Doctoral Dissertation

Safe Natural Policy Gradient Algorithms

Ryo Iwaki

January 2019

Graduate School of Engineering
Osaka University

Supervisor: Dr. Minoru Asada

Title: Safe Natural Policy Gradient Algorithms

Committee: Dr. Minoru Asada, Chair

Dr. Koh Hosoda

Dr. Koichi Osuka

Abstract

Reinforcement learning (RL) is one of machine learning techniques where an agent tries to find an optimal decision making strategy, *policy*, in an unknown environment. Many successful applications of RL have been demonstrated such as the tax collecting and the game of Go. *Natural policy gradient* (NPG) methods are subset of RL methods, where the agent aims to acquire the locally optimal policy parameter using the natural gradients. Many recent advances in RL researches are based on the framework of NPG. Though for general machine learning methods the estimation of the natural gradient requires to compute the inverse of Fisher information matrix, it was shown more than a decade ago that the compatible function approximation technique enables the NPG for RL to be estimated with $\mathcal{O}(d)$ computational complexity and memory per update, where d is the dimensionality of the parameter vector to be optimized. However, the incremental estimation of the NPG is computationally unstable owing to its high sensitivity to the meta-parameter values, especially to the step-size used to update the estimate of NPG.

In this dissertation we address the problem of how to *safely* estimate NPGs. In the first study, we derive an adaptive step-size strategy for the incremental NPG estimation. We first derive an upper bound for the step-size for a single update of the parameter, and propose an adaptive step-size to implement the derived upper bound. The proposed adaptive step-size guarantees that an updated parameter does not overshoot the target signal, which is achieved by weighting the learning samples according to their relative importances.

In the second study, we propose a new incremental and stable algorithm for the NPG estimation. We call the proposed algorithm the *implicit incremental natural actor critic* (I2NAC), and it is based on the idea of the implicit update of the parameter vector. The asymptotic convergence analysis for I2NAC is provided. Theoretical analysis results indicate the stability of I2NAC and the instability of conventional incremental NPG methods.

The usefulness of the proposed methods were confirmed by numerical experiments on the classical benchmark problems, and the results show that the proposed

methods are less sensitive to the values of the meta-parameters, including the step-size for the NPG update, compared to the existing incremental NPG method. It was suggested that even in the meta-parameter settings where the conventional incremental NPG method resulted in the policy degradation or the divergence of the estimated parameter, I2NAC could still improve the policy.

Recently, the concept of compatible function approximation was generalized to the broad class of gradient-like learning rules. The proposed methods in this dissertation can be widely applicable to the approximation of the natural gradient with $\mathcal{O}(d)$ computational complexity and memory per update, not only for the policy-gradient-based RL methods but also for the general gradient-based machine learning methods and are worth further studies in the future.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction: Safety in Reinforcement Learning

In order to make a intelligent machine that can behave adaptively in unknown environments, the system should be designed so that it is capable of making the most of experiences and improving its own decision making strategy. If some evaluation signals to its behaviors are available, the decision maker should acquire the appropriate association between the stimuli and the responses. However, the problems of how to assign an immediate evaluation to each past decision making and how to reinforce the associations between the inputs and the outputs are non-trivial. *Reinforcement learning* is a strong candidate to answer this open problem.

1.1 Reinforcement Learning

The problem we are interested in is a sequential decision making in an unknown *environment* (see Figure 1.1). The environment is a subset of the universe that is relevant to the problem we try to solve. The decision maker is called an *agent*. The agent observes the *state* of the environment, and makes a decision about which *action* to choose. The action causes the environment to change the state according to its *dynamics*. At the same time the agent receives a scalar evaluative signal called a *reward* according to a predefined *reward function*. The reward is the immediate evaluation to the decision that was just made. The decision making rule of the agent is called a *policy*. The agent, who initially does not know what are the "good behaviors" in the environment, tries to change the policy so that the amount of the reward it will

receive in the long run future is maximized. As a criterion to determine whether the policy is "good" or "bad", it is often convenient to use *value functions*, which are defined as the expected value of the *return*, the cumulative future rewards. The agent should learn the value functions and regulate the policy so that the value functions are maximized.

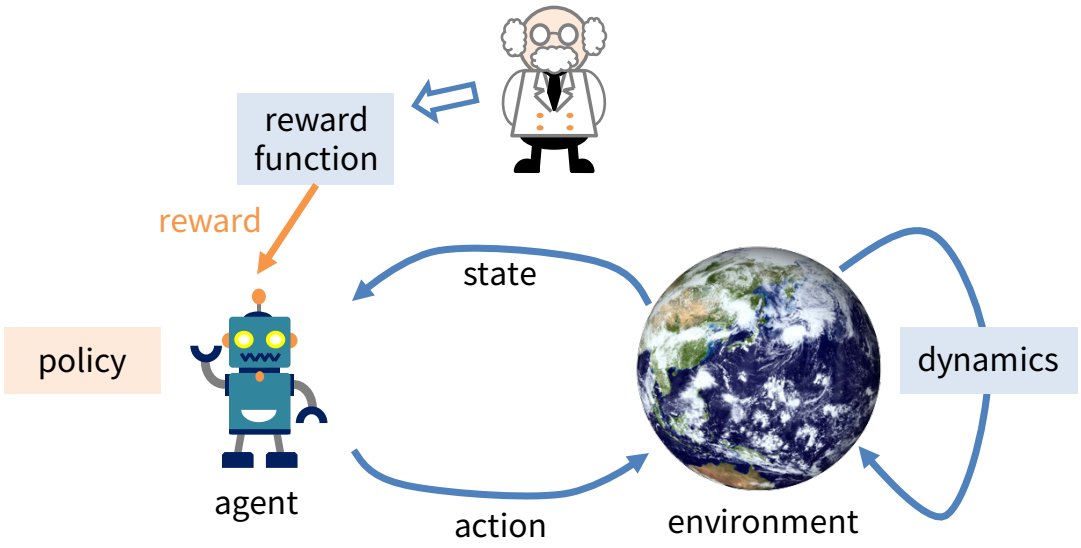


FIGURE 1.1: An overview of reinforcement learning.

If the complete model of the environment is available, that is, if the agent can access to the dynamics and the reward function directly, an optimal policy can be obtained by *dynamic programming* methods such as *policy iteration* or *value iteration* (Howard, 1960; Bertsekas and Tsitsiklis, 1996). However, in practice, it is rare that the dynamics and the reward function are available, thus somehow the agent has to learn good policy without knowing the model of environment. This setting is called *model-free*. Therefore it is required for the agent to collect the samples, the tuples consist of state, action, successor state and reward, through the trial-and-error interaction with the environment. This optimization problem is called *reinforcement learning* (RL) (Bertsekas and Tsitsiklis, 1996; Kaelbling, Littman, and Moore, 1996; Sutton and Barto, 1998; Szepesvári, 2010; Sugiyama, 2015; Sutton and Barto, 2017).

Ideally, the users of RL methods need not consider "how to achieve the goal"; the agent learns it autonomously and we just need to design a reward function in which "what the agent should achieve" is encoded. This philosophy was clearly stated by Sutton and Barto, 1998; Sutton and Barto, 2017:

> [...] the reward signal is not the place to impart to the agent prior knowledge about how to achieve what we want it to do [...]. The reward signal is your way of communicating to the robot **what** you want it to achieve, **not how** you want it achieved.

