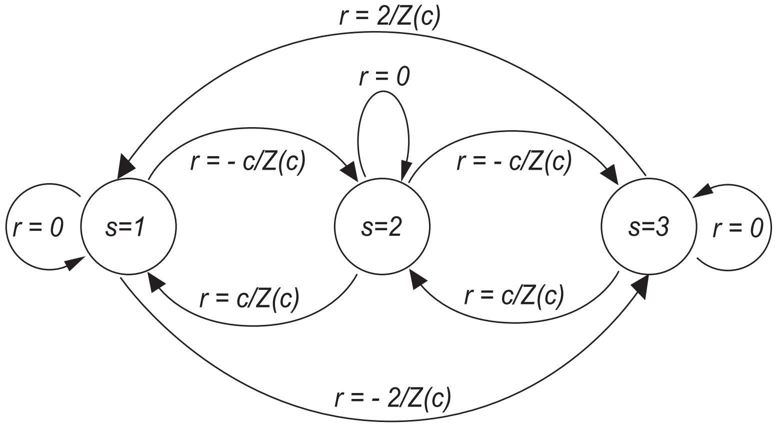


Figure 4.2. Reward setting of 3-state MDPs used in our comparative studies. c is selected by the uniform distribution $\mathtt{U}[0.95, 1)$ for each simulation run. $Z(c)$ is a normalizing function to assure $\mathtt{max}_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = 1$.

4.4.2 Comparison to other PG methods

We compared the $\mathcal{LS}$LSD(λ=0)-PG algorithm with the other PG algorithm in 3-state MDPs, concerned with the estimation of PG $\nabla_\theta R(\boldsymbol{\theta})$ and the optimization of the policy parameter $\boldsymbol{\theta}$. The policy and the state transition probability were set as each $\theta_i \sim \mathtt{N}(0, 0.5^2)$ and $q_i \sim \mathtt{U}[0.95, 1]$ for every $i \in \{1, 2, 3\}$, respectively. Figure 4.2 shows the reward setting in the MDP. There are two types of rewards: "$r = (\pm)2/Z(c)$" and "$r = (\pm)c/Z(c)$", where the variable c was initialized by the uniform distribution over $[0.95, 1)$ for each episode and the function $Z(c)$ was the normalizing constant to assure $\mathtt{max}_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = 1$. Note that the reward c defines the minimum value of γ to find the optimal policy: $\gamma^2 + \gamma > \frac{2c}{2-c}$. Therefore, the setting of γ is important and difficult in this task. From the performance baselines of the existing PG methods, we adopted two algorithms: GPOMDP (Baxter and Bartlett, 2001) and Konda's actor-critic (Konda and Tsitsiklis, 2003). These algorithm used the baseline function being state value estimates which were estimated by LSTD(0) (Bradtke and Barto, 1996; Boyan, 2002; Yu and Bertsekas, 2006), while these original did not use the baseline function.

Figure 4.3 shows the results about the estimation of PG $\nabla_\theta R(\boldsymbol{\theta})$ by eq.4.14. The forgetting rates for the sufficient statistics were set as $\beta = 1$ for all algorithms.

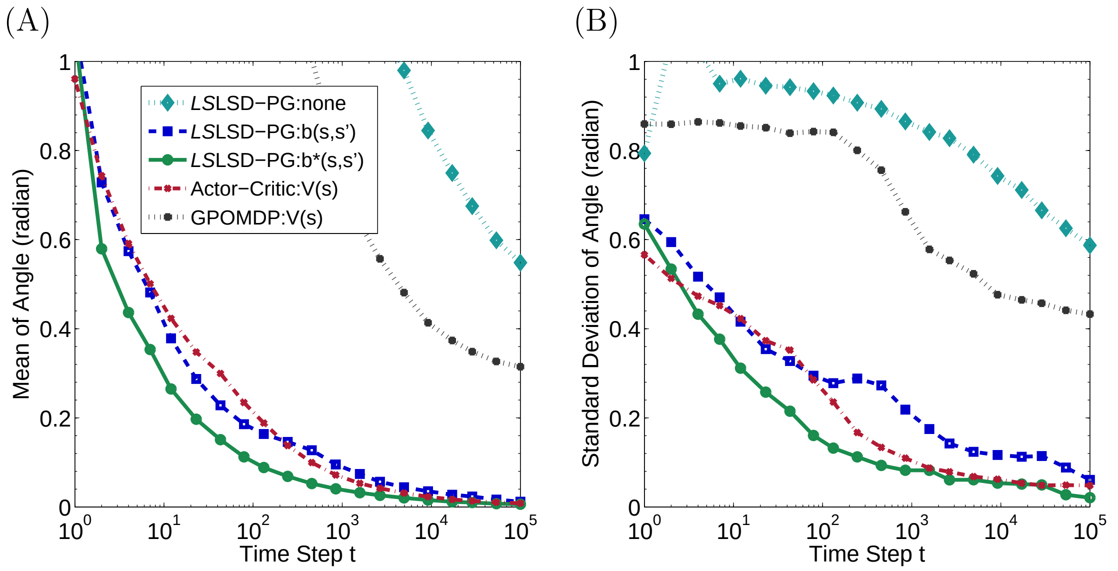


Figure 4.3. Comparison with various PG algorithms about the estimation of the PG over 2500 episodes: (A) and (B) are the mean and the standard deviation of angles between the estimates and the exact PG, respectively

(A) and (B) represents the mean and the standard deviation of angles between the estimates and the exact PG, respectively. These results was that $\mathcal{LS}$LSD-PG with estimating the optimal baseline function $b^*(s, s_{+1})$, termed $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$, worked best to estimate the PG.

Finally, we examined the optimization of the policy parameter $\boldsymbol{\theta}$, i.e. the average reward, by these PG methods. In this experiment, the forgetting rate was set as $\beta = 0.99$. In order to avoid the effect from poor estimations of the functions for the PG estimate, there was pre-learning period of 50 time-steps, where the learning rate α was set to zero. Figures 4.4 shows the comparison with PG algorithms about various learning rate α over independent 1000 simulation runs (episodes). It is confirmed that $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$ worked best except for the high learning rate, in which the learning speed of $b^*(s, s_{+1})$ could not properly follow the changes of the policy rather than that of $b(s, s_{+1})$. Figure 4.5 shows the time courses of the average reward, where we chosen appropriate learning rates for the PG algorithms by drawing upon the previous results; $\alpha = .16$ in $\mathcal{LS}$LSD-PG:$b(s, s_{+1})$, $\alpha = .08$ in $\mathcal{LS}$LSD-PG:$b^*(s, s_{+1})$, $\alpha = .08$ in Actor-Critic:$V(s)$, and

$\alpha = .007$ in GPOMDP:$V(s)$. This result also indicates that our $\mathcal{LS}$LSD-PG algorithm with the optimal baseline function $b^*(s, s_{+1})$ outperformed the other PG algorithms, since the algorithm increased the average reward and suppressed its standard deviation most efficiently.

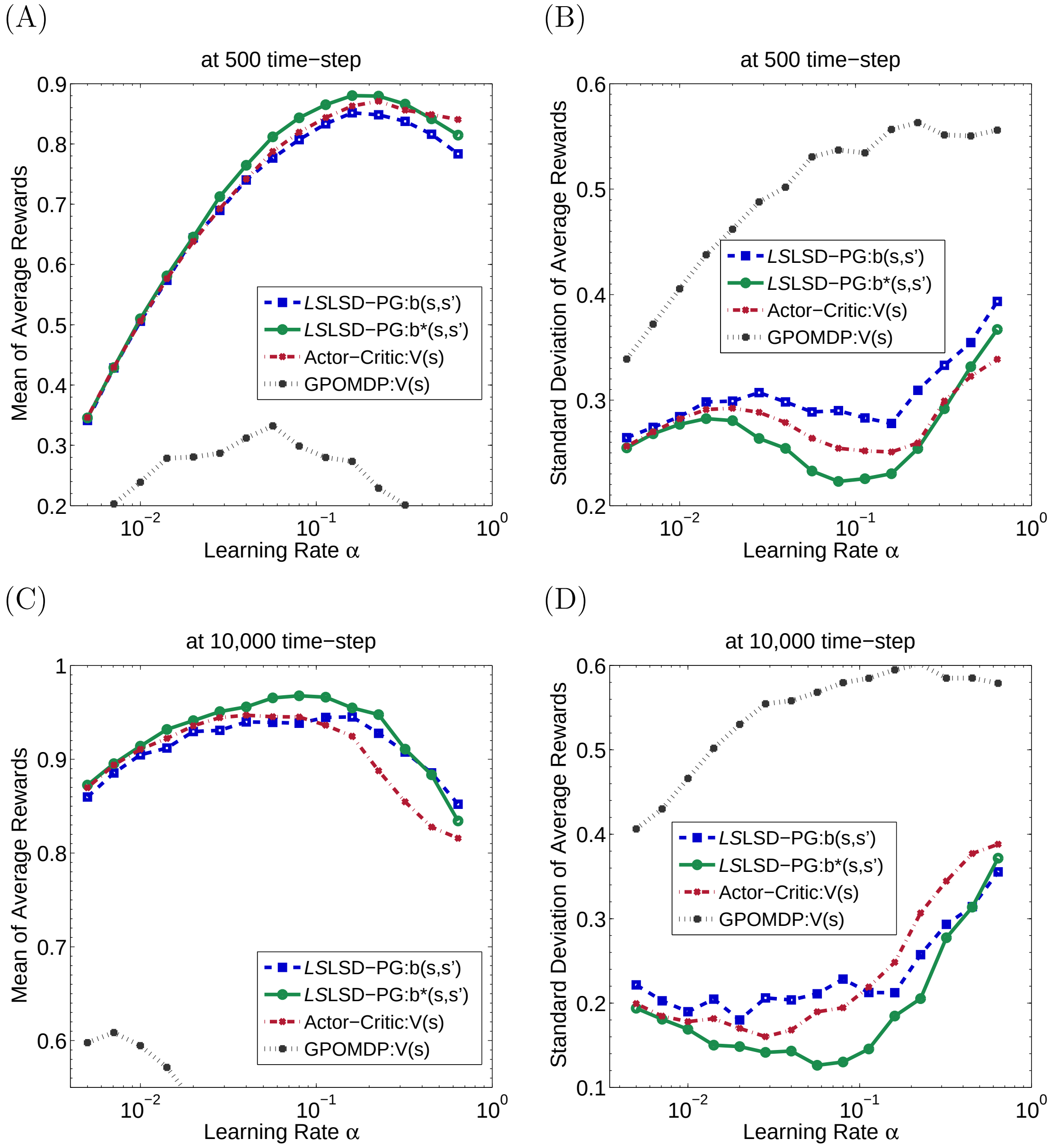


Figure 4.4. Comparison with various PG algorithms about the optimization of the policy parameter with various learning rates over 1000 episodes.

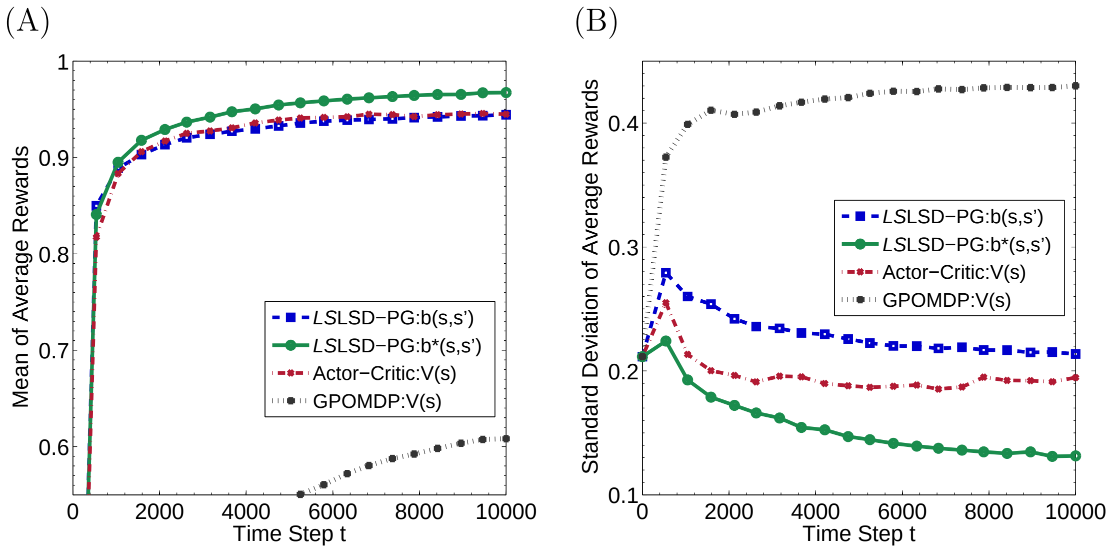


Figure 4.5. Comparison with various PG algorithms about the optimization of the policy parameter with the appropriated learning rate over 1000 episodes.