This is the big advantage of RL because it is often intractable to design how to solve the problem in all the possible circumstances. In the literature, there are many successful applications of RL methods; for example, playing the games such as backgammon (Tesauro, 1995), Atari 2600 (Mnih et al., 2015), and Go (Silver et al., 2016; Silver et al., 2017), soccer ball shooting (Asada et al., 1996), helicopter flight (Ng et al., 2003; Abbeel et al., 2007), dexterous in-hand manipulation (OpenAI, 2018), tax collecting (Abe et al., 2010), theorem proving (Kaliszyk et al., 2018) and devise placement (Gao, Chen, and Li, 2018).

There are many taxonomies in RL methods. One of the important taxonomies is *value-based* versus *policy-based*. In value-based methods, the *action value function*, an expected value of the return conditioned on state-action pair, is learned. The decision making follows the policy derived from the estimated action value function. The classical and common strategies are *greedy*, *ϵ-greedy* and *softmax*. The famous examples of value-based methods are *Q-learning* (Watkins, 1989; Watkins and Dayan, 1992) and *SARSA* (Rummery and Niranjan, 1994).

On the other hand, in policy-based methods, the agent has a *parameterized policy* which is represented independently of value functions. The purpose of the agent is to optimize the policy parameter so that the expected return is maximized. A value function may be used in the learning process, but is not required for the decision making. Policy-based methods are also referred to as *policy search* methods. One of the advantages of the policy-based methods is that since the policy is represented by a function approximator, it is easy to deal with the continuous action space. There are two main methods to optimize the policy parameter; the derivative free optimization and *policy gradient*. The former includes evolutionary algorithm (Moriarty, Schultz, and Grefenstette, 1999) and cross entropy method (Mannor, Rubinstein, and Gat, 2003). In the latter approach, the policy parameter is updated by the gradient ascent, where the policy gradient is a steepest direction which maximizes the expected

return (Gullapalli, 1990; Kimura and Kobayashi, 1998; Sutton et al., 1999; Baxter and Bartlett, 2001). The focus of this dissertation is on the *natural policy gradient* methods, which form a subset of policy gradient methods.

1.1.1 Natural Policy Gradient

The natural policy gradient (NPG) was originally proposed by Kakade, 2001, and is an application of the *natural gradient* (Amari, 1998) to the policy gradient. The "gradient" is the steepest direction to maximize the given learning objective function and derived by solving a constrained optimization problem. Amari, 1998 proposed to use the "natural" metric as the constraint and the derived steepest direction is called the natural gradient. The natural gradient is a general technique and has been widely applied in many fields with appropriate metrics, for example statistical estimation of probability density, optimization of neural network, and blind source deconvolution (see (Amari, 1998) for the detail). In the literature of RL, the Fisher information matrix (FIM) of the policy weighted by a state distribution is used as the metric of policy parameter space. The important property of the NPG was shown by Bagnell and Schneider, 2003, that it is *covariant*, that is, roughly speaking, the behavior of learning rule is invariant even if the parameterization of the policy changes. The usefulness of natural gradients in the RL domain was confirmed more than a decade ago in high-dimensional control tasks of humanoid robots (Peters, Vijayakumar, and Schaal, 2003; Peters and Schaal, 2008a). Furthermore, recent advances in policy-gradient-based *deep reinforcement learning* methods are significantly based on the NPG (Schulman et al., 2015; Duan et al., 2016; Wang et al., 2017; Chou, Maturana, and Scherer, 2017; Wu et al., 2017; Schulman et al., 2017; OpenAI, 2018).

In general, the optimization of the policy parameter using NPG requires the following steps:

1. Estimation of the "vanilla" gradient of the policy parameter with respect to given objective function.

2. Estimation of the FIM of the policy parameter.

3. Calculation of the inverse of the estimated FIM.

4. Computation of the product of the inverse FIM and the estimated gradient. This product gives the estimate of NPG.

5. The update of the policy parameter using the estimated NPG.

In order to implement the above procedures, one needs $\mathcal{O}(d^2)$ memory complexity and $\mathcal{O}(d^3)$ computational complexity per update, where d is the dimensionality of the parameter vector to be optimized. Even by the sophisticated methods (Amari, Park, and Fukumizu, 2000; Park, Amari, and Fukumizu, 2000), $\mathcal{O}(d^2)$ computational complexity are required. Recently it was proposed to estimate the inverse of FIM using Kronecker-factorization with very cheap memory and computation (Martens and Grosse, 2015), it requires some assumptions such as the FIM is block-diagonal. However, in RL, it was shown by Kakade, 2001 that using the *compatible* function approximator (Sutton et al., 1999), the NPG can be estimated directly as a vector without even storing any matrix.

1.2 Safety in Reinforcement Learning Algorithms

Given the brilliant successes, now a question arises: **is the reinforcement learning matured enough as a practical technique?** The researchers will answer **"No"**, because there are many potential dangers in RL. A considerable research has been done to ensure the *safety* of RL methods.

The word *safe* has been used as several meanings in the literature of RL. In this section, we briefly review the researches directed to the safety of RL. The safeties in RL methods can be categorized into four classes:

1. Safety in Exploration,

2. Safety via Optimization Criterion,

3. Safety via Constrained Optimization,

4. Safety in Optimization Procedure.

For the first three categories, a superb survey was provided by García and Fernández, 2015. In the following we review only a small subset of the literature and some

recent advances. The survey here is mainly focused on the model-free methods. The safety that we focus on in this dissertation belongs to the last category.

1.2.1 Safety in Exploration

As stated in the above, it is usually assumed that the agent collects the learning samples through the trial-and-error interaction with the unknown environment. In order to learn the good policy, the agent should explore the state and action spaces sufficiently. The most naive exploration strategy is ϵ-greedy in which the agent chooses a random action with probability ϵ while the agent exploits the current knowledge by choosing the action whose value is the largest with probability $1 - \epsilon$. Such a exploration strategy is dangerous because it is possible to result in a catastrophic state, for example, self-driving car causes an accident. For the purpose of *safe exploration*, many methods have been proposed:

- **Incorporating Prior Knowledge**: Many methods aim at the safe exploration requiring prior knowledge of the target task. Maire, 2005 proposed a method to derive initial value functions from a finite set of demonstrations by an expert. The high quality initial value function leads to the safe initial behavior. Abbeel and Ng, 2005 proposed to estimate the dynamics of the environment from demonstrations, derive the policy using the estimated dynamics and collect new samples by exploiting the derived policy. Importantly, in their approach there is no need to use the exploring policy explicitly. On the other hand, there are studies in which the existence of a "teacher", who provides advice to the agent when it is necessarily. The teacher advise can be an action (Clouse and Utgoff, 1992), a set of actions from which the agent has to choose, or an reward signal (Thomaz and Breazeal, 2006). As another line of study, Perkins and Barto, 2001; Perkins and Barto, 2002 proposed to apply Lyapunov design methods so that the state of environment is brought and kept in a predefined subset of state space. Perkins and Barto, 2001 proposed to restrict the agent to choose the action by which the designed Lyapunov function decreases. On the other hand, in (Perkins and Barto, 2002), the agent learns to

switch base controllers which are designed using Lyapunov domain knowledge and guaranteed to be safe even when taking the exploratory actions.

- **Risk-directed Exploration**: Another kind of researches does not require the prior knowledge, in which a measurement of *risk* is learned in the learning process instead. For example, Gehring and Precup, 2013 proposed to use the mean absolute value of temporal difference error, an approximation error of value function, as an indicator of how "controllable" the state is. If the temporal difference error of a particular state(-action pair) is varies considerably, the state is less controllable. The estimated controllability is used as an exploration bonus for the greedy action selection. Thus, the agent explores the controllable subset of state space and the sample collection becomes safer.