4.5 Summary and Discussion

We showed that the actual forward and backward Markov chains are closely related and have common properties in the propositions. Utilizing these, we proposed $\mathcal{LS}$LSD(λ) as the estimation algorithm of the log stationary distribution derivative (LSD), and $\mathcal{LS}$LSD(λ)-PG as the PG algorithm utilizing the LSD estimate. The experimental results also demonstrated that $\mathcal{LS}$LSD(λ) could work at $\lambda \in [0, 1)$ and $\mathcal{LS}$LSD(λ)-PG could learn regardless of the task's requirment of the smallest value of γ to optimize the average reward. However, it has been suggested that there is theoretically no significant difference in performances between the average reward based PG methods and the regular based PG methods with discount factor γ close to 1 (Tsitsiklis and Van Roy, 2002). It might hold true in the case of our proposed PG, $\mathcal{LS}$LSD-PG.

The realization of LSD estimation opens up new possibility of a natural policy gradient (NPG) learning. That is, it enables model-free computation of an alternative, effective Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta})$ for NPG especially in the large scale MDP, which is proposed in chapter 5: for $\iota \in [0.1]$

$$\boldsymbol{G}(\boldsymbol{\theta}) := \mathbb{E}_{M(\boldsymbol{\theta})}\Big\{\nabla_\theta \ln \pi_\theta(a|s)\nabla_\theta \ln \pi_\theta(a|s)^\top + \iota\nabla_\theta \ln d^\pi(s)\nabla_\theta \ln d^\pi(s)^\top\Big\}.$$

This realization of LSD estimation would also open novel methods for the trade-off problem between exploration and exploitation. This is because LSD gives statistical information how much a change of the state stationary distribution is caused by the perturbation of each element of policy parameter, while an awful biasing of the stationary distribution would make the exploration hard.

Chapter 5

Natural Policy Gradients on Valid Riemannian Metrics

Amari (1998) proposed the concept of the natural gradient. Kakade (2002) derived a natural policy gradient by applying the natural gradient to the policy gradient reinforcement learning (RL). Since the natural gradient depends on the applied Riemannian metric, the design of the metric is an important issue. However, the only Riemannian metric for RL, proposed by Kakade, takes into account only changes in the action distribution for improving the policy parameter and omits changes in the state distribution, which also depends on the policy in almost all cases. In this chapter, we propose a new Riemannian metric considering the state distribution as well as the action distribution and, based on the metric, derive a new robust natural policy gradient named "*Natural Stationary policy Gradient*" (NSG). We also prove that NSG becomes equal to the adjustable parameter of the linear function approximator with the basis function defined by the policy parameter, if the linear function approximates the immediate rewards. In the numerical experiments with Markov decision problems with varying number of states, we showed that the proposed method in comparison to previous studies, improved the performances especially in cases of a large number of states.

5.1 Background of natural policy gradients

In section 5.1.1, we briefly review the concept of natural gradients (NGs) proposed by Amari (1998) and the natural policy gradient (NPG) as NG for PGRL. In section 5.1.2, we introduce the controversy of NPGs.

5.1.1 Natural gradient (Amari, 1998)

Natural gradient learning is a gradient method on a Riemannian space. The parameter space being a Riemannian space implies that the parameter $\boldsymbol{\theta} \in \mathcal{R}^d$ is on the Riemannian manifold defined by the Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta}) \in \mathcal{R}^{d \times d}$ (positive definite matrix) and the squared length of a small incremental vector $\varDelta\boldsymbol{\theta}$ connecting $\boldsymbol{\theta}$ to $\boldsymbol{\theta} + \varDelta\boldsymbol{\theta}$ is given by

$$\|\varDelta\boldsymbol{\theta}\|^2_{\boldsymbol{G}} = \sum_{i=1}^{d}\sum_{j=1}^{d} g_{i,j}(\boldsymbol{\theta}) d\theta_i d\theta_j = d\boldsymbol{\theta}^\top \boldsymbol{G}(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

where $g_{i,j}$ is the $[i, j]$-th element of matrix $\boldsymbol{G}$ [1]. Under the constraint $\|\varDelta\boldsymbol{\theta}\|^2_{\boldsymbol{G}} = \varepsilon^2$ for a sufficiently small constant ε, the steepest ascent direction of a function $R(\boldsymbol{\theta})$ is given by

$$\widetilde{\nabla}_{\boldsymbol{G},\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \boldsymbol{G}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}). \tag{5.1}$$

It is called the natural gradient of R in a Riemannian space. In RL, the parameter $\boldsymbol{\theta}$ is the policy parameter, the function $R(\boldsymbol{\theta})$ is the average reward, and the gradient is called the natural policy gradient (NPG) (Kakade, 2002). Accordingly, in order to (locally) maximize $R(\boldsymbol{\theta})$, $\boldsymbol{\theta}$ is incrementally updated by

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha\, \widetilde{\nabla}_{\boldsymbol{G},\boldsymbol{\theta}} R(\boldsymbol{\theta}), \tag{5.2}$$

where is η is the learning rate.

When we consider a statistical model of a variable x defined by a parameter $\boldsymbol{\theta}$, $\Pr(x|\boldsymbol{\theta})$, the Fisher information matric (FIM) $\boldsymbol{F}_x(\boldsymbol{\theta})$ is often selected as as the

[1] When $\boldsymbol{G}$ is the unit matrix, the parameter space is called a Euclidean space, especially.

Riemannian metric matrix:[2]

$$\boldsymbol{F}_x(\boldsymbol{\theta}) \equiv \sum_{x\in\mathcal{X}} \Pr(x|\boldsymbol{\theta})\, \nabla_\theta \ln \Pr(x|\boldsymbol{\theta})\, \nabla_\theta \ln \Pr(x|\boldsymbol{\theta})^\top$$
$$= -\sum_{x\in\mathcal{X}} \Pr(x|\boldsymbol{\theta})\, \nabla_\theta^2 \ln \Pr(x|\boldsymbol{\theta}), \qquad (5.3)$$

where $\mathcal{X}$ is a set of possible values taken by x. $\nabla_\theta^2 a_{\boldsymbol{\theta}}$ denotes $\nabla_\theta(\nabla_\theta a_{\boldsymbol{\theta}})$. The reason for using $\boldsymbol{F}(\boldsymbol{\theta})$ as $\boldsymbol{G}(\boldsymbol{\theta})$ comes from the fact that $\boldsymbol{F}(\boldsymbol{\theta})$ is a unique metric matrix of the second-order Taylor expansion of Kullback-Leibler (KL) divergence (Amari and Nagaoka, 2000)[3], i.e.,

$$D_{\mathrm{KL}}\{\Pr(x|\boldsymbol{\theta})|\Pr(x|\boldsymbol{\theta}{+}\Delta\boldsymbol{\theta})\} = \frac{1}{2}\Delta\boldsymbol{\theta}^\top \boldsymbol{F}_x(\boldsymbol{\theta})\,\Delta\boldsymbol{\theta} + O(\|\Delta\boldsymbol{\theta}\|^3),$$

where $\|\boldsymbol{a}\|$ denotes the Euclidean norm of a vector $\boldsymbol{a}$.

5.1.2 Controversy of natural policy gradients

Policy gradient reinforcement learning (PGRL) is regarded as an optimizing process of the policy parameter $\boldsymbol{\theta}$ on some statistical models relevant to both a stochastic policy $\pi_\theta(a|s)$ and a state transition probability $p(s'|s,a)$. If a Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta})$ can be designed on the basis of the FIM of an apposite statistical model, $\boldsymbol{F}^*(\boldsymbol{\theta})$, an efficient NPG $\widetilde{\nabla}_{\boldsymbol{F}^*,\boldsymbol{\theta}} R(\boldsymbol{\theta})$ is instantly derived by eq.5.1. Since the natural policy gradient method is the gradient descent in the Riemannian space defined by $\boldsymbol{G}(\boldsymbol{\theta})$ rather than the space defined by an arbitrarily-parameterized policy, it is very efficient to use the NPG with a valid Riemannian metric for PGRL.

As Kakade (2002) pointed out, the choice of the Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta})$ for PGRL is not unique and the question what metric is apposite to $\boldsymbol{G}(\boldsymbol{\theta})$ is still open. Therefore, it is much important to discuss what is an appropriate Riemannian metric. Nevertheless, all previous studies on NPGs (Bagnell and Schneider, 2003; Peters et al., 2003, 2005; Nakamura et al., 2004; Morimura

[2] The last equality is derived by differentiating $\sum_{x\in\mathcal{X}} \Pr(x|\boldsymbol{\theta})\nabla_\theta \ln \Pr(x|\boldsymbol{\theta}) = \mathbf{0}$ with respect to the parameter $\boldsymbol{\theta}$

[3] It is same in the case of all f -divergences in general, except for scale (Amari and Nagaoka, 2000)

et al., 2005; Richter et al., 2007) did not seriously address the above problem and (naively) used the Riemannian metric matrix proposed by Kakade (2002). We discuss the statistical models and meric spaces for PGRL and propose a new Riemannian metric matrix.

5.2 Riemannian metric matrix for PGRL

In section 5.2.1, a novel Riemannian metric matrix for RL is proposed. In sections 5.2.2 and 5.2.3, we discuss the validity of this Riemannian metric by comparing it with the Riemannian metric proposed by Kakade (2002) and the Hessian matrix of the average reward.

5.2.1 A novel Riemannian metric matrix and NPG based on state-action probability

Since the only adjustable function in PGRL is the policy function $\pi_\theta(a|s)$, previous studies on NPG focused on the policy function $\pi_\theta(a|s)$, i.e., the statistical models $\Pr(a|s, M(\boldsymbol{\theta}))$. However, the perturbations in the policy parameter $\boldsymbol{\theta}$ cause the probability of the state $\Pr(s|M(\boldsymbol{\theta}))$ to change. Because the average reward $R(\boldsymbol{\theta})$ as the objective function of PGRL is specified by the joint probability distribution of the state and the action $(s, a) \in \mathcal{S} \times \mathcal{A}$ (eq.2.6), it is natural and adequate to focus on the statistical model $\Pr(s, a|M(\boldsymbol{\theta}))$. For this case, the FIM of $\Pr(s, a|M(\boldsymbol{\theta}))$ can be used as the Riemannian metric $\boldsymbol{G}(\boldsymbol{\theta})$. Then, its NPG consists with the direction maximizing the average reward under the constraint that a measure of changes in the KL divergence of the stationary state-action distribution with respect to $\boldsymbol{\theta}$ is fixed by a sufficient small constant ε: $D_{\mathrm{KL}}\{\Pr(s, a|M(\boldsymbol{\theta}))| \Pr(s, a|\mathrm{M}(\boldsymbol{\theta} + \mathit{\Delta}\boldsymbol{\theta}))\} = \varepsilon^2$. The FIM of this statistical model, $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, is calculated with $\Pr(s,a|M(\boldsymbol{\theta})) = d^{\pi}(s)\pi_{\theta}(a|s)$ and eq.5.3 to be

$$\begin{aligned}\boldsymbol{F}_{s,a}(\boldsymbol{\theta}) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \Pr(s,a|M(\boldsymbol{\theta}))\nabla_{\theta}\ln\Pr(s,a|M(\boldsymbol{\theta}))\nabla_{\theta}\ln\Pr(s,a|M(\boldsymbol{\theta}))^{\top} \\ &= -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\nabla_{\theta}^{2}\ln\left(d^{\pi}(s)\pi_{\theta}(a|s)\right) \\ &= \boldsymbol{F}_{s}(\boldsymbol{\theta}) + \sum_{s\in\mathcal{S}} d^{\pi}(s)\,\boldsymbol{F}_{a}(s,\boldsymbol{\theta}), \qquad (5.4)\end{aligned}$$

where

$$\boldsymbol{F}_{s}(\boldsymbol{\theta}) = \sum_{s\in\mathcal{S}} d^{\pi}(s)\nabla_{\theta}\ln d^{\pi}(s)\nabla_{\theta}\ln d^{\pi}(s)^{\top} \qquad (5.5)$$

is the FIM defined from the statistical model comprising the state distribution, $\Pr(s|M(\boldsymbol{\theta}))=d^{\pi}(s)$, and

$$\boldsymbol{F}_{a}(s,\boldsymbol{\theta}) = \sum_{a\in\mathcal{A}} \pi_{\theta}(a|s)\nabla_{\theta}\ln\pi_{\theta}(a|s)\nabla_{\theta}\ln\pi_{\theta}(a|s)^{\top} \qquad (5.6)$$

is the FIM of the policy comprising the action distribution given the state s, $\Pr(a|s,M(\boldsymbol{\theta})) = \pi_{\theta}(a|s)$. Hence, the new NPG on the FIM of the stationary state-action distribution is

$$\widetilde{\nabla}_{\boldsymbol{F}_{s,a},\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \boldsymbol{F}_{s,a}(\boldsymbol{\theta})^{-1}\,\nabla_{\theta}R(\boldsymbol{\theta}).$$

We term it the "natural stationary policy gradient"(NSG).