1.2.2 Safety via Optimization Criterion

In another branch of researches, the optimization criterion, or the objective function of the learning, is changed so that it is possible to deal with a notion of *risk*. In this context, the conventional definition of optimization criterion – the expected value of the return – is called a *risk neutral* (Puterman, 1994, § 4.1.1). Because this criterion is a expectation over the state and action spaces, it is not always the suitable one for the tasks where catastrophic states exist; again, for example, self-driving car causes an accident.

- **Worst Case Criterion**: Heger, 1994 proposed the *minimax criterion*, which is defined as the minimum expected value of the return. Maximizing the minimax criterion means that the total reward in the "worst case scenario" is maximized, therefore the minimax criterion is also called the *worst case criterion*. However, the worst case criterion is too restrictive in general because it takes into account very rare events that possibly never happens.

- **Risk-Sensitive Criterion**: In the *risk-sensitive* case, the objective criterion is defined by the exponential utility functions (Howard and Matheson, 1972; Mihatsch and Neuneier, 2002). Gosavi, 2009 proposed the penalization of the objective function by the variance of the return. The exponential utility also corresponds to the sum of expectation and variance of the return in the first order.

For both cases, a scalar parameter decides whether the agent is risk-aversion or risk-seeking. Though these methods are *model-based*, Borkar, 2001 proposed a model-free approach for the exponential utility objective and proved it converges to the near-optimal with probability one. Tamar, Castro, and Mannor, 2013; Tamar, Castro, and Mannor, 2016 also proposed a model-free method to estimate the variance of the return and showed its convergence with probability one. Morimura et al., 2010c; Morimura et al., 2010b proposed to estimate the density of the return instead of the expected value or variance and confirmed its usefulness in the risk-sensitive RL.

1.2.3 Safety via Constrained Optimization

The third safety is achieved by maximizing the expected return subject to some constraints. Depending on the design of the constraints, the property of the resultant RL algorithm dramatically changes.

- **Constraint on Expected Return**: In (Geibel, 2006), the expected return is constrained so that it is ensured to exceed some given threshold. This approach is useful when we know a reasonable value of threshold, for example, we already have good policy and want to improve it by additional samples. This motivation is also related to the that of the monotonic improvement methods introduced in §1.2.4.

- **Constraint on Action Selection**: Abe et al., 2010 proposed a constrained RL algorithm and applied their method to optimize the actual tax collection at New York state. Their method uses the constraint for the action selection which reflect business needs, resources, and legal constraints.

- **Constraint on Policy Parameter**: Thomas et al., 2013 proposed to use the constraint by which dangerous regions in policy space are specified. Their algorithm projects the parameter to the safe region so that the constraints are satisfied.

1.2.4 Safety in Optimization Procedure

The last category is about the safety for the optimization procedure itself, where the focus is on neither the exploration nor the learning objective. The focus is on how to acquire the good policy without failure making the most of the collected samples so that the given optimization criterion is maximized.

- **Distribution Mismatch**: The methods in this category aim at the *off-policy* learning. In general, it is called off-policy if the distribution of the learning samples does not match the distribution induced from the policy we want to evaluate or improve. On the other hand, it is called *on-policy* if these distributions are equivalent. The distribution mismatch crucially affect the learning performance of RL methods. Notice that, as stated in the above, the purpose of the general RL methods is to find a policy which maximize the expected long term future reward. Therefore, the calculation of expectation from the collected samples is the key factor of RL methods. It is obvious that when we calculate an expected value of a random valuable empirically, we have to care about the distribution from which the sample were generated. In RL setting, the action of the agent is determined by the current policy and the state is determined by a distribution induced by the current policy and the dynamics of the environment. Therefore, once we update the policy, the collected samples using the previous policy becomes off-policy; the distribution of the collected learning samples so far does not match the current distribution of the state-action pair. There are many research efforts which aim at efficient off-policy learning methods for both value-based methods (Singh et al., 2000; Precup, Sutton, and Singh, 2000; Precup, Sutton, and Dasgupta, 2001; Sutton, Szepesvári, and Maei, 2008; Maei et al., 2009; Sutton et al., 2009; Maei et al., 2010; Harutyunyan et al., 2016; Munos et al., 2016) and policy-based methods (Degris, White, and Sutton, 2012; Silver et al., 2014; Nachum et al., 2017a; Imani, Graves, and White, 2018). Furthermore, off-policy learning is an important element of deep RL methods (Mnih et al., 2015; Lillicrap et al., 2016; Gu et al., 2017b; Gu et al., 2017a; Haarnoja et al., 2017; Haarnoja et al., 2018).

Though the notion of safety has not been used very often in the off-policy learning literature, we introduce an important example: Munos et al., 2016 proposed an off-policy policy evaluation algorithm called Retrace in the paper entitled "Safe and efficient off-policy reinforcement learning":

> In that sense this algorithm is not **safe**: it does not handle the case of arbitrary "off-policyness".

Thus, an off-policy algorithm is called safe if it can handle arbitrary off-policyness.

- **Monotonic Improvement**: The purpose of the methods in this category is to guarantee that each update of the policy produces at least as good policy as the previous one. Though this is guaranteed for the policy iteration or the value iteration methods if the complete model of the environment is available and the representation of the value function is exact, it is non-trivial to guarantee the performance improvement for RL settings where the model of the environment is unknown and function approximation is used.

  Several researches (Kakade and Langford, 2002; Pirotta et al., 2013; Abbasi-Yadkori, Bartlett, and Wright, 2016) address this problem where the authors derived the lower bounds of the performance difference between two policies and proposed RL algorithms which use the derived bound and generate the non-decreasing sequence of the policies.

  On the other hand, the *safe reinforcement learning* proposed by Thomas, 2015 (see also Thomas, Theocharous, and Ghavamzadeh, 2015a; Thomas, Theocharous, and Ghavamzadeh, 2015b) uses importance sampling techniques and concentration inequalities in order to guarantee that the performance of the policy improves with probability at least bigger than $1 - \delta$, where δ is a positive confidence level. Thomas, 2015 defined the safety as follows:

  > [...] we call an RL method **safe** if it guarantees with high confidence that every change that it makes to the decision-making mechanism (policy) will be an improvement.

  Though these methods are very sophisticated and have strong theoretical guarantees, their criterion to generate a new policy is sometimes too strict and

these methods are computationally expensive compared to the conventional RL methods.

- **Learning Stability**: Though there are many research efforts which aim at the stability of RL methods, here we focus on the researches that are related to the *policy oscillation* or *policy degradation* phenomena (Bertsekas and Tsitsiklis, 1996; Gordon, 2000; Bertsekas, 2010; Bertsekas, 2011; Wagner, 2011; Wagner, 2013; Wagner, 2014). It has been observed in the literature for many learning algorithms that the performance of the policy first improves and then it is followed by the oscillation between the sub-optimal policies and sometimes the learning results in the degradation of the policy performance. These phenomena are called policy oscillation and policy degradation, respectively. Bertsekas, 2010 proposed an hypothesis called *greedy partition* that explains one possible cause of the policy degradation in value-based RL methods. Roughly speaking, the parameter of a approximated value function iteratively converges to some fixed points whose corresponding greedy policies are suboptimal, and under these greedy policies the fixed point of the approximated value function is different. Though both policy oscillation and policy degradation are severe problems especially in value-based RL methods, these phenomena were also observed in a policy gradient approach (Wagner, 2014). Of course, if the policy improves monotonically in the learning then policy oscillation and degradation does not occur, thus the researches in this category should be connected with the previous one.