5.2.2 Comparison with Kakade's Riemannian metric matrix

The only Riemannian metric matrix for RL that has been proposed so far is the following matrix, which was proposed by Kakade (2002) and was the weighted sum of the FIMs of the policy by the stationary state distribution $d^{\pi}(s)$,

$$\overline{\boldsymbol{F}}_{a}(\boldsymbol{\theta}) \equiv \sum_{s\in\mathcal{S}} d^{\pi}(s)\,\boldsymbol{F}_{a}(s,\boldsymbol{\theta}). \qquad (5.7)$$

This is equal to the second term in eq.5.4. If it is assumed that the stationary state distribution is not changed by a variation in the policy, i.e., if $\nabla_{\theta}d^{\pi}(s) = \boldsymbol{0}$ holds, then $\boldsymbol{F}_{s}(\boldsymbol{\theta}) = \boldsymbol{0}$ holds according to eq.5.5. Under this assumption, Kakade's metric $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is equivalent to $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, while this assumption is not true in general. These facts indicate that $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is the Riemannian metric matrix ignoring the the change in the stationary state distribution $d^\pi(s)$ brought about by the perturbation in the policy parameter $\boldsymbol{\theta}$ in terms of the statistical model of the stationary state-action distribution $\Pr(s,a|M(\boldsymbol{\theta}))$.

Meanwhile, Bagnell and Schneider (2003) and Peters et al. (2003) independently, showed the relationship between the Kakade's metric and the system trajectories $\xi_T = (s_0, a_0, s_1, ..., a_{T-1}, s_T) \in \Xi_T$: When the FIM of the statistical model for the system trajectory ξ_T,

$$\Pr(\xi_T|M(\boldsymbol{\theta})) = \Pr(s_0)\prod_{t=0}^{T-1}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t,a_t),$$

is normalized by the time steps T with the limit $T \to \infty$, it is equivalent to the Kakade's Riemannian metric,

$$\begin{aligned}\lim_{T\to\infty}\frac{1}{T}\boldsymbol{F}_{\xi_T}(\boldsymbol{\theta}) &= -\lim_{T\to\infty}\frac{1}{T}\sum_{\xi_T\in\Xi_T}\Pr(\xi_T|M(\boldsymbol{\theta}))\nabla_\theta^2\left\{\sum_{t=0}^{T-1}\ln\pi_\theta(a_t|s_t)\right\}\\ &= -\sum_{s\in\mathcal{S}}d^\pi(s)\sum_{a\in\mathcal{A}}\pi_\theta(a|s)\nabla_\theta^2\ln\pi_\theta(a|s)\\ &= \sum_{s\in\mathcal{S}}d^\pi(s)\,\boldsymbol{F}_a(s,\boldsymbol{\theta}) = \overline{\boldsymbol{F}}_a(\boldsymbol{\theta}) \qquad (5.8)\end{aligned}$$

Since the PGRL objective, i.e., the maximization of the average reward, is reduced to the optimization of the system trajectory by eq.2.4, Bagnell and Schneider (2003); Peters et al. (2003) suggest that the Kakade's metric $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ could be a good metric. However, being equal to $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$, the normalized FIM for the infinite-horizon system trajectory obviously differs with $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ and is the metric that ignores the information $\boldsymbol{F}_s(\boldsymbol{\theta})$ about the stationary state distribution $\Pr(s|M(\boldsymbol{\theta}))$. This is due to the fact that the statistical model of the system trajectory considers not only the state-action joint distribution but also the progress for the (infinite) time steps, as follows.

Here, s_{+t} and a_{+t} are the state and the action, respectively, progressed in t time steps after converging the stationary distribution. Since the distribution of the system trajectory for T time steps from the stationary distribution, $\xi_{+T} \equiv$

$(s, a_{+0}, s_{+1}, ..., a_{+T-1}, s_{+T}) \in \Xi_T$, is

$$\Pr(\xi_{+T}|M(\boldsymbol{\theta})) = d^{\pi}(s) \prod_{t=0}^{T-1} \pi_\theta(a_{+t}|s_{+t}) p(s_{+t+1}|s_{+t}, a_{+t}),$$

its FIM is calculated such that

$$\boldsymbol{F}_{\boldsymbol{\xi}_{+T}}(\boldsymbol{\theta}) = \boldsymbol{F}_s(\boldsymbol{\theta}) + T\overline{\boldsymbol{F}}_a(\boldsymbol{\theta}), \tag{5.9}$$

the derivation of which is shown in 3.1. Because of $\lim_{T\to\infty} \boldsymbol{F}_{\xi_{+T}}/T = \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$, the Kakade's metric $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is regarded as the limit $T \to \infty$ of the system trajectory distribution for T time steps from the stationary state distribution. Consequently, $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ omits the FIM of the state distribution, $\boldsymbol{F}_s(\boldsymbol{\theta})$. On the other hand, the FIM of the system trajectory distribution for one time step is obviously equivalent to the FIM of the state-action joint distribution, i.e., $\boldsymbol{F}_{\xi_{+1}}(\boldsymbol{\theta}) = \boldsymbol{F}_{s,a}(\boldsymbol{\theta})$.

Now, discuss which FIM is adequate for the average reward maximization. As declared in section 5.2.1, the average reward in eq.2.6 is the expectation of $\bar{r}(s, a)$ from the distribution of the state-action (or 1-time-step system trajectory) and does not depend on the system trajectories after +2 time steps. Therefore, it indicates that the Kakade's metric $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ supposed a redundant statistical model and the proposed metric for state-action distribution, $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, would be more natural and adequate for PGRL. We give comparisons among various metrics such as $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$, and a unit matrix $\boldsymbol{I}$ through the numerical experiments in section 5.5.

Similarly, when the reward function is temporarily a function of T time steps, $r(s_t, a_t, ..., a_{t+T}, s_{t+T+1})$, instead of one time step, $r(s_t, a_t, s_{t+1})$, the FIM of the T-time-step system trajectory distribution, $\boldsymbol{F}_{\xi_{+T}}(\boldsymbol{\theta})$, would be a natural metric because the average reward becomes $R(\boldsymbol{\theta}) = \sum_{\xi_{+T}\in\Xi_T} \Pr(\xi_{+T}|M(\boldsymbol{\theta}))r(\xi_{+T})$.

5.2.3 Analogy with Hessian matrix

We discuss the analogies between the Fisher information matrices $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ and $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ and the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta})$, which is the second derivative of the average reward with respect to the policy parameter $\boldsymbol{\theta}$,

$$\begin{aligned}
\boldsymbol{H}(\boldsymbol{\theta}) &\equiv \nabla_\theta^2 R(\boldsymbol{\theta}) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \bar{r}(s,a) \nabla_\theta^2 \big(d^\pi(s)\pi_\theta(a|s)\big) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \bar{r}(s,a) d^\pi(s)\pi_\theta(a|s) \\
&\quad \Big\{\nabla_\theta^2 \ln\big(d^\pi(s)\pi_\theta(a|s)\big) + \nabla_\theta \ln\big(d^\pi(s)\pi_\theta(a|s)\big)\nabla_\theta \ln\big(d^\pi(s)\pi_\theta(a|s)\big)^\top\Big\} \qquad (5.10) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \bar{r}(s,a) d^\pi(s)\pi_\theta(a|s) \\
&\quad \Big\{\nabla_\theta^2 \ln \pi_\theta(a|s) + \nabla_\theta^2 \ln d^\pi(s) \\
&\quad\quad + \nabla_\theta \ln \pi_\theta(a|s) \nabla_\theta \ln \pi_\theta(a|s)^\top + \nabla_\theta \ln d^\pi(s) \nabla_\theta \ln d^\pi(s)^\top \\
&\quad\quad + \nabla_\theta \ln d^\pi(s) \nabla_\theta \ln \pi_\theta(a|s)^\top + \nabla_\theta \ln \pi_\theta(a|s) \nabla_\theta \ln d^\pi(s)^\top\Big\}. \qquad (5.11)
\end{aligned}$$

Comparing eq.5.7 of the Kakade's metric matrix $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ with eq.5.11 of the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta})$, the Kakade's metric does not have any information about the last two terms in curly brackets $\{\cdot\}$ of eq.5.11, as Kakade (2002) pointed out[4]. This is because $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is derived under $\nabla_\theta d^\pi(s) = \mathbf{0}$. By eq.5.4 and eq.5.10, meanwhile, the proposed metric $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ obviously has some information about all the terms of $\boldsymbol{H}(\boldsymbol{\theta})$. Therefore, even through the comparison with the Hessian matrix, it is suggested that $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ should be an appropriate metric for PGRL. Additionally, $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ becomes equivalent to the Hessian matrix in the cases using an atypical reward function that depends on $\boldsymbol{\theta}$ (see Appendix 3.2).

It is noted that the average reward would not be a quadratic form with respect to the policy parameter $\boldsymbol{\theta}$ in general. In particularly when $\boldsymbol{\theta}$ is far from the optimal parameter $\boldsymbol{\theta}^*$, the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta})$ is prone to an indefinite matrix. Meanwhile, no FIM $\boldsymbol{F}(\boldsymbol{\theta})$ becomes an indefinite matrix and is always positive (semi-)definite, assured by its definition in eq.5.3. Accordingly, the natural gradient method using FIM might be a more versatile covariant gradient ascent for PGRL than the Newton-Raphson method (Nocedal and Wright, 2006), the gradient direction of which is the same as that of $\widetilde{\nabla}_{-\boldsymbol{H},\boldsymbol{\theta}} R(\boldsymbol{\theta})$. Comparison experiments

[4] $\boldsymbol{H}(\boldsymbol{\theta})$ is sligthtly different from the Hessian matrix used in (Kakade, 2002) in a precise sense. However, the burden of the argument is the same as in (Kakade, 2002).

are presented in section 5.5.

5.3 NPG on Fisher information matrix $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$

In this section, we view the estimation of the NSG, the NPG on the metric $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$. It will be shown that this estimation can be reduced to the regression problem of the immediate rewards.

Consider the following linear regression model

$$f^{\pi}_{\omega}(s,a) \equiv \boldsymbol{\phi}_{\theta}(s,a)^{\top}\boldsymbol{\omega}, \tag{5.12}$$

where $\boldsymbol{\omega}$ is the adjustable parameter and $\boldsymbol{\phi}_{\theta}(s,a)$ is the basis function of the state and action, also depending on the policy parameter $\boldsymbol{\theta}$,

$$\begin{aligned}\boldsymbol{\phi}_{\theta}(s,a) &\equiv \nabla_{\theta} \ln\left(d^{\pi}(s)\pi_{\theta}(a|s)\right) \\ &= \nabla_{\theta}\ln d^{\pi}(s) + \nabla_{\theta}\ln \pi_{\theta}(a|s).\end{aligned} \tag{5.13}$$

Then, the following theorem holds:

Theorem 3 *Let the Markov chain $M(\boldsymbol{\theta})$ have the fixed policy parameter $\boldsymbol{\theta}$, if the objective is to minimize the mean square error $\epsilon(\boldsymbol{\omega})$ of the linear regression model $f^{\pi}_{\omega}(s_t,a_t)$ in eq.5.12 for the rewards r_{t+1},*

$$\epsilon(\boldsymbol{\omega}) = \lim_{T\to\infty}\frac{1}{2T}\sum_{t=0}^{T-1}\left\{r_{t+1} - f^{\pi}_{\omega}(s_t,a_t)\right\}^2, \tag{5.14}$$

then the optimal adjustable parameter $\boldsymbol{\omega}^{*}$ is equal to NSG as the natural policy gradient on $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$:

$$\widetilde{\nabla}_{\boldsymbol{F}_{s,a},\boldsymbol{\theta}} R(\boldsymbol{\theta}) = \boldsymbol{\omega}^{*}.$$

Proof: By the ergodic property of $M(\boldsymbol{\theta})$, eq.5.14 is transcribed to

$$\varepsilon(\boldsymbol{\omega}) = \frac{1}{2}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_{\theta}(a|s)\left(\bar{r}(s,a) - f^{\pi}_{\omega}(s,a)\right)^2.$$

Since the optimal adjustable parameter $\boldsymbol{\omega}^*$ that minimizes the error $\varepsilon(\boldsymbol{\omega})$ satisfies $\nabla_\omega \varepsilon(\boldsymbol{\omega})|_{\boldsymbol{\omega}=\boldsymbol{\omega}^*} = \mathbf{0}$,

$$\begin{aligned}
&\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\phi}_\theta(s,a)\big(\bar{r}(s,a) - \boldsymbol{\phi}_\theta(s,a)^\top \boldsymbol{\omega}^*\big) = \mathbf{0}\\
\Leftrightarrow\ &\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\phi}_\theta(s,a)\boldsymbol{\phi}_\theta(s,a)^\top \boldsymbol{\omega}^*\\
&\qquad\qquad = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\phi}_\theta(s,a)\bar{r}(s,a)
\end{aligned}$$

holds. By the definition of the basis function (eq.5.13),

$$\begin{aligned}
&\sum_{s,a} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\phi}_\theta(s,a)\boldsymbol{\phi}_\theta(s,a)^\top = \boldsymbol{F}_{s,a}(\boldsymbol{\theta}),\\
&\sum_{s,a} d^\pi(s)\pi_\theta(a|s)\boldsymbol{\phi}_\theta(s,a)r(s,a) = \nabla_\theta R(\boldsymbol{\theta}),
\end{aligned}$$

hold. Therefore,

$$\begin{aligned}
\boldsymbol{\omega}^* &= \boldsymbol{F}_{s,a}(\boldsymbol{\theta})^{-1}\nabla_\theta R(\boldsymbol{\theta})\\
&= \widetilde{\nabla}_{R,\boldsymbol{\theta}}(\boldsymbol{\theta})
\end{aligned}$$

holds. □

It is confirmed by theorem 3 that if the least-square regression to the immediate reward r_{t+1} by the linear function approximator $f^\pi_\omega(s_t, a_t)$ with the basis function $\boldsymbol{\phi}_\theta(s,a) \equiv \nabla_\theta \ln(d^\pi(s)\pi_\theta(a|s))$ is performed, the adjustable parameter $\boldsymbol{\omega}$ becomes the unbiased estimate of NSG $\widetilde{\nabla}_{\boldsymbol{F}_{s,a},\boldsymbol{\theta}} R(\boldsymbol{\theta})$. Therefore, since the NSG estimation problem is reduced to the regression problem of the reward function, NSG would be simply estimated by the least-square technique or by such a gradient descent technique as the method with the eligibility traces proposed by Morimura et al. (2005), where the matrix inversion is not required.