Safety of Natural Policy Gradient Methods

In this dissertation, we focus on NPG methods, especially on incremental natural actor critic (INAC) algorithms (Bhatnagar et al., 2009; Degris, Pilarski, and Sutton, 2012; Morimura, Uchibe, and Doya, 2005; Thomas, 2014a), which form a subset of NPG methods. INAC methods have three advantages: (i) the required memory and computation per time step scale linearly with the dimensions of the estimated parameters, (ii) all the update procedures can be executed by a simple stochastic gradient descent, (iii) no assumption is required on the FIM such as block diagonal,

and (iv) even when the FIM degenerates, INACs estimate the NPG by implicitly calculating the pseudo inverse of the FIM (Thomas, 2014b).

However, INAC methods have a serious disadvantage in that the iteration for NPG estimation is very unstable and divergent owing to its high sensitivity to the values of the meta-parameters, especially to the *step-size*, used to update the estimate of NPG. There are many studies that have improved the stability of the iteration to update the policy (Greensmith, Bartlett, and Baxter, 2004; Matsubara, Morimura, and Morimoto, 2010; Pirotta, Restelli, and Bascetta, 2013) and the state-value function (Dabney and Barto, 2012; Tamar et al., 2014). However, to the best of our knowledge, very few studies have addressed the stability of NPG estimation.

In this dissertation, we call an RL method **safe** if it has a theoretical guarantee that it can avoid the divergence of the estimated parameter. Now we can clearly state that **the purpose of this dissertation is to propose a *safe* algorithm to estimate the natural policy gradient incrementally**.

1.3 Organization and Contributions of Dissertation

The organization and contributions of this dissertation are summarized as follows:

- Chapter 2: The purpose of this chapter is to provide the mathematical formulation of reinforcement learning methods. In order to describe *what* the reinforcement learning estimate and learn, first we begin with formalizing the Markov decision process, and review a fundamental dynamic programming method by which the optimal policy is obtained exactly. Then, we show how reinforcement learning methods approximate the exact solutions through trial-and-error interaction with the unknown environment.

- Chapter 3: The purpose of this chapter is to provide the mathematical formulation of **incremental natural actor critic** (INAC) methods. This chapter describes *how* the existing natural policy gradient methods approximate the dynamic programming method. After reviewing the natural gradient, we provide the derivation of the policy gradient and the natural policy gradient. Then, we review the existing methods which use NPGs, including INACs,

in which the NPG is estimated by a simple linear regression of the temporal difference error. Importantly, in the last part of this chapter, we provide an motivative example for the safety in natural policy gradient methods.

- Chapter 4: In this chapter, we propose the first safe algorithm to estimate the NPG incrementally; we propose an adaptive step-size strategy for INAC methods. We first derive an upper bound for the step-size to estimate the NPG. If the magnitude of step-size is held below the derived upper bound, it is guaranteed that each update of the parameter vector does not overshoot the given target signal. Then, we propose an adaptive step-size strategy to implement the derived upper bound. The adaptive step-size is derived by considering the infinite-times-update with infinitesimal step-size. The proposed adaptive step-size strategy is guaranteed that it does not exceed the derived upper bound.

- Chapter 5: In this chapter, we propose the second safe algorithm to estimate the NPG incrementally; we propose a new incremental and stable algorithm for the NPG estimation based on the idea of the implicit update of the parameter vector. We call the proposed algorithm the *implicit incremental natural actor critic* (I2NAC), The asymptotic convergence analysis for I2NAC is provided. Then, we compare the learning stability of the I2NAC and INACs by calculating the upper bound of the norm of the estimated NPGs. Theoretical analysis results shows the stability of I2NAC and that the conventional INACs have a element of danger which leads the estimated NPG to divergence even when the learning procedure successes.

- Chapter 6: This brief chapter summarizes the contributions of this dissertation and discusses possible directions for future works.

Chapter 2

Background of Reinforcement Learning

In this chapter, we provide the formal description of the sequential decision making problem that we focus on in this dissertation. More specifically, in Section 2.1, we introduce the Markov decision process and reinforcement learning problem. Then, we describe the *policy iteration* in Section 2.2, which is one of the dynamic programming method to optimize the decision making strategy. Policy iteration is model-based and exact. Finally, in Section 2.3, we introduce the reinforcement learning as a approximate dynamic programming.

2.1 Markov Decision Process

We formalize the sequential decision making problem as an infinite horizon Markov decision process (MDP) (Puterman, 1994) throughout this dissertation. Let t denote the time step, which start from $t = 0$. An MDP is specified by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \rho_0)$, where:

1. $\mathcal{S}$ is the finite set of possible states of an environment, and is called the *state space*. We write s to denote an element of $\mathcal{S}$ and s_t to denote the state that occurs at time t.

2. $\mathcal{A}$ is the finite set of possible actions the agent can choose, and is called the *action space*. We write a to denote an element of $\mathcal{A}$ and a_t to denote the action that the agent takes at time t.

For the convenience, let $\xi_{t:T}$ denote the history of states and actions starting from time t and up until T:

$$\xi_{t:T} \triangleq (s_t, a_t, s_{t+1}, a_{t+1}, \ldots, s_T, a_T).$$

ξ is also called *trajectory* or *sample path*.

3. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is called the *state transition probability*, the *state transition law* or the *dynamics*. We define $\mathcal{P}(s'|s, a)$ to be the probability that the next state is s' if action a is taken in state s:

$$\mathcal{P}\left(s'|s, a\right) \triangleq \mathbb{P}\left(s_{t+1} = s'|s_t = s, a_t = a\right),$$

for all s, a, s' and $t \geq 0$. Notice that we introduced the first Markov assumption in the definition of the transition probability: the distribution over s_{t+1} depends only on s_t and a_t, that is,

$$\mathbb{P}\left(s_{t+1} = s'|s_t = s, a_t = a, \xi_{0:t-1}\right) = \mathbb{P}\left(s_{t+1} = s'|s_t = s, a_t = a\right),$$

for all s, a, s', ξ and $t \geq 0$. Furthermore, we assume the stationary Markov chain: the transition probability is independent of the time, t, that is, for any $t \geq 0$ and $t' \geq 0$, it holds that

$$\mathbb{P}\left(s_{t+1} = s'|s_t = s, a_t = a\right) = \mathbb{P}\left(s_{t'+1} = s'|s_{t'} = s, a_{t'} = a\right),$$

for all s, a and s'.

4. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ is called the *reward function*, which determines the bounded reward $r \in [R_{\min}, R_{\max}]$. We define $\mathcal{R}(s, a)$ to be the expected value of reward r that the agent receives if action a is taken in state s:

$$\begin{aligned}\mathcal{R}(s, a) &\triangleq \mathbb{E}\left[r_t \,|s_t = s, a_t = a\right] \\ &= \sum_{r \in [R_{\min}, R_{\max}]} r\, \mathbb{P}\left(r_t = r|s_t = s, a_t = a\right),\end{aligned}$$

for all s, a and $t \geq 0$. We often write $r = \mathcal{R}(s, a)$. Here we introduced the second Markov assumption: the reward at time t, r_t, depends only on s_t and a_t, that is,

$$\mathbb{P}\left(r_t = r | s_t = s, a_t = a, \xi_{0:t-1}\right) = \mathbb{P}\left(r_t = r | s_t = s, a_t = a\right),$$

for all s, a, ξ and $t \geq 0$. Furthermore, we assume that the reward function is stationary: the distribution over r_t is independent of the time, t, that is, for any $t \geq 0$ and $t' \geq 0$, it holds that

$$\mathbb{P}\left(r_t = r | s_t = s, a_t = a\right) = \mathbb{P}\left(r_{t'} = r | s_{t'} = s, a_{t'} = a\right),$$

for all s, a and r.

The tuple of the transition probability $\mathcal{P}$ and the reward function $\mathcal{R}$ is often called the *model* of the environment.

5. $\rho_0 : \mathcal{S} \to [0, 1]$ is called the *initial state distribution* because ρ_0 is the distribution over s_0: $\rho_0(s) = \mathbb{P}\left(s_0 = s\right)$.

Our purpose is to optimize the decision making strategy so that the trajectory $\xi_{0:\infty}$ is generated in compliance with our desire. The decision making rule of the agent to be optimized is called the *policy* and is denoted by π. The policy can be either stochastic or deterministic.

- The *stochastic policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the conditional distribution that the agent takes action a given that the state of environment is s:

$$\pi(a|s) \triangleq \mathbb{P}\left(a_t = a | s_t = s\right).$$

Note that we introduced the final Markov assumption that the policy is memoryless and stationary. The policy is memoryless because the probability of the action at time t depends only on the state at time t:

$$\mathbb{P}\left(a_t = a | s_t = s, \xi_{0:t-1}\right) = \mathbb{P}\left(a_t = a | s_t = s\right),$$

for all s, a, ξ and $t \geq 0$. The policy is stationary because the action probability is independent of the time, t, that is, for any $t \geq 0$ and $t' \geq 0$, it holds that

$$\mathbb{P}\left(a_t = a | s_t = s\right) = \mathbb{P}\left(a_{t'} = a | s_{t'} = s\right),$$

for all s and a.

- On the other hand, the *deterministic policy* $\pi : \mathcal{S} \to \mathcal{A}$ is a function whose input is a state and output is an action: $a_t \triangleq \pi(s_t)$. Note that the deterministic policy we consider is also memoryless and stationary.

The purpose of the reinforcement learning methods is to find the optimal policy which maximizes the long term future reward. The "long term future reward" is defined by using the cumulative of all the reward that agent will receive in the future. The optimal policy enables the agent to maximize the expected cumulative future reward irrespectively of the initial state. It is convenient to use *value functions* for the definition of the expected cumulative future reward. In this dissertation, we consider the two formulations of the value functions and the objective function as follows.

2.1.1 Discounted Reward Formulation

First, we consider the *discounted reward* formulation. The cumulative future reward, called a *return*, is defined by

$$R_t = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}, \tag{2.1}$$

where $\gamma \in (0, 1)$ is a *discount factor*. The discount factor is defined so that the return is finite. Note that $\gamma = 1$ is allowed only for *episodic* MDPs, where the interaction between the agent and the environment terminates in finite steps. Since our notion of the problem is infinite-horizon MDP in this dissertation, we do not treat $\gamma = 1$. In the following, we write $\mathop{\mathbb{E}}_{\pi,\mathcal{P}}[\cdot]$ to show that the trajectory ξ is drawn from the tuple $\pi, \mathcal{P}$. The *state value function* for the policy π, $V^\pi(s)$, is defined by

$$V^\pi(s) = \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[R_t | s_t = s\right]. \tag{2.2}$$

Similarly, the *state-action value function* or simply *action value function* for the policy π, $Q^\pi(s,a)$, is defined by

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[R_t | s_t = s, a_t = a\right]. \tag{2.3}$$

The state value and action value represent the expected future reward obtained by following the policy π starting from a state and a state-action pair, respectively. The following relation holds directly from the definitions of $V^\pi(s)$ and $Q^\pi(s,a)$:

$$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q^\pi(s,a). \tag{2.4}$$

Though value functions are the functions of state or state-action pair, the main purpose to consider these quantities in the learning perspective is to evaluate the policy. Notice that the return R_t satisfies the following recursion:

$$R_t = r_t + \gamma R_{t+1}, \tag{2.5}$$

and thus

$$\sum_{\tau=0}^{\infty} \gamma^\tau \mathcal{R}(s_{t+\tau}, a_{t+\tau}) = \mathcal{R}(s_t, a_t) + \gamma \sum_{\tau=0}^{\infty} \gamma^\tau \mathcal{R}(s_{t+1+\tau}, a_{t+1+\tau}). \tag{2.6}$$

Using the recursion above, we acquire the one step temporal consistencies of the value functions. Namely, we have

$$
\begin{aligned}
&V^{\pi}(s) \\
&= \mathbb{E}_{\pi,\mathcal{P}}\left[R_t \,|s_t = s\right] \\
&= \mathbb{E}_{a_t\sim\pi,\xi_{t+1:\infty}\sim(\pi,\mathcal{P})}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}\mathcal{R}(s_{t+\tau},a_{t+\tau})\,|s_t = s\right] \\
&= \mathbb{E}_{a_t\sim\pi,\xi_{t+1:\infty}\sim(\pi,\mathcal{P})}\left[\mathcal{R}(s_t,a_t) + \gamma\sum_{\tau=0}^{\infty}\gamma^{\tau}\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_t = s\right] \\
&= \mathbb{E}_{a_t\sim\pi}\left[\mathcal{R}(s_t,a_t)\,|s_t = s\right] + \gamma\,\mathbb{E}_{a_t\sim\pi,\xi_{t+1:\infty}\sim(\pi,\mathcal{P})}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_t = s\right] \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\mathcal{R}(s,a) + \gamma\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\times \\
&\quad\ \mathbb{E}_{a_{t+1}\sim\pi,\xi_{t+2:\infty}\sim(\pi,\mathcal{P})}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_{t+1} = s'\right] \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\mathcal{R}(s,a) + \gamma\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\,\mathbb{E}_{\pi,\mathcal{P}}\left[R_{t+1}\,|s_{t+1} = s'\right] \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\mathcal{R}(s,a) + \gamma\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^{\pi}(s') \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^{\pi}(s')\right).
\end{aligned}
$$

The similar argument holds for the action value function $Q^{\pi}(s,a)$. The *Bellman equations*, stated in the following proposition, form the basis of RL theory.

Proposition 2.1 (Bellman Equations for Discounted Reward Formulation)**.** *Value functions satisfy the following one step temporal consistencies:*

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}}\pi(a|s)\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^{\pi}(s')\right), \tag{2.7}$$

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\sum_{a'\in\mathcal{A}}\pi(a'|s')Q^{\pi}(s',a'). \tag{2.8}$$

The Bellman equations represent *backups*, that is, how value functions can be expressed recursively by using the values of successive state or state-action pair. This important property is depicted in the Figure 2.1, called *backup diagrams*. Another important relation between the state value function $V^{\pi}(s)$ and the action value function

$Q^\pi(s,a)$ is the following:

$$Q^\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^\pi(s'). \tag{2.9}$$

Eq. (2.9) is easily confirmed by by combining (2.8) and (2.4). Notice that both Eqs. (2.4) and (2.9) can be easily derived from the backup diagrams 2.1.