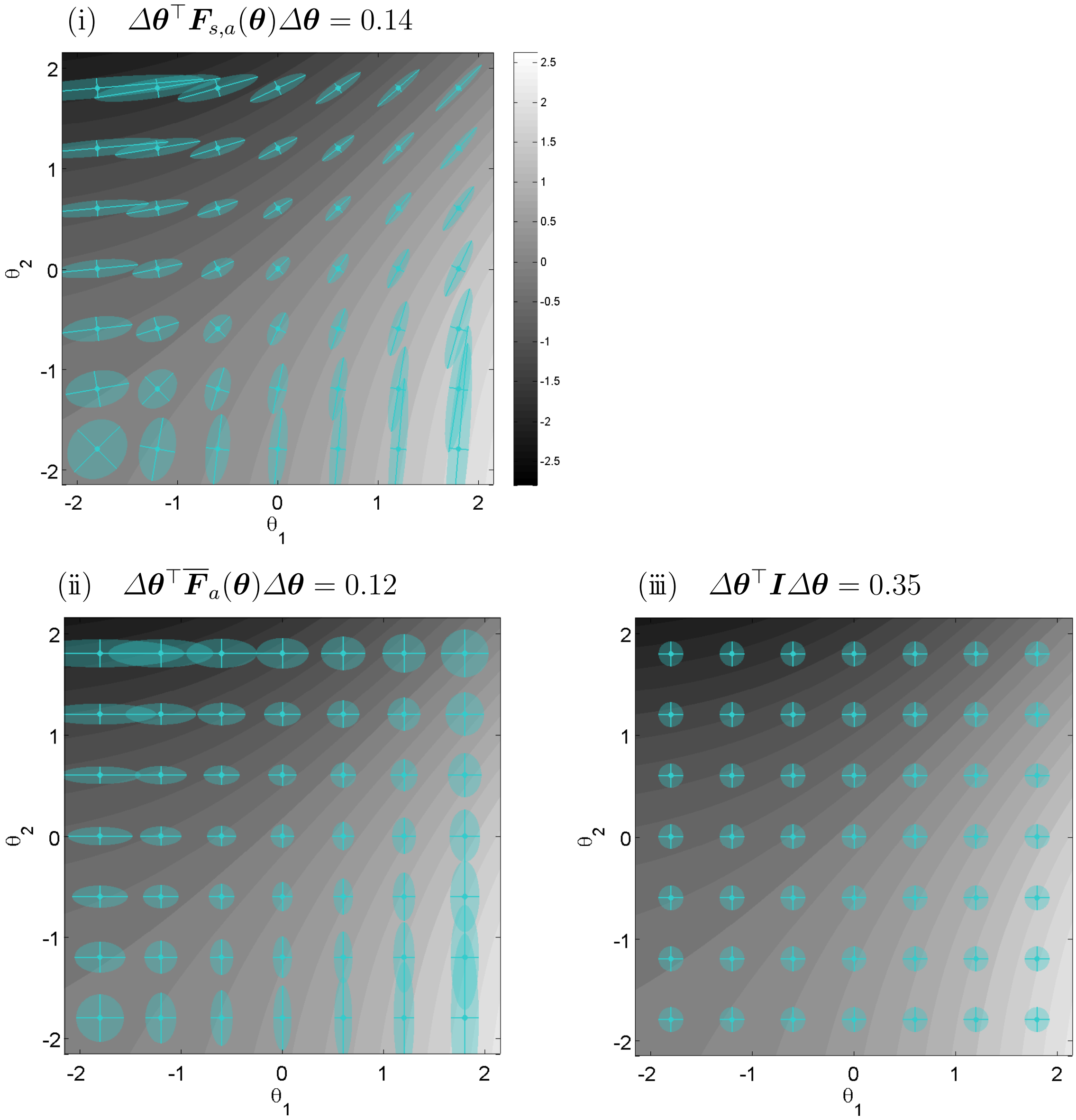
It is to be noted that, in order to implement this estimation, the computation of both the derivatives, $\nabla_\theta \ln \pi_\theta(a|s)$ and $\nabla_\theta \ln d^\pi(s)$, is required for the basis function $\boldsymbol{\phi}_\theta(s,a)$. While $\nabla_\theta \ln \pi_\theta(a|s)$ can be instantly calculated, $\nabla_\theta \ln d^\pi(s)$ cannot be solved analytically because the state transition probabilities are generally unknown in RL. However, an efficient online estimation manner for $\nabla_\theta \ln d^\pi(s)$, which is similar to the method of estimating the value function, has been established by Morimura et al. (2007b). However, we have not discussed the concrete implementations in the thesis.

5.4 Numerical experiment I: comparison of Riemannian metrics

In this section, we look into the differences among the fixed-distance spaces defined by the Riemannian metric matrices $\boldsymbol{G}(\boldsymbol{\theta})$—the proposed metric $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, Kakade's metric $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$, and unit matrix $\boldsymbol{I}$—in a simple two-state MDP (Kakade, 2002), where each state $s \in \{1, 2\}$ has self- and cross-transition actions $\mathcal{A} = \{l, m\}$ and each state transition is deterministic. The policy with $\boldsymbol{\theta} \in \mathcal{R}^2$ is represented by the sigmoidal function:

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

where $\boldsymbol{\psi}(s) \in \mathcal{R}^{|\mathcal{S}|}$ is the feature vector of the state. Here, we set $\boldsymbol{\psi}(1) = [1, 0]^\top$ and $\boldsymbol{\psi}(2) = [0, 1]^\top$. Figure 5.1 shows the phase planes of the policy parameter $\boldsymbol{\theta}$. The gray level denotes the log ratio of the stationary state distribution, and each ellipsoid corresponds to the set of $\varDelta\boldsymbol{\theta}$ satisfying a constant distance $\varDelta\boldsymbol{\theta}^\top \boldsymbol{G}(\boldsymbol{\theta}) \varDelta\boldsymbol{\theta} = \varepsilon^2$ as the fixed distance spaces by $\boldsymbol{G}(\boldsymbol{\theta})$, in which NPG looks for the steepest direction maximizing the average reward. It is confirmed that the ellipsoids by the proposed metric $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ coped with the changes in the state distribution by the perturbation in $\boldsymbol{\theta}$ because the alignment of the minor axis of the ellipsoid on $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ complied with the direction significantly changing the $d^\pi(s)$. This indicates that the policy update with NSG does not drastically change $d^\pi(s)$ and also $\nabla_\theta R(\boldsymbol{\theta})$ in eq.2.7. Thus, though it does not get out of our prospect, it might not be easy that $\nabla_\theta R(\boldsymbol{\theta})$ (and also NSG) becomes $\mathbf{0}$ by the update with NSG. If it is true, the speed of approach to plateau and also the local maximum might be slow in NSG learning. On the contrary the other metrics could not grasp the changes even though $\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ is the expectation of $\boldsymbol{F}_a(\boldsymbol{\theta})$ over $d^\pi(s)$, as we see in theoretical studies.



(i) $\mathit{\Delta}\boldsymbol{\theta}^\top \boldsymbol{F}_{s,a}(\boldsymbol{\theta})\mathit{\Delta}\boldsymbol{\theta} = 0.14$

(ii) $\mathit{\Delta}\boldsymbol{\theta}^\top \overline{\boldsymbol{F}}_{a}(\boldsymbol{\theta})\mathit{\Delta}\boldsymbol{\theta} = 0.12$

(iii) $\mathit{\Delta}\boldsymbol{\theta}^\top \boldsymbol{I}\mathit{\Delta}\boldsymbol{\theta} = 0.35$

Figure 5.1. Phase planes of a policy parameter in a two-state MDP: The gray level denotes $\ln d_{\boldsymbol{\theta}}(1)/d_{\boldsymbol{\theta}}(2)$. Each ellipsoid denotes the fixed distance spaces by each metric $G(\boldsymbol{\theta}) :=$ (i) $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$, (ii) $\overline{\boldsymbol{F}}_{a}(\boldsymbol{\theta})$, or (iii) $\boldsymbol{I}$.

5.5 Numerical experiment II: comparison of policy optimization

In this section, we compare the proposed NPG (NSG) learning with the Kakade's NPG and the other policy gradient learnings through arbitrary Markov decision problems with various number of states.

5.5.1 Experimental setting

Setting of MDP

For each $|\mathcal{S}| \in \{3, 6, 10, 15, 20, 25, 30, 40\}$ states MDP with $|\mathcal{A}| = 2$ actions is constructed by the following procedures.

The state transition matrix $p(s'|s, a)$ was set not to break the ergodicity of the Markov chain $M(\boldsymbol{\theta})$ for any policy, which is the assumption required in theoretical study, and for the connections between the states to be rough. Specifically, for each condition of $p(s'|s, a)$, each pair of the state $s \in \mathcal{S} = \{1, 2, ..., |\mathcal{S}|\}$ and the action $a \in \mathcal{A} = \{l, m\}$, unnormalized probabilities are temporarilly set

$$p(s'|i, l) := \begin{cases} q_1^{i,l} & \text{if } s' = i + 1 \\ q_2^{i,l} & \text{if } s' = i \\ 0 & \text{otherwise} \end{cases}$$

$$p(s'|i, m) := \begin{cases} q_1^{i,m} & \text{if } s' = i - 1 \\ q_2^{i,m} & \text{if } s' = i \\ 0 & \text{otherwise,} \end{cases}$$

where $s' = 0$ and $s' = |\mathcal{S}|$ ($s' = 1$ and $s' = |\mathcal{S}| + 1$) are the identical states. The set values are normalized to satisfy $\sum_{s' \in \mathcal{S}} p(s'|s, a) = 1$,

$$\begin{aligned} p(j^{i,l}|i, l) &:= p(j^{i,l}|i, l) + 1 - q_1^{i,l} - q_2^{i,l} \\ p(j^{i,m}|i, m) &:= p(j^{i,l}|i, m) + 1 - q_1^{i,m} - q_2^{i,m}. \end{aligned}$$

Here, $q_1^{s,a}$, $q_2^{s,a}$, and $j^{s,a}$ are the following random variables for each state-action pair (s, a),

$$\begin{aligned} q_1^{s,a} &= \frac{1}{1+\exp(\mathrm{U}(-10,10))}, \\ q_2^{s,a} &= \frac{1-q_{1,s,a}}{1+\exp(\mathrm{U}(-10,10))}, \\ j^{s,a} &= \mathrm{U_d}(|\mathcal{S}|), \end{aligned}$$

where $\mathrm{U}(a, b)$ and $\mathrm{U_d}(n)$ denote the uniform random number of $[a, b]$ and the discrete uniform random number from 1 to n, respectively. An example of $p(s' s, a)$ set by the above procedures is shown figure in 5.2, where each line thickness corresponds to each measure of $p(s'|s, a)$ [5].

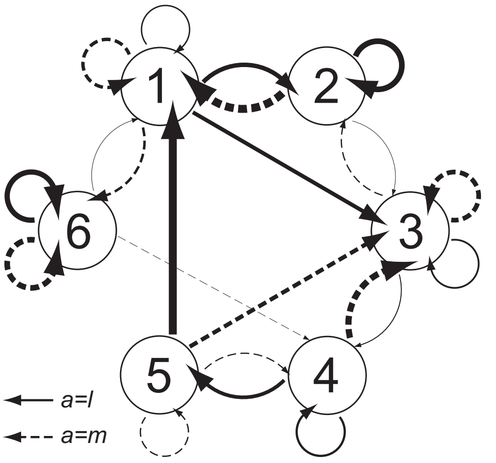


Figure 5.2. An expample of the setting of the state transition probability $p(s'|s, a)$ on MDP.

The reward function $r(s, a, s')$ was temporarilly set for each combination of arguments by standard normal distribution $\mathrm{N}(0, 1)$ and was normalized to uniform

[5] the line thickness h is defined by an inverse sigmoidal function of the state transition probability p:

$$h \propto \begin{cases} \ln(p/(1-p)) + 5 & \text{if} \quad \ln(p/(1-p)) > 5 \\ 0 & \text{otherwise.} \end{cases}$$

the maximum and the minimum of the average reward, i.e., $\max_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = 1$ and $\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = 0$:

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

The policy $\pi_\theta(a|s)$ was parameterized with the policy parameter $\boldsymbol{\theta} \in \mathcal{R}^{|\mathcal{S}|}$ by the sigmoidal function:

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

where $\boldsymbol{\psi}(i) \in \mathcal{R}^{|\mathcal{S}|}$ was the feature vector, each element of which was set by a normal distribution $\mathrm{N}(0, 1)$. Similarly, each element of the initial policy parameter vector $\boldsymbol{\theta}_0$ was set by the normal distribution $\mathrm{N}(0, 1)$.

Setting of policy gradient algorithms

Four types of gradient descents, the proposed and three policy gradient methods, are applied to the MDPs set in previous section. The only difference among these is as to the Riemannian metric matrix $\boldsymbol{G}(\boldsymbol{\theta})$ defining the direction of the gradient (eq.5.1):
(i) $\boldsymbol{G}(\boldsymbol{\theta}) := \boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ as the proposed NPG method,
(ii) $\boldsymbol{G}(\boldsymbol{\theta}) := \overline{\boldsymbol{F}}_a(\boldsymbol{\theta})$ as the Kakade's NPG method (Kakade, 2002),
(iii) $\boldsymbol{G}(\boldsymbol{\theta}) := \boldsymbol{I}$ as the ordinary policy gradient method, and
(iv) $\boldsymbol{G}(\boldsymbol{\theta}) := -\tilde{\boldsymbol{H}}(\boldsymbol{\theta})$ as a pseudo-Newton method [6].

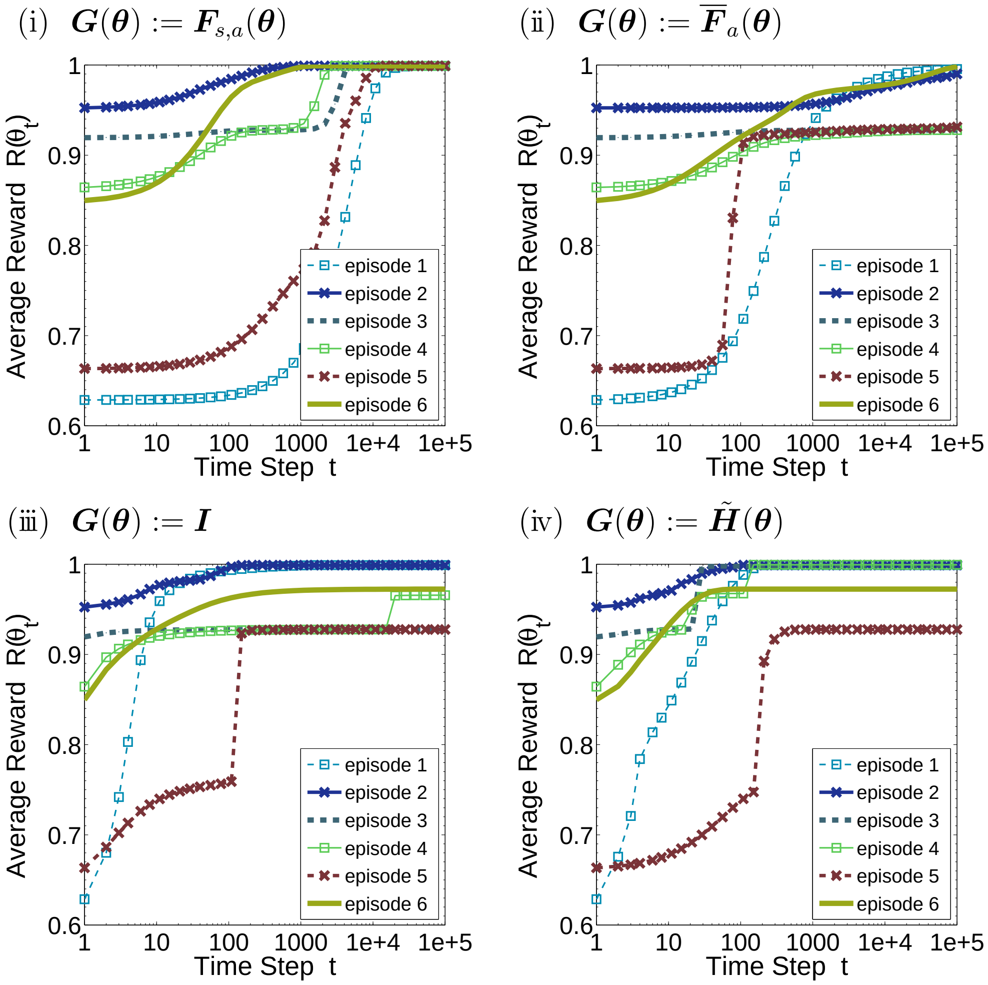
5.5.2 Results and discussions

We first introduce results by each individual episode and then show a success rate and a plateau extent of learning by all (900) episodes.

Figure 5.3 is the results about the learning curves by a total of six episodes on the MDPs of a k'th settings, $(p_k, r_k, \boldsymbol{\psi}_{k,1})$ and $(p_k, r_k, \boldsymbol{\psi}_{k,2})$, about the number of states $|\mathcal{S}| = 30$. It is noted that similar results were confirmed in the other

[6] It is noted that this pseudo-Newton method is different with so-called "the quasi Newton method".

settings than k'th. Figure 5.3 showed that the proposed natural policy gradient method could uniformly succeed at the optimization of the policy parameter, compared with the other policy gradient methods. Also, the results in figure 5.3 consistent with the results about the application of the natural gradient method for the learning of the multi-layer perceptron (Amari et al., 2000).



(i) $\boldsymbol{G}(\boldsymbol{\theta}) := \boldsymbol{F}_{s,a}(\boldsymbol{\theta})$

(ii) $\boldsymbol{G}(\boldsymbol{\theta}) := \overline{\boldsymbol{F}}_{a}(\boldsymbol{\theta})$

(iii) $\boldsymbol{G}(\boldsymbol{\theta}) := \boldsymbol{I}$

(iv) $\boldsymbol{G}(\boldsymbol{\theta}) := \tilde{\boldsymbol{H}}(\boldsymbol{\theta})$

Figure 5.3. Examples of the time courses of average reward : (i) proposed natural policy gradient, (ii) Kakade's natural policy gradient (Kakade, 2002), (iii) ordinary policy gradient, (iv) pseudo-Newton policy gradient.

The results about the success rate of the learning by 900 episodes at each number of states $|S| \in \{3, 6, 10, 15, 20, 25, 30, 40\}$ is shown in figure 5.4. Since the maximum of the average reward was set to 1, we regarded the episodes satisfying $R(\boldsymbol{\theta}_T) > 0.95$ as "success" episodes of the learning. It suggests that, in the case of the MDPs with small number of states, the proposed NPG method and Kakade's NPG method could avoid to fall sevire plateau phenomenons and learns appropriately, compared with the other methods. The reason that Kakade's method could work as well as the proposed method, is thought that the Riemannian metric used in Kakade's method has partial information about the statistical model $\Pr(s, a|M(\boldsymbol{\theta}))$. Meanwhile, Kakade's method was more losing the learning than the proposed method in the case of the MDPs with large number of states. This is thought that Kakade's Riemannian metric omits the Fisher information about the state distribution, $\boldsymbol{F}_s(\boldsymbol{\theta})$, but the proposed metric takes over $\boldsymbol{F}_s(\boldsymbol{\theta})$, as discussed theoretically in section 5.2.2.

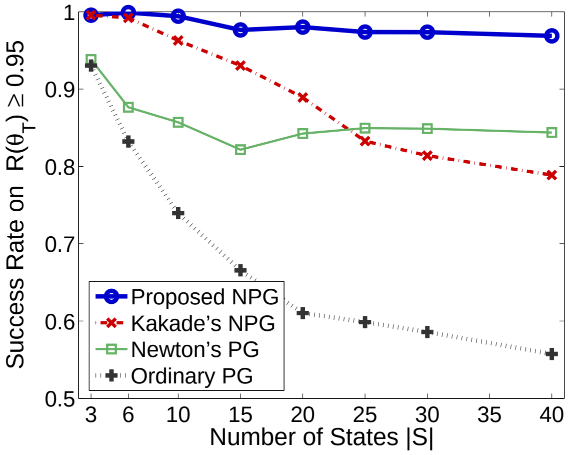


Figure 5.4. Learning success rates of the policy gradient methods

Finally, we analyzed how much of plateau each PG method, (i)∼(iv), fell to. For its criterion, we utilized a smoothness of the learning curve (discrete curvature),

$$\begin{aligned}\Delta^2 R(\boldsymbol{\theta}_t) &= \Delta R(\boldsymbol{\theta}_{t+1}) - \Delta R(\boldsymbol{\theta}_t)\\ &= R(\boldsymbol{\theta}_{t+1}) - 2R(\boldsymbol{\theta}_t) + R(\boldsymbol{\theta}_{t-1}),\end{aligned}$$

where $\Delta R(\boldsymbol{\theta}_t) \equiv R(\boldsymbol{\theta}_t) - R(\boldsymbol{\theta}_{t-1})$. The criterion for the plateau extent (PE) of a episod was defined by

$$PE = \sum_{t=1}^{T-1} \|\Delta^2 R(\boldsymbol{\theta}_t)\|.$$

Figure 5.5 represents the average of PE in 900 episodes for each PG method and shows that the proposed NPG method could learn most smoothly. This results indicates that the proposed NPG method was most avoidable from plateau phenomenons, as consisting with all other results in this chapter.

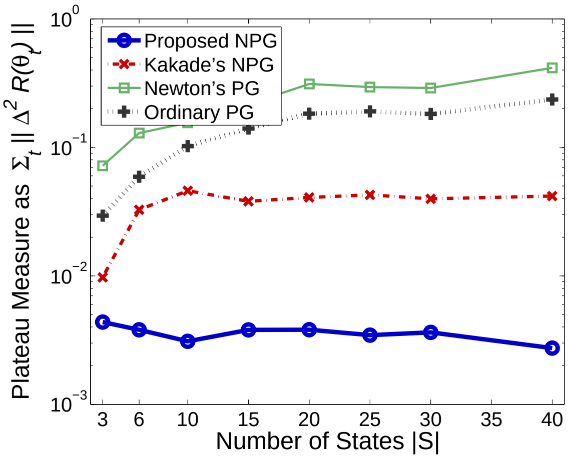


Figure 5.5. Plateau extents of the policy gradient methods.

From avobe numerical experiments, it was confirmed that the proposed NPG method could avoid from falling the plateau and learn appropriately without serious effect of the setting MDP $(p, r, \boldsymbol{\psi})$ and the initial policy parameter, especially, even if the number of states is large. Consequently, it is thought that the proposed NPG method is more natural NPG method than the NPG method proposed by (Kakade, 2002).

5.6 Summary and discussion

In this chapter, we proposed a new Riemannian metric matrix on the state-action joint distribution for the natural gradient of the average reward with respect to the policy parameter. We elucidated that the natural gradient method that has been proposed by Kakade (2002) and widely used as the natural policy gradient in RL, omited the changes in the state probability distribution brought about by the perturbation in the policy parameter, which was took into account by the proposed natural gradient method. This difference was confirmed in numerical experiments, where the proposed method worked better than the other policy gradients and rarely fell into the plateau. Additionally, it was proven that, if the immediate rewards were appropriated by using the linear function with the basis function defined by the policy, the adjustable parameter of the linear function became the unbiased estimate of the proposed natural policy gradient (NSG).

Chapter 6

Conclusion

In this thesis, we studied and developed the efficient task-independent reinforcement based on policy gradient and natural gradient.

In chapter 3, we presents the NTD algorithm, in which the regression weights of the TD error with the basis functions defined by the policy parameterization represents the natural policy gradient. If the eligibility decay rate of the NPG estimator is equal to one, the NPG estimate is updated by using the gradient of the actual observed rewards and not those of the estimated state value function; hence, the estimate is unbiased under a fixed policy. The experimental results showed that the NTD algorithm could represent the natural policy gradient and could avoid plateaus, which is consistent with the results of Amari (1998). This is extremely useful because plateaus often occur in RL problems when a suboptimal policy is more easily obtained than an optimal policy, as presented in the pendulum swing-up problem, The experimental results also demonstrated that the NTD algorithm suppresses computational costs than the existing NPG method (Peters et al., 2003) and the eligibility trace for the NPG estimator works efficiently.

In chapter 4, we showed that the actual forward and backward Markov chains are closely related and have common properties in the propositions. Utilizing these, we proposed $\mathcal{LS}$LSD(λ) as the estimation algorithm of the log stationary distribution derivative (LSD), and $\mathcal{LS}$LSD(λ)-PG as the PG algorithm utilizing the LSD estimate. The experimental results also demonstrated that $\mathcal{LS}$LSD(λ) could work at $\lambda \in [0, 1)$ and $\mathcal{LS}$LSD(λ)-PG could learn independent of the temporal discounted rate γ.

In chapter 5, we proposed a new Riemannian metric matrix on the state-action joint distribution for the natural gradient of the average reward with respect to the policy parameter. We elucidated that the natural gradient method that has been proposed by Kakade (2002) and widely used as the natural policy gradient in RL, omited the changes of the state probability distribution by the perturbation of the policy parameter, which was took into account by the proposed natural gradient method. This difference was confirmed in numerical experiments, where the proposed method worked better than the other policy gradients and rarely fell into the plateau. Additionally, it was proven that, if the immediate rewards were appropriated by the linear function with the basis function defined by the policy, the adjustable parameter of the linear function became the unbiased estimate of the proposed natural policy gradient.