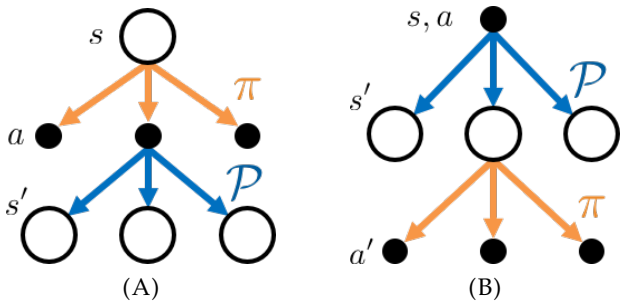


FIGURE 2.1: Backup Diagrams for (A) $V^\pi(s)$ and (B) $Q^\pi(s,a)$. Each white circle represents a state and each black circle represents a state-action pair. Orange lines are possible action selections according to the policy of agent and blue lines are possible state transitions according to the dynamics of environment.

Given value functions we can evaluate the policy, because value functions define a partial ordering over policies. Namely, for two policies π and π', we write $\pi \geq \pi'$ if and only if $V^\pi(s) \geq V^{\pi'}(s)$ for all $s \in \mathcal{S}$. For any MDP, there exists at least one *optimal policy*, π^*, that is better than or equal to all other policies, $\pi^* \geq \pi$ for all π. Though there may exist more than one optimal policies, they share the same *optimal state value function*

$$V^*(s) = \max_\pi V^\pi(s) = V^{\pi^*}(s) \tag{2.10}$$

and *optimal action value function*

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) = Q^{\pi^*}(s,a). \tag{2.11}$$

Optimal value functions represent the expected future reward obtained by following an optimal policy starting from a state and a state-action pair, respectively. Recall that under the optimal policy, the agent choose an action by which the value is maximized. In other words, the probability mass of π^* must be concentrated on the set

of actions that maximize $Q^*(s,a)$. Therefore, from (2.4) the optimal value functions $V^*(s)$ and $Q^*(s,a)$ satisfy

$$V^*(s) = \sum_{a\in\mathcal{A}} \pi^*(a|s) = \max_{a\in\mathcal{A}} Q^*(s,a) \tag{2.12}$$

Thus, the one step temporal consistency of the optimal state value function is:

$$\begin{aligned}
V^*(s) &= \max_{a\in\mathcal{A}} Q^{\pi^*}(s,a) \\
&= \max_{a\in\mathcal{A}} \mathbb{E}_{\pi^*,\mathcal{P}}\left[R_t \,|s_t = s, a_t = a\right] \\
&= \max_{a\in\mathcal{A}} \mathbb{E}_{\xi_{t+1:\infty}\sim(\pi^*,\mathcal{P})}\left[\mathcal{R}(s_t,a_t) + \gamma\sum_{\tau=0}^{\infty}\gamma^\tau\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_t = s, a_t = a\right] \\
&= \max_{a\in\mathcal{A}}\left(\mathcal{R}(s,a) + \gamma\,\mathbb{E}_{\xi_{t+1:\infty}\sim(\pi^*,\mathcal{P})}\left[\sum_{\tau=0}^{\infty}\gamma^\tau\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_t = s, a_t = a\right]\right) \\
&= \max_{a\in\mathcal{A}}\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\times\right. \\
&\qquad\left.\mathbb{E}_{a_{t+1}\sim\pi^*,\xi_{t+2:\infty}\sim(\pi^*,\mathcal{P})}\left[\sum_{\tau=0}^{\infty}\gamma^\tau\mathcal{R}(s_{t+1+\tau},a_{t+1+\tau})\,|s_{t+1} = s'\right]\right) \\
&= \max_{a\in\mathcal{A}}\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\mathbb{E}_{\pi^*,\mathcal{P}}\left[R_{t+1}\,|s_{t+1} = s'\right]\right) \\
&= \max_{a\in\mathcal{A}}\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^*(s')\right).
\end{aligned}$$

The similar argument holds for the action value function $Q^*(s,a)$. The following proposition states the *Bellman optimality equations*.

Proposition 2.2 (Bellman Optimality Equations for Discounted Reward Formulation)**.** *Optimal value functions satisfy the following one step temporal consistencies:*

$$V^*(s) = \max_{a\in\mathcal{A}}\left(\mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^*(s')\right), \tag{2.13}$$

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\max_{a'\in\mathcal{A}} Q^*(s',a'). \tag{2.14}$$

Combining (2.12) and (2.14), we have

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V^*(s'). \tag{2.15}$$

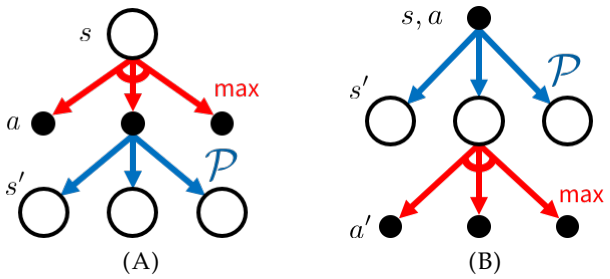


FIGURE 2.2: Backup Diagrams for (A) $V^*(s)$ and (B) $Q^*(s,a)$. Each white circle represents a state and each black circle represents a state-action pair. Red lines and arcs represent that the action is chosen so that the values are maximized and blue lines are possible state transitions according to the dynamics of environment.

As wee see later, the state value function V^π and the action value function Q^π are the unique fixed points of the Bellman equations (2.7) and (2.8), respectively. Similarly, the optimal state value function V^* and the optimal action value function Q^* are the unique fixed points of the Bellman optimality equations (2.13) and (2.14), respectively.

Further, we define $\rho^\pi(s)$ as the (unnormalized) γ-discounted future state distribution for the initial state distribution ρ_0 under the policy π:

$$\rho^\pi(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\rho_0, \pi). \tag{2.16}$$

In order to express $\rho^\pi(s)$ in terms of $\rho_0, \mathcal{P}$ and π, let us consider the n-step state transition probability $\mathcal{P}^{\pi,n}$ from state s_t to s_{t+n} under the Markov chain according to the policy π:

$$\begin{aligned}
&\mathcal{P}^{\pi,n}(s_{t+n}|s_t) \\
&= \sum_{a_t \in \mathcal{A}} \pi(a_t|s_t) \sum_{s_{t+1} \in \mathcal{S}} \mathcal{P}(s_{t+1}|s_t, a_t) \sum_{a_{t+1} \in \mathcal{A}} \pi(a_{t+1}|s_{t+1}) \cdots \\
&\quad \times \sum_{s_{t+n-1} \in \mathcal{S}} \mathcal{P}(s_{t+n-1}|s_{t+n-2}, a_{t+n-2}) \sum_{a_{t+n-1} \in \mathcal{A}} \pi(a_{t+n-1}|s_{t+n-1}) \mathcal{P}(s_{t+n}|s_{t+n-1}, a_{t+n-1}).
\end{aligned} \tag{2.17}$$

Note that

$$\mathbb{P}(s_n = s|s_0 = s', \pi) = \mathcal{P}^{\pi,n}(s|s') \tag{2.18}$$

and $\mathbb{P}(s_0 = s|s_0 = s, \pi) = 1$. Then, $\rho^\pi(s)$ can be decomposed into

$$\begin{aligned}\rho^\pi(s) &= \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\rho_0, \pi) \\ &= \sum_{s_0 \in \mathcal{S}} \rho_0(s') \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|s_0 = s', \pi) \\ &= \sum_{s_0 \in \mathcal{S}} \rho_0(s') \sum_{t=0}^{\infty} \gamma^t \mathcal{P}^{\pi,t}(s|s').\end{aligned} \tag{2.19}$$