Appendix

1 For chapter 2

1.1 Derivation of Eq.2.9

Using the relation between the discounted state value function and the average reward, $R(\boldsymbol{\theta}) = (1-\gamma)\sum_{s\in\mathcal{S}} d^{\pi}(s)V(s)$ (Singh et al., 1994), PG is calculated as

$$\nabla_{\theta} R(\boldsymbol{\theta}) = (1-\gamma)\left(\sum_{s\in\mathcal{S}} \nabla_{\theta} d(s)V(s) + \sum_{s\in\mathcal{S}} d(s)\nabla_{\theta} V(s)\right), \tag{1}$$

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

$$\begin{aligned}
&\sum_{s\in\mathcal{S}} d(s)\nabla_{\theta} V(s) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\nabla_{\theta}\{\pi_{\theta}(a|s)Q(s,a)\}\\
&\quad = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\Big[\nabla_{\theta}\pi_{\theta}(a|s)Q(s,a) + \pi_{\theta}(a|s)\nabla_{\theta}Q(s,a)\Big]\\
&\quad = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\Big[\nabla_{\theta}\pi_{\theta}(a|s)Q(s,a) + \pi_{\theta}(a|s)\nabla_{\theta}\sum_{s'} p(s'|s,a)\big\{\,\langle r(s',s,a)\rangle + \gamma V(s')\big\}\Big]\\
&\quad = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\nabla_{\theta}\pi_{\theta}(a|s)Q(s,a) + \gamma\sum_{s} d(s)\nabla_{\theta} V^{\pi}(s) \qquad (2)\\
&\quad = \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\nabla_{\theta}\pi_{\theta}(a|s)Q(s,a). \qquad (3)
\end{aligned}$$

Eq.2 is given by the property of stationary distribution, $d(s') = \sum_{s,a} d(s)\pi_{\theta}(a|s)$ $p(s'|s,a)$. Substituting Eq.3 in Eq.1,

$$\begin{aligned}
\nabla R(\boldsymbol{\theta}) &= (1-\gamma)\sum_{s\in\mathcal{S}} \nabla_{\theta} d(s)V(s) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\nabla_{\theta}\pi_{\theta}(a|s)Q(s,a)\\
&= (1-\gamma)\sum_{s\in\mathcal{S}} \nabla_{\theta} d(s)V(s) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\nabla_{\theta}\pi_{\theta}(a|s)\{Q(s,a) - b(s)\}, \qquad (4)
\end{aligned}$$

where the property $\sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|s) = 0$ is utilized. □

1.2 Proof of lemma 1

For simplicity, we use ∇ as ∇_θ and R as $R(\boldsymbol{\theta})$, and define the numerator of Eq.2.11 as $\ell \equiv (1-\gamma)\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \nabla d(s)\pi_\theta(a|s)Q(s,a)$. Then, we rewrite Eq.2.11 as

$$\varepsilon = \frac{||\boldsymbol{\ell}||}{||\nabla^\gamma R||}. \tag{5}$$

It is noted that ε represents the ratio between the norms of the first and second terms in Eq.4. Therefore, it implies the dominancy of the first term, which is ignored in the biased policy gradient. The cosine of the angle ϕ between the policy gradient ∇R and the biased $\nabla^\gamma R$ is bounded below by

$$\begin{aligned}\cos\phi &= \frac{\nabla R^\top \nabla^\gamma R}{||\nabla R||\,||\nabla^\gamma R||}\\ &= \frac{(\boldsymbol{\ell}+\nabla^\gamma R)^\top \nabla^\gamma R}{||(\boldsymbol{\ell}+\nabla^\gamma R)||\,||\nabla^\gamma R||}\\ &\geq \frac{||\nabla^\gamma R||^2 - ||\boldsymbol{\ell}||\,||\nabla^\gamma R||}{||\boldsymbol{\ell}+\nabla^\gamma R||\,||\nabla^\gamma R||}\\ &\geq \frac{||\nabla^\gamma R|| - ||\boldsymbol{\ell}||}{||\boldsymbol{\ell}|| + ||\nabla^\gamma R||}\\ &= \frac{1 - ||\boldsymbol{\ell}||/||\nabla^\gamma R||}{1 + ||\boldsymbol{\ell}||/||\nabla^\gamma R||}.\end{aligned}$$

That is,

$$\cos\phi \geq \frac{1-\varepsilon}{1+\varepsilon}. \tag{6}$$

It indicates that ϕ lies within $(-\pi/2, \pi/2)$ for $\varepsilon < 1$.

If the discounted value functions are normalized in the limit $\gamma \to 1$, they are equal to the average reward for state-action pairs $\{s,a\} \in \{\grave{\mathcal{S}}, \grave{\mathcal{A}}\}$ satisfying $d(s) > 0$ and $\pi_\theta(a|s) > 0$,

$$R(\boldsymbol{\theta}) = \lim_{T\to\infty}\frac{1}{T}V^{\gamma\to 1}(s) = \lim_{T\to\infty}\frac{1}{T}Q^{\gamma\to 1}(s,a), \qquad \{s,a\} \in \{\grave{\mathcal{S}}, \grave{\mathcal{A}}\}.$$

Then,

$$\frac{Q^{\gamma\to 1}(s_a, a_a)}{Q^{\gamma\to 1}(s_b, a_b)} = 1, \qquad s_a, a_b \in \grave{\mathcal{S}},\ s_b, a_b \in \grave{\mathcal{A}}. \tag{7}$$

Therefore, it is apparent that in the limit $\gamma \to 1$, ε becomes zero by Eq.2.11 and then $\cos\phi = 1$ by Eq.6; hence, $\phi = 0$ holds. It indicates that the biased policy gradient direction is the same as the true policy gradient direction in the limit $\gamma \to 1$. □

2 For chapter 3

2.1 Derivation of Eq.3.1 (Kakade, 2002)

$$\begin{aligned}
&\overline{\boldsymbol{F}}_a(\boldsymbol{\theta})(\boldsymbol{\theta})^{-1}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\nabla_\theta \log\pi_\theta(a|s) f(s,a;\boldsymbol{w})\\
&= \left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\boldsymbol{F}_a(s,\boldsymbol{\theta})\right)^{-1}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\nabla_\theta \ln\pi_\theta(a|s)\nabla_\theta \ln\pi_\theta(a|s)^\top\boldsymbol{w}\\
&= \left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\boldsymbol{F}_a(s,\boldsymbol{\theta})\right)^{-1}\left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\boldsymbol{F}_a(s,\boldsymbol{\theta})\right)\boldsymbol{w}\\
&= \boldsymbol{w}
\end{aligned}$$

□

2.2 Derivation of Eq.3.4

We introduce a proposition for the derivation of Eq.3.4, derived based on Sutton et al. (2000). Suppose that a state $\boldsymbol{z}$ is sampled by a probability density function $p(\boldsymbol{z})$, $\boldsymbol{\psi}(\boldsymbol{z})$ is a known vector function and $v(\boldsymbol{z})$ is an unknown scalar function. The object is to express the marginalized vector $\boldsymbol{\varrho} \equiv \int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})v(\boldsymbol{z})$.

Proposition 9 *The marginalized vector $\boldsymbol{\varrho}$ is rewritten as*

$$\int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})v(\boldsymbol{z}) = \int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})^\top\boldsymbol{w}^*|_{v(\boldsymbol{z})},$$

where $\boldsymbol{w}^*|_{v(\boldsymbol{z})}$ is the weight vector that minimizes the mean square error $\epsilon(\boldsymbol{w}) \equiv \frac{1}{2}\int d\boldsymbol{z}\ p(\boldsymbol{z})\left\{\boldsymbol{\psi}(\boldsymbol{z})^\top\boldsymbol{w} - v(\boldsymbol{z})\right\}^2$.

Proof: When $\epsilon(\boldsymbol{w})$ is minimized at $\boldsymbol{w}^*|_{\upsilon(\boldsymbol{z})}$, $\nabla_w \epsilon(\boldsymbol{w}^*|_{\upsilon(\boldsymbol{z})}) = 0$ holds, that is,

$$\int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z}) \left\{\boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{w}^*|_{\upsilon(\boldsymbol{z})} - \upsilon(\boldsymbol{z})\right\} = 0$$

$$\Leftrightarrow \int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})\upsilon(\boldsymbol{z}) = \int d\boldsymbol{z}\ p(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{w}^*|_{\upsilon(\boldsymbol{z})} \qquad \square$$

In the case of the policy gradient, if $\int d\boldsymbol{z}$ is replaced by $\sum_{s,a}$ and $p(\boldsymbol{z}) = p(s,a) \equiv d^\pi(s)\pi_\theta(a|s)$, $\upsilon(\boldsymbol{z}) = \upsilon(s,a) \equiv Q^\pi(s,a) - b(s)$, and $\boldsymbol{\psi}(\boldsymbol{z}) = \boldsymbol{\psi}(s,a) \equiv \nabla_\theta \log \pi_\theta(a|s)$ are substituted in Eq.9, then

$$\nabla_\theta^\gamma R(\boldsymbol{\theta}) = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^\pi(s)\pi(a|s)\nabla_\theta \log \pi_\theta(a|s) f(s,a;\boldsymbol{w}^*|_{Q^\pi(s,a)-b(s)}). \qquad (3.4)$$

2.3 Proof of proposition 1

The variation of the residual sum of squares in terms of $b(s)$ is

$$\begin{aligned}
&\frac{\delta}{\delta b(s)} \left\{ \sum_{s,a} d^\pi(s)\pi(a|s) \left(Q^\pi(s,a) - b(s) - f(s,a;\boldsymbol{w})\right)^2 \right\} \\
&= \sum_{s,a} d^\pi(s)\pi(a|s) \left(Q^\pi(s,a) - b(s) - f(s,a;\boldsymbol{w})\right) \\
&= \sum_{s,a} d^\pi(s)\pi(a|s) \left(Q^\pi(s,a) - b(s)\right),
\end{aligned}$$

using the property of the compatible function $f(s,a;\boldsymbol{w})$ (Eq.3.6) for last transformation. By the variation principle, when the residual sum of squares is minimized,

$$\sum_{s,a} d^\pi(s)\pi(a|s) \left(Q^\pi(s,a) - b(s)\right) = 0$$

holds. This is satisfied, if $b(s) = V^\pi(s)$. $\square$

In addition, when the baseline, b, is just a scalar instead of the function of a state, the residual sum of squares is minimized at

$$b = \frac{R(\boldsymbol{\theta})}{1-\gamma},$$

where $R(\boldsymbol{\theta})$ is the average reward. Similarly, it is derived by the condition

$$\frac{d}{db} \left\{ \sum_{s,a} d^\pi(s)\pi(a|s) \left(Q^\pi(s,a) - b - f(s,a;\boldsymbol{w})\right)^2 \right\} = 0.$$

2.4 Proof of proposition 2

Consider the expectation of the TD of the state value function in state-action space, $S_{sa} = \{(s,a) \in (\mathcal{S}, \mathcal{A})\}$,

$$\langle \delta(s,a) \rangle = \sum_{s'} p(s'|s,a) \left(\langle r(s,s',a) \rangle + \gamma V^{\pi}(s') \right) - V^{\pi}(s).$$

Note that $\langle \delta(s) \rangle = 0$. With the Bellman equation, $Q^{\pi}(s,a)$ is expressed as

$$Q^{\pi}(s,a) = \sum_{s'} p(s'|s,a) \left(\langle r(s,s',a) \rangle + \gamma V^{\pi}(s') \right). \tag{8}$$

By subtracting $V^{\pi}(s)$ from both-sides of Eq.8, we obtain

$$A^{\pi}(s,a) \;=\; \langle \delta^{\pi}(s,a) \rangle.$$

If $\mathrm{Var}_{\pi}(\delta^{\pi}(s,a)) \equiv \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d(s) \pi_{\theta}(a|s) \left\langle (\delta^{\pi}(s,a) - \langle \delta^{\pi}(s,a) \rangle)^2 \right\rangle$ is equal to zero, $\delta^{\pi}(s,a) = \langle \delta^{\pi}(s,a) \rangle$ holds; hence, $A^{\pi}(s,a) = \delta^{\pi}(s,a)$ holds. □

2.5 Supplement of the proof of lemma 2

We show two things; the convergence of $f(s,a;\hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)})$ and Eq.3.9. V_t, f_t, and $\boldsymbol{\psi}_t$ denote $f(s_t,a_t;\hat{\boldsymbol{w}})$, $V^{\pi}(s_t)$, and $\nabla_{\theta} \ln \pi_{\theta}(a_t|s_t)$, respectively.

Convergence of $f(s,a;\hat{\boldsymbol{w}}|_{\delta^{\pi}(s,a)})$ to $f(s,a;\boldsymbol{w}^*)$

Let the regression, gradient descent-like algorithm 1 or least squares-like algorithm 2 be performed with infinite samples from Markov chain $M(\boldsymbol{\theta})$.

In the case of gradient descent, algorithm 1 at $\lambda = 0$ and $\iota = 0$, the expectation of the gradient as the update direction of $\Delta\hat{\boldsymbol{w}}$ is

$$\begin{aligned}
\left\langle \Delta\hat{\boldsymbol{w}}|_{\delta(s,a)} \right\rangle &= \lim_{k\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (\delta_t - f_t)\boldsymbol{\psi}_t \\
&= \lim_{k\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left(r_{t+1} + \gamma V_{t+1} - V_t - f_t\right) \boldsymbol{\psi}_t \\
&= \sum_{s',s,a} d(s)\pi(a|s)p(s'|\boldsymbol{x},\boldsymbol{a}) \left(\langle r(s',s,a)\rangle + \gamma V(s') - V(x) - f(s,a;\hat{\boldsymbol{w}})\right) \nabla_\theta \ln \pi_\theta(a_t|s_t) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d(s)\pi(a|s) \left(\langle \delta(s,a)\rangle - f(s,a;\hat{\boldsymbol{w}})\right) \nabla_\theta \ln \pi_\theta(a_t|s_t) \\
&= \nabla_w \left\{ \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d(s)\pi(a|s) \left(A^\pi(s,a) - f(s,a;\hat{\boldsymbol{w}})\right)^2 \right\},
\end{aligned}$$

where the last transformation is with $\langle\delta(s,a)\rangle = A^\pi(s,a)$ in proposition 2. Therefore, $f(s,a;\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)})$ converges to $f(s,a;\boldsymbol{w}^*)$ with an appropriate learning rate (Bertsekas and Tsitsiklis, 1996), because $\hat{\boldsymbol{w}}|_{A^\pi(s,a)}$ clearly converges $\boldsymbol{w}^*$ by the definition $A^\pi(s,a) \equiv Q^\pi(s,a) - V^\pi(s)$ with an appropriate method.

In the case of least squares, algorithm 2 at $\lambda = 0$ and $\iota = 0$,

$$\begin{aligned}
& \boldsymbol{A}^{-1}\boldsymbol{b} = \hat{\boldsymbol{w}}|_{\delta(s,a)} \\
\Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{\psi}_0 & \boldsymbol{\psi}_1 & \dots & \boldsymbol{\psi}_T \end{pmatrix} \begin{pmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \delta_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\psi}_0 & \boldsymbol{\psi}_1 & \dots & \boldsymbol{\psi}_T \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_T \end{pmatrix} \\
\Leftrightarrow \quad & \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}_t \delta_t = \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}_t f_t.
\end{aligned}$$

With infinite samples, by similar transformations as the gradient descent case,

$$\begin{aligned}
& \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\psi}_t \left(\delta_t - f_t\right) = 0 \\
\Leftrightarrow \quad & \nabla_w \left\{ \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d(s)\pi_\theta(a|s) \left(A^\pi(s,a) - f(s,a;\hat{\boldsymbol{w}}|_{\delta(s,a)})\right)^2 \right\} = 0.
\end{aligned}$$

Therefore, the fact that $f(s,a;\hat{\boldsymbol{w}}|_{\delta^\pi(s,a)})$ converges to $f(s,a;\boldsymbol{w}^*)$ is proved by similar way as in the gradient descent case.

Derivation of eq.3.9

$$
\begin{aligned}
\mathrm{RSS}_f(\delta(s,a)) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\pi_\theta(a|s)(\delta(s,a) - f(s,a;\boldsymbol{w}))^2 \\
&= \sum_{s',s,a} d(s)\pi_\theta(a|s)p(s'|s,a) \\
&\qquad (r(s',s,a) + \gamma V(s') - Q(s,a) + Q(s,a) - V(s) - f(s,a;\boldsymbol{w}))^2 \\
&= \sum_{s',s,a} d(s)\pi_\theta(a|s)p(s'|s,a)(r(s',s,a) + \gamma V(s') - V(s) - A(s,a))^2 \\
&\qquad + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\pi_\theta(a|s)(Q(s,a) - V(s) - f(s,a;\boldsymbol{w}))^2 \qquad (9) \\
&= \sum_{s',s,a} d(s)\pi_\theta(a|s)p(s'|s,a)(r(s',s,a) + \gamma V(s') - V(s) - \langle\delta(s,a)\rangle)^2 \\
&\qquad + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d(s)\pi_\theta(a|s)(A(s,a) - f(s,a;\boldsymbol{w}))^2 \qquad (10) \\
&= \sum_{s',s,a} d(s)\pi_\theta(a|s)\left\langle(\delta(s,a) - \langle\delta(s,a)\rangle)^2\right\rangle + \mathrm{RSS}_f(A(s,a)) \\
&= \mathrm{Var}_\pi(\delta(s,a)) + \mathrm{RSS}_f(A(s,a)),
\end{aligned}
$$

where the transformation to Eq.9 utilizes Eq.3.6 and the property of $Q(s,a)$ in Eq.8 and the transformation to Eq.10 uses Eq.3.8.

2.6 Estimation of NPG based on least squares

Algorithm 2 Estimation of NPG based on least squares

```
Given:
  • a policy $\pi_\theta(a|s)$.
  • the system trajectory by the policy, $\{s_0, a_0, r_1, ..., r_T, s_T, a_T\}$.
  • an estimated state value function $\hat{V}(s)$.
Initialize: $\gamma$, $\lambda$, $\iota$.
Set: $\boldsymbol{z} = \boldsymbol{0}$;, $\boldsymbol{A} := 0$;, $\boldsymbol{b} := 0$;.
For $t = 0 : T-1$ do
  $\boldsymbol{z} := \gamma\lambda\boldsymbol{z} + \nabla_\theta \ln \pi_\theta(a_t|s_t)$;
  $\boldsymbol{A} := \boldsymbol{A} + \boldsymbol{z}(\nabla_\theta \ln \pi_\theta(a_t|s_t) - \iota\nabla_\theta \ln \pi_\theta(a_{t+1}|s_{t+1}))^\top$;
  $\boldsymbol{b} := \boldsymbol{b} + \boldsymbol{z}(r_t + \gamma\hat{V}(s_{t+1}) - \hat{V}(s_t))$;
end
$\hat{\boldsymbol{w}} := \boldsymbol{A}^{-1}\boldsymbol{b}$;
Return: $\hat{\boldsymbol{w}}$ .
```

Proof of theorem 2 in the least squares case:
We denote $\boldsymbol{\psi}_t \equiv \nabla_\theta \log \pi a_t | s_t$ and $\hat{V}_t \equiv \hat{V}(\boldsymbol{x}_t)$ for simplicity. Then,

$$\begin{aligned}
& \boldsymbol{b} = \boldsymbol{A}\hat{\boldsymbol{w}} \\
\Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{\psi}_0 & \gamma\boldsymbol{\psi}_0 + \boldsymbol{\psi}_1 & \dots & \sum_{t=1}^{T} \gamma^{T-t}\boldsymbol{\psi}_{t-1} \end{pmatrix} \begin{pmatrix} r_1 + \gamma\hat{V}_1 - \hat{V}_0 \\ r_2 + \gamma\hat{V}_2 - \hat{V}_1 \\ \vdots \\ r_T + \gamma\hat{V}_T - \hat{V}_{T-1} \end{pmatrix} \\
& \quad = \begin{pmatrix} \boldsymbol{\psi}_0 & \gamma\boldsymbol{\psi}_0 + \boldsymbol{\psi}_1 & \dots & \sum_{t=1}^{T} \gamma^{T-t}\boldsymbol{\psi}_{t-1} \end{pmatrix} \begin{pmatrix} (\boldsymbol{\psi}_0 - \iota\boldsymbol{\psi}_1)^\top \\ (\boldsymbol{\psi}_1 - \iota\boldsymbol{\psi}_2)^\top \\ \vdots \\ (\boldsymbol{\psi}_{T-1} - \iota\boldsymbol{\psi}_T)^\top \end{pmatrix} \hat{\boldsymbol{w}} \\
\Leftrightarrow \quad & \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\psi}_t \left(\sum_{\tau=t}^{n} \gamma^{\tau-1} r_{\tau+1} + \gamma^{T-t} V_T - \hat{V}_\tau \right) \\
& \quad = \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\psi}_t \left(\boldsymbol{\psi}_t - \sum_{\tau=t}^{T-1} \gamma^{\tau-t}(\iota - \gamma)\boldsymbol{\psi}_{\tau+1} - \gamma^{T-1-t}\iota\boldsymbol{\psi}_T \right)^\top \hat{\boldsymbol{w}}. \qquad (11)
\end{aligned}$$

By similar transformations as those for Eq.3.12,

$$\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\nabla_\theta \ln \pi_\theta(a|s)\left(Q^{\pi}(s,a)-\hat{V}(s)-\langle\hat{\boldsymbol{w}}\rangle^{\top}\nabla_\theta \ln \pi_\theta(a|s)\right)=0$$

$$\Leftrightarrow \quad \nabla_w\left\{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi}(s)\pi_\theta(a|s)\left(Q^{\pi}(s,a)-\hat{V}(s)-\langle\hat{\boldsymbol{w}}\rangle^{\top}\nabla_\theta \ln \pi_\theta(a|s)\right)^2\right\}=0$$

For the above equality to be true,

$$\langle\hat{\boldsymbol{w}}\rangle=\boldsymbol{w}^{*}$$

holds. □

2.7 Finite Markov chain

We consider a finite Markov chain $M(\boldsymbol{\theta},\tau)$ in which the chain terminates with a probability $1-\tau$ at each time step and the chain is restarted from the initial state s_0 following the initial state distribution $p(s_0=s)$ [1]. The state distribution at time step $t\geq 1$ is calculated as $p(s_t=s'|s_0,\boldsymbol{\theta})=\tau\sum_{s,a}p(s'|s,a)\pi_\theta(a|s)p(s_{t-1}=s|s_0,\boldsymbol{\theta})$, and then the discounted stationary distribution of the state is $d_{dis}^{\pi,\tau}(s)=(1-\tau)\sum_{t=0}^{\infty}\tau^t p(s_t=s|s_0,\boldsymbol{\theta})$ because the average time steps of this chain is $1/(1-\tau)$. Therefore, the discounted average reward as an objective function is $R_{dis}(\boldsymbol{\theta},\tau)\equiv\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}d_{dis}^{\pi,\tau}(s)\pi_\theta(a|s)r(s,a)$ [2].

Corollary 1 *When the discount rate of the value function, γ is equal to τ of the finite Markov chain $M(\boldsymbol{\theta},\tau)$, $\hat{\boldsymbol{w}}|_{\delta(s,a)}$ is an unbiased estimate of the natural policy gradient in the finite Markov chain $M(\boldsymbol{\theta},\tau)$:*

$$\widetilde{\nabla}_\theta R_{dis}(\boldsymbol{\theta},\tau)=\left\langle\hat{\boldsymbol{w}}|_{\delta(s,a)}\right\rangle.$$

Proof: The average reward is calculated as

$$\begin{aligned}R_{dis}(\boldsymbol{\theta},\tau)&=\frac{1}{1-\tau}\left\langle\sum_{t=0}^{\infty}\tau^t r(s_t,a_t)\right\rangle\\&=\frac{1}{1-\tau}\sum_{s\in\mathcal{S}}p(s_0=s)V^{\pi,\tau}(s).\end{aligned}$$

[1] Although the initial state distribution does not depend on the policy, we will notate $p(s_0=s|s_0,\boldsymbol{\theta})$ for simplicity, instead of $p(s_0=s)$.