The objective function of the learning is defined as the discounted future reward:

$$\eta(\pi) \triangleq \sum_{s \in \mathcal{S}} \rho_0(s) V^\pi(s) = \sum_{s \in \mathcal{S}} \rho^\pi(s) \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}(s, a). \tag{2.20}$$

2.1.2 Average Reward Formulation

Next, we consider the *average reward* formulation. In the average reward formulation, we require the following assumption:

Assumption 2.3. *Under any policy π, the Markov chain resulting from the given MDP is irreducible and aperiodic (ergodic).*

Roughly speaking, if the Markov chain under the policy π is ergodic, then any state is eventually reachable from any other state by following π. Under Assumption 2.3, there exists a limiting state distribution $\rho^\pi(s)$ independent of the initial state:

$$\rho^\pi(s) = \lim_{t \to \infty} \mathbb{P}\left(s_t = s|s_0 = s', \pi\right), \quad \forall s' \in \mathcal{S}. \tag{2.21}$$

The limiting distribution $\rho^\pi(s)$ is the unique stationary distribution and satisfies the balance equation:

$$\rho^\pi(s') = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mathcal{P}(s'|s, a) \pi(a|s) \rho^\pi(s). \tag{2.22}$$

The objective function of the learning is defined as the average reward:

$$\eta(\pi) \triangleq \lim_{T\to\infty} \frac{1}{T} \mathop{\mathbb{E}}_{\pi,\mathcal{P}} \left[\sum_{t=0}^{T-1} r_t \right] = \sum_{s\in\mathcal{S}} \rho^{\pi}(s) \sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{R}(s,a). \tag{2.23}$$

For each policy π, the state value function $V^{\pi}(s)$ and the state-action value function $Q^{\pi}(s,a)$ are given by

$$V^{\pi}(s) = \sum_{\tau=0}^{\infty} \mathop{\mathbb{E}}_{\pi,\mathcal{P}} \left[r_{t+\tau} - \eta(\pi) | s_t = s \right], \tag{2.24}$$

$$Q^{\pi}(s,a) = \sum_{\tau=0}^{\infty} \mathop{\mathbb{E}}_{\pi,\mathcal{P}} \left[r_{t+\tau} - \eta(\pi) | s_t = s, a_t = a \right], \tag{2.25}$$

respectively. Analogous to the discounted reward case, the state value function (2.24) and the action value function (2.25) satisfy the Bellman equations.

Proposition 2.4 (Bellman Equations for Average Reward Formulation)**.** *Value functions satisfy the following one step temporal consistencies:*

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}(s,a) - \eta(\pi) + \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V^{\pi}(s') \right), \tag{2.26}$$

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) - \eta(\pi) + \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a'\in\mathcal{A}} \pi(a'|s') Q^{\pi}(s',a'). \tag{2.27}$$

Another relations between the state value function $V^{\pi}(s)$ and the action value function $Q^{\pi}(s,a)$ are the following equations:

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q^{\pi}(s,a), \tag{2.28}$$

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) - \eta(\pi) + \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V^{\pi}(s'). \tag{2.29}$$

Furthermore, the optimal state value function $V^{*}(s)$ and the action value function $Q^{*}(s,a)$ satisfy the following Bellman optimality equations.

Proposition 2.5 (Bellman Optimality Equations for Average Reward Formulation)**.** *Optimal value functions satisfy the following one step temporal consistencies:*

$$V^*(s) = \max_{a \in \mathcal{A}} \left(\mathcal{R}(s,a) - \max_{\pi} \eta(\pi) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^*(s') \right), \tag{2.30}$$

$$Q^*(s,a) = \mathcal{R}(s,a) - \max_{\pi} \eta(\pi) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a'). \tag{2.31}$$

The optimal value functions $V^*(s)$ and $Q^*(s,a)$ satisfy

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a), \tag{2.32}$$

$$Q^*(s,a) = \mathcal{R}(s,a) - \max_{\pi} \eta(\pi) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^*(s'). \tag{2.33}$$

2.2 Dynamic Programming

Now we can formally state that the purpose of reinforcement learning methods is to find an *optimal policy* π^* which maximizes the given objective function $\eta(\pi)$:

$$\pi^* \in \arg\max_{\pi} \eta(\pi). \tag{2.34}$$

Note that any MDPs have at least one optimal deterministic policy π^* whose corresponding value functions are optimal: $V^{\pi^*} = V^*, Q^{\pi^*} = Q^*$.

Although the focus of this dissertation is *model-free* reinforcement learning methods, let us assume here that the transition probability $\mathcal{P}$ and the reward function $\mathcal{R}$ are provided. In the following, we describe *policy iteration* (Howard, 1960), which is one of the *dynamic programming* methods to acquire the optimal policy in MDP. For the brevity, here we consider the discounted reward formulation only. First, we introduce the *Bellman operators* and show their *contraction property*. Next, we provide the *policy improvement theorem*. Then, we policy iteration is described, which consists of *policy evaluation* step and *policy improvement* step. Importantly, policy iteration finds the optimal policy by generating a sequence of *monotonically improving* policies.

2.2.1 Bellman Operators

Let $\mathcal{V} = \mathbb{R}^{\mathcal{S}}$ and $\mathcal{Q} = \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. Then $\mathcal{V}$ is a space of all the functions V such that $V : \mathcal{S} \to \mathbb{R}$ and $\mathcal{Q}$ is a space of all the functions Q such that $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We define the *Bellman operators* underlying π $\mathcal{T}^\pi : \mathcal{V} \to \mathcal{V}$ and $\mathcal{T}^\pi : \mathcal{Q} \to \mathcal{Q}$ as follows:

$$\begin{aligned}
\left(\mathcal{T}^\pi V\right)(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V(s')\right), \\
\left(\mathcal{T}^\pi Q\right)(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a'\in\mathcal{A}} \pi(a'|s') Q(s',a').
\end{aligned}$$

Similarly, we define the *Bellman optimality operators* $\mathcal{T} : \mathcal{V} \to \mathcal{V}$ and $\mathcal{T} : \mathcal{Q} \to \mathcal{Q}$ with a slight of notation:

$$\begin{aligned}
\left(\mathcal{T} V\right)(s) &= \max_{a\in\mathcal{A}} \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V(s')\right), \\
\left(\mathcal{T} Q\right)(s,a) &= \mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \max_{a'\in\mathcal{A}} Q(s',a').
\end{aligned}$$

Note that the value functions V^π and Q^π are the fixed points of the Bellman operators, respectively: $V^\pi = \mathcal{T}^\pi V^\pi, Q^\pi = \mathcal{T}^\pi Q^\pi$. Similarly, the optimal value functions V^* and Q^* are the fixed points of the Bellman operators, respectively: $V^* = \mathcal{T} V^*, Q^* = \mathcal{T} Q^*$. Let $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$ be a supremum norm for a function f whose domain is $\mathcal{X}$. As we see in the following, the Bellman (optimality) operators are *contraction mappings* in $\|\cdot\|_\infty$ norms with contraction factor γ.