[2] Corollary 1 is approved in the case that the reward function is $r(s_{t+1},s,a)$, by simple extension of the proof

The gradient of the average reward about $\boldsymbol{\theta}$ is

$$\begin{aligned}
\nabla_\theta R_{dis}(\boldsymbol{\theta}, \tau) &= \frac{1}{1-\tau} \sum_{s \in \mathcal{S}} p(s_0 = s) \nabla_\theta V^{\pi,\tau}(s) \\
&= \frac{1}{1-\tau} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p(s_0 = s) \nabla_\theta \left\{ \pi_\theta(a|s) Q^{\pi,\tau}(s,a) \right\} \\
&= \frac{1}{1-\tau} \left\{ \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p(s_0 = s) \nabla_\theta \pi_\theta(a|s) Q^{\pi,\tau}(s,a) + \tau \sum_{s \in \mathcal{S}} p(s_1 = s | s_0, \boldsymbol{\theta}) \nabla_\theta V^{\pi,\tau}(s) \right\} \\
&= \frac{1}{1-\tau} \left\{ \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \sum_{t=1}^{\infty} \tau^{t-1} p(s_t = s | s_0, \boldsymbol{\theta}) \nabla_\theta \pi_\theta(a|s) Q^{\pi,\tau}(s,a) \right\} \\
&= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_{dis}^{\pi,\tau}(s) \nabla_\theta \ln \pi_\theta(a|s) Q^{\pi,\tau}(s,a)
\end{aligned}$$

These transformations are similar to those in 1.1. Therefore, when $\delta^{\pi,\tau}(s,a)$ is defined as the TD about $V^{\pi,\tau}(s)$, $\nabla_\theta R_{dis}(\boldsymbol{\theta}, \tau) = d_{dis}^{\pi;\tau}(s) \nabla_\theta \ln \pi_\theta(a|s) f(s, a; \langle \hat{\boldsymbol{w}}|_{\delta^{\pi,\tau}(s,a)} \rangle)$ holds as Eq.3.4 and lemma 2 (I). Hence, with the Fisher information matrix on $M(\boldsymbol{\theta}, \tau)$ derived as $G_{dis}^{\tau}(\boldsymbol{\theta}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_{dis}^{\pi,\tau}(s) \nabla_\theta \ln \pi_\theta(a|s) \nabla_\theta \ln \pi_\theta(a|s)^\top$ (Bagnell and Schneider, 2003; Peters et al., 2003), the natural policy gradient on $M(\boldsymbol{\theta}, \tau)$ is calculated as

$$\begin{aligned}
\widetilde{\nabla}_\theta R_{dis}(\boldsymbol{\theta}, \tau) &= G_{dis}^{\tau}(\boldsymbol{\theta})^{-1} \nabla_\theta R_{dis}(\boldsymbol{\theta}, \tau) \\
&= \left\langle \hat{\boldsymbol{w}}|_{\delta^{\pi,\tau}(s,a)} \right\rangle . \qquad \square
\end{aligned}$$

3 For chapter 5

3.1 Derivation of eq.5.9

For simplicity, we notate $\pi_{+t} \equiv \pi_\theta(a_{_{+t}}|s_{_{+t}})$, $p_{+t} \equiv p(s_{+t}|s_{+t-1}, a_{+t-1})$. Since ξ_{+T} is the system trajectory for T time steps from $d^\pi(s)$, $\boldsymbol{F}_{\boldsymbol{\xi}_{+T}}(\boldsymbol{\theta})$ is calculated to be

$$\begin{aligned}
&\boldsymbol{F}_{\boldsymbol{\xi}_{+T}}(\boldsymbol{\theta}) \\
&= -\sum_{\xi_{+T}\in\Xi_T} \Pr(\xi_{+T})\,\nabla_\theta^2\Big\{\ln d^\pi(s) + \sum_{t=0}^{T-1}\ln \pi_\theta(a_{_{+t}}|s_{_{+t}})\Big\} \\
&= -\sum_{s\in\mathcal{S}} d^\pi(s)\Big(\nabla_\theta^2 \ln d^\pi(s) + \sum_{a\in\mathcal{A}} {}_{+0}\pi_{+0}\Big(\nabla_\theta^2 \ln \pi_{+0} + \\
&\qquad \sum_{s\in\mathcal{S}} {}_{+1}p_{+1} \sum_{a\in\mathcal{A}} {}_{+1}\pi_{+1}\Big(\nabla_\theta^2 \ln \pi_{+1} + \\
&\qquad\quad \sum_{s\in\mathcal{S}} {}_{+2}p_{+2} \sum_{a\in\mathcal{A}} {}_{+2}\pi_{+2}\Big(\nabla_\theta^2 \ln \pi_{+2} + \cdots + \\
&\qquad\qquad \sum_{s\in\mathcal{S}} {}_{+T-1}p_{+T-1} \sum_{a\in\mathcal{A}} {}_{+T-1}\pi_{+T-1}\ \nabla_\theta^2 \ln \pi_{+T-1}\Big)\Big)\cdots\Big),
\end{aligned}$$

and, by using the balance equation of the stationary distribution (eq.2.2), the following holds:

$$\begin{aligned}
\boldsymbol{F}_{\boldsymbol{\xi}_{+T}}(\boldsymbol{\theta}) &= \boldsymbol{F}_s(\boldsymbol{\theta}) + \sum_{t=0}^{T-1}\Big(\sum_{s\in\mathcal{S}} {}_{+t}d^\pi(s_{+t})\boldsymbol{F}_a(\boldsymbol{\theta}|s_{+t})\Big) \\
&= \boldsymbol{F}_s(\boldsymbol{\theta}) + T\overline{\boldsymbol{F}}_a(\boldsymbol{\theta}). \qquad \square
\end{aligned}$$

3.2 Consistency of $\boldsymbol{F}_{s,a}(\boldsymbol{\theta})$ and $\boldsymbol{H}(\boldsymbol{\theta})$

If the immediate reward is dependent on $\boldsymbol{\theta}$

$$r(s,a;\boldsymbol{\theta}) = \frac{\Pr(s,a|\mathrm{M}(\boldsymbol{\theta}^*))}{\Pr(s,a|M(\boldsymbol{\theta}))}\ln \Pr(s,a|M(\boldsymbol{\theta})), \tag{12}$$

then the average reward becomes the negative cross entropy

$$R(\boldsymbol{\theta}) = \sum_{s,a}\Pr(s,a|\mathrm{M}(\boldsymbol{\theta}^*))\ln\Pr(s,a|M(\boldsymbol{\theta})).$$

Hence, $\Pr(s,a|\mathrm{M}(\boldsymbol{\theta}^*)) = \Pr(s,a|M(\boldsymbol{\theta}))$ holds, if the average reward is maximized. The Hessian matrix becomes $\boldsymbol{H}(\boldsymbol{\theta}) = \sum_{s,a} \Pr(s,a|\mathrm{M}(\boldsymbol{\theta}^*)) \nabla_\theta^2 \ln \Pr(s,a|M(\boldsymbol{\theta}))$. If the policy parameter is nearly optimal $\boldsymbol{\theta} \approx \boldsymbol{\theta}^*$, $\Pr(s,a|M(\boldsymbol{\theta})) \approx \Pr(s,a|\mathrm{M}(\boldsymbol{\theta}^*))$ holds by the assumption of the smoothness of $\pi_\theta(a|s)$ with respect to $\boldsymbol{\theta}$. Therefore, at this time, the Hessian matrix approximately equates the negative, proposed FIM:

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{\theta}) &\approx \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \Pr(s,a|\mathrm{M}(\boldsymbol{\theta})) \nabla_\theta^2 \ln \Pr(s,a|M(\boldsymbol{\theta})) \\ &= -\boldsymbol{F}_{s,a}(\boldsymbol{\theta}).\end{aligned}$$

$\boldsymbol{H}(\boldsymbol{\theta}^*) = -\boldsymbol{F}_{s,a}(\boldsymbol{\theta}^*)$ obviously holds. Therefore, when the reward function is in eq.12 and the policy parameter is close to the optimal, NSG almost consists with the Newton direction and the NSG learning attains quadratic convergence.

4 List of publication

Journal papers

1. 森村 哲郎, 内部 英治, 吉本 潤一郎, 銅谷 賢治. “自然方策勾配法：平均報酬の自然勾配に基づく方策探索”. 電子情報通信学会論文誌 Vol.J91-D,No.6,Jun. 2008.

2. T. Morimura, N. Noda, Y. Kato, T. Watanabe, T. Saitoh, T. Yamazaki, K. Takada, S. Aoki, K. Ohta, M. Ohshige, K. Sakaguchi, F. Sugawara. “Identification of Antibiotic Clarithromycin Binding Peptide Displayed by T7 Phage Particles”. Journal of Antibiotics, vol. 59, no. 10, pages 625–632, 2006.

Reviewed international conference papers

1. T. Morimura, E. Uchibe, K. Doya. “Natural Actor-Critic with Baseline Adjustment for Variance Reduction”. International Symposium on Artificial Life and Robotics, 2008.

2. T. Morimura, E. Uchibe, K. Doya. “Utilizing Natural Gradient in Temporal Difference Reinforcement Learning with Eligibility Traces”. International Symposium on Information Geometry and its Applications, pages 256–263, 2005.

Others

1. 森村 哲郎, 内部 英治, 銅谷 賢治. “制御器、制御方法および制御プログラム”. 特許, 特開 2007-65929 (申請中).

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

3. 吉田 岳彦, 伊藤 真, 吉本 潤一郎, 森村 哲郎, 銅谷 賢治, “遅延、確率的報酬下での意思決定の数理モデルの実験的研究”. 脳と心のメカニズム, 2007.

4. 森村 哲郎, 内部 英治, 銅谷 賢治, “割引報酬のもとでの強化学習における TD 誤差を利用した自然方策勾配法”. 電子情報通信学会技術研究報告, vol. 104, no. 759, pages 137–142, 2005.

5. 森村 哲郎, 松山 和裕, 林 卓治, 鮫島 和行, 銅谷 賢治. “人の行動データの系列モンテカルロ法解析による意思決定モデルの検証”. 脳と心のメカニズム, 2004.

6. 森村 哲郎, 野田 直子, 青木 仁子, 太田 慶祐, 坂口 謙吾, 菅原 二三男. “ファージディスプレイ法によるクラリスロマイシン結合タンパク質 p8 の同定”. 日本分子生物学会, 2003.

7. 野田 直子, 森村 哲郎, 青木 仁子, 太田 慶祐, 坂口 謙吾,菅原 二三男. “ファージディスプレイ法によるクラリスロマイシン結合タンパク質 HCC1 の同定”. 日本癌学会, 2003.

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007) "An Application of Reinforcement Learning to Aerobatic Helicopter Flight," in *Advances in Neural Information Processing Systems*: MIT Press.

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

Amari, S. (1998) "Natural Gradient Works Efficiently in Learning," *Neural Computation*, Vol. 10, No. 2, pp. 251–276.

Amari, S. and Nagaoka, H. (2000) *Method of Information Geometry*: Oxford University Press.

Amari, S., Park, H., and Fukumizu, K. (2000) "Adaptive method of realizing natural gradient learning for multilayer perceptrons," *Neural Computation*, Vol. 12, No. 6, pp. 1399–1409.

Bagnell, D. and Schneider, J. (2003) "Covariant Policy Search," in *Proceedings of the International Joint Conference on Artificial Intelligence.*

Bagnell, D., Kakade, S., Ng, A., and Schneider, J. (2004) "Policy Search by Dynamic Programming," in *Advances of Neural Information Processing Systems.*

Baird, L. C. (1993) "Advantage Updateing," technical report, Wright-Patterson Air Force Base.

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

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

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

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

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

Boyan, J. A. (1999) “Least-Squares Temporal Difference Learning,” in *The 16th International Conf. on Machine Learning*, pp. 49–56: Morgan Kaufmann, San Francisco, CA.

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

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

Brafman, R. I. and Tennenholtz, M. (2003) “R-max – A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning,” *Journal of Machine Learning Research*, Vol. 3, pp. 213–231.

Dayan, P. and Singh, S. P. (1996) “Improving Policies without Measuring Merits,” in *Advances in Neural Information Processing Systems*, Vol. 8.

Dearden, R., Friedman, N., and Andre, D. (1999) “Model based Bayesian exploration,” in *Conference on Uncertainty in Artificial Intelligence*, pp. 150–159.

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

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

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

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

Kakade, S. (2002) “A Natural Policy Gradient,” in *Advances in Neural Information Processing Systems*, Vol. 14: MIT Press.

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

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

Mori, T., Nakamura, Y., and Ishii, S. (2005) “Off-Policy Natural Actor-Critic,”Technical report, Nara Institute of Science and Technology.

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

Morimura, T., Uchibe, E., and Doya, K. (2007a) “Utilizing Temporal Difference for Natural Policy Gradient.” (submitted).

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

Nakamura, Y., Mori, T., and Ishii, S. (2004) "Natural policy gradient reinforcement learning for a CPG control of a biped robot," in *International conference on parallel problem solving from nature*, pp. 972–981.

Ng, A. Y., Harada, D., and Russell, S. (1999) "Policy invariance under reward transformations: theory and application to reward shaping," in *The 16th International Conference on Machine Learning*, pp. 278–287: Morgan Kaufmann, San Francisco, CA.

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

Nocedal, J. and Wright, S. J. (2006) *Numerical Optimization*: Springer.

Osogami, T. and Kato, S. (2007) "Optimizing system configurations quickly by guessing at the performance," in *The ACM International Conference on Measurement and Modeling of Computer Systems*, pp. 145–156.

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

Peters, J. (2005) "Machine Learning of Motor Skills for Robotics," Ph.D. dissertation, University of Southern California.

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

Peters, J., Vijayakumar, S., and Schaal, S. (2003) "Reinforcement learning for humanoid robotics," in *IEEE-RAS International Conference on Humanoid Robots.*

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

Poupart, P., Vlassis, N., Hoey, J., and Regan, K. (2006) "An Analytic Solution to Discrete Bayesian Reinforcement Learning," in *International Conference on Machine learning.*

Richter, S., Aberdeen, D., and Yu, J. (2007) "Natural Actor-Critic for Road Traffic Optimisation," in *Advances in Neural Information Processing Systems*: MIT Press.

Ronsenstein, M. T. and Barto, A. G. (2004) "Supervised actor-critic reinforcement learning," in *Learning and Approximate Dynamic Programming: Scaling Up to the Real World*: John Wiley & Sons, Inc., pp. 359–380.

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

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

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

Stone, P. and Veloso, M. (1999) "Team-partitioned, opaque-transition reinforcement learning," in *International Conference on Autonomous Agents*, pp. 206–212.

Strehl, A. and Littman, M. (2005) "A Theoretical Analysis of Model-Based Interval Estimation," in *International Conference on Machine Learning*, pp. 857–864.

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

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

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

Tesauro, G. (1995) “Temporal Difference Learning and TD-Gammon,” *Communications of the ACM*, Vol. 38, No. 5, pp. 58–68.

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

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

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

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

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