Proposition 2.6 (Contraction Properties of Bellman (Optimality) Operators)**.** *Let $V_1, V_2 \in \mathcal{V}$ and $Q_1, Q_2 \in \mathcal{Q}$. Then it holds that*

$$\begin{aligned}
\left\|\mathcal{T}^\pi V_1 - \mathcal{T}^\pi V_2\right\|_\infty &\le \gamma \left\|V_1 - V_2\right\|_\infty, \\
\left\|\mathcal{T} V_1 - \mathcal{T} V_2\right\|_\infty &\le \gamma \left\|V_1 - V_2\right\|_\infty, \\
\left\|\mathcal{T}^\pi Q_1 - \mathcal{T}^\pi Q_2\right\|_\infty &\le \gamma \left\|Q_1 - Q_2\right\|_\infty, \\
\left\|\mathcal{T} Q_1 - \mathcal{T} Q_2\right\|_\infty &\le \gamma \left\|Q_1 - Q_2\right\|_\infty.
\end{aligned}$$

Proof. We provide the proof only for $\mathcal{T}^\pi$ and $\mathcal{T}$. First, it holds that

$$\begin{aligned}
\|\mathcal{T}^\pi V_1 - \mathcal{T}^\pi V_2\|_\infty &= \sup_{s\in\mathcal{S}} \Bigg| \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V_1(s') \right) \\
&\qquad - \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V_2(s') \right) \Bigg| \\
&= \gamma \sup_{s\in\mathcal{S}} \left| \sum_{s'\in\mathcal{S}} \mathcal{P}^{\pi,1}(s'|s) \left(V_1(s') - V_2(s') \right) \right| \\
&\le \gamma \sup_{s\in\mathcal{S}} \sum_{s'\in\mathcal{S}} \mathcal{P}^{\pi,1}(s'|s) \left| V_1(s') - V_2(s') \right| \\
&\le \gamma \sup_{s\in\mathcal{S}} \sum_{s'\in\mathcal{S}} \mathcal{P}^{\pi,1}(s'|s) \|V_1 - V_2\|_\infty \\
&= \gamma \|V_1 - V_2\|_\infty .
\end{aligned}$$

The last equality follows from $\sum_{s'\in\mathcal{S}} \mathcal{P}^{\pi,1}(s'|s) = 1$. Then, $\mathcal{T}^\pi$ is a contraction because $\gamma < 1$. Similarly, we have

$$\begin{aligned}
\|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty &= \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \Bigg| \mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \max_{a'\in\mathcal{A}} Q_1(s',a') \\
&\qquad - \mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \max_{a'\in\mathcal{A}} Q_2(s',a') \Bigg| \\
&= \gamma \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \left(\max_{a'\in\mathcal{A}} Q_1(s',a') - \max_{a'\in\mathcal{A}} Q_2(s',a') \right) \right| \\
&\le \gamma \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \left| \max_{a'\in\mathcal{A}} Q_1(s',a') - \max_{a'\in\mathcal{A}} Q_2(s',a') \right| \\
&\overset{(a)}{\le} \gamma \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \max_{a'\in\mathcal{A}} \left| Q_1(s',a') - Q_2(s',a') \right| \\
&\le \gamma \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \|Q_1 - Q_2\|_\infty \\
&= \gamma \|Q_1 - Q_2\|_\infty ,
\end{aligned}$$

where (a) follows from

$$\max_{x\in\mathcal{X}} f(x) - \max_{x\in\mathcal{X}} g(x) \le \max_{x\in\mathcal{X}} |f(x) - g(x)| .$$

■

Given Proposition 2.6, it follows immediately that the fixed points of Bellman (optimality) operators, V^π, Q^π, V^* and Q^*, are unique for the corresponding operators. Furthermore, by applying the operators successively, $\forall V \in \mathcal{V}$ and $\forall Q \in \mathcal{Q}$ converges to the corresponding fixed point. For example, assume that $\mathcal{T}^\pi$ has a fixed point V' other than V^π. Then, we have

$$\begin{aligned} \|V^\pi - V'\|_\infty = \|\mathcal{T}^\pi V^\pi - \mathcal{T}^\pi V'\|_\infty \leq \gamma \|V^\pi - V'\|_\infty \\ \Leftrightarrow \quad (1-\gamma)\|V^\pi - V'\|_\infty \leq 0. \end{aligned}$$

Since $\gamma < 1$ and $\|\cdot\|_\infty \geq 0$, we get

$$\|V^\pi - V'\|_\infty = 0 \quad \Leftrightarrow \quad \forall s \in \mathcal{S}, V^\pi(s) = V'(s).$$

Next, let $\forall V_0 \in \mathcal{V}$ and let $V_{k+1} = \mathcal{T}^\pi V_k$. Then, it holds that

$$\begin{aligned} \|V_k - V^\pi\|_\infty &= \|\mathcal{T}^\pi V_{k-1} - \mathcal{T}^\pi V^\pi\|_\infty \\ &\leq \gamma \|V_{k-1} - V^\pi\|_\infty \\ &\leq \gamma^2 \|V_{k-2} - V^\pi\|_\infty \\ &\leq \cdots \\ &\leq \gamma^k \|V^0 - V^\pi\|_\infty \end{aligned}$$

and thus

$$\lim_{k \to \infty} \|V_k - V^\pi\|_\infty = 0.$$

2.2.2 Policy Improvement Theorem

As wee see later, policy iteration generates a sequence of monotonically improving policies. This fact is justified by the following *policy improvement theorem*.

Proposition 2.7 (Policy Improvement Theorem)**.** *Let π and π' be two policies such that*

$$V^\pi(s) \leq \sum_{a \in \mathcal{A}} \pi'(a|s) Q^\pi(s,a) \tag{2.35}$$

for all $s \in \mathcal{S}$. Then it holds that

$$V^{\pi}(s) \leq V^{\pi'}(s) \tag{2.36}$$

for all $s \in \mathcal{S}$.

Proof. From Eqs. (2.35) and (2.9), it holds that

$$\begin{aligned}
V^{\pi}(s) &\leq \sum_{a \in \mathcal{A}} \pi'(a|s) Q^{\pi}(s,a) \\
&= \sum_{a \in \mathcal{A}} \pi'(a|s) \left(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^{\pi}(s') \right) \\
&\leq \sum_{a \in \mathcal{A}} \pi'(a|s) \left(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a \in \mathcal{A}} \pi'(a'|s') Q^{\pi}(s',a') \right) \\
&= \mathbb{E}_{a_t \sim \pi'} \left[\mathcal{R}(s_t, a_t) + \gamma V^{\pi}(s_{t+1}) | s_t = s \right].
\end{aligned}$$

By iteratively substituting the left hand side for the right hand side, we get, for any $n \in \mathbb{N}$,

$$\begin{aligned}
V^{\pi}(s) &\leq \mathbb{E}_{a_t \sim \pi', s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi'} \left[\mathcal{R}(s_t, a_t) + \gamma \mathcal{R}(s_{t+1}, a_{t+1}) + \gamma^2 V^{\pi}(s_{t+2}) | s_t = s \right] \\
&\leq \mathbb{E}_{a_t \sim \pi', \xi_{t+1:t+n-1} \sim (\pi', \mathcal{P})} \left[\sum_{\tau=0}^{n-1} \gamma^{\tau} \mathcal{R}(s_{t+\tau}, a_{t+\tau}) + \gamma^n V^{\pi}(s_{t+n}) | s_t = s \right].
\end{aligned}$$

Finally, taking the limit $n \to \infty$ yields

$$V^{\pi}(s) \leq \mathbb{E}_{a_t \sim \pi', \xi_{t+1:\infty} \sim (\pi', \mathcal{P})} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} \mathcal{R}(s_{t+\tau}, a_{t+\tau}) | s_t = s \right] = V^{\pi'}(s).$$

■

Notice that if π' is deterministic, Eq. (2.35) reduces to

$$V^{\pi}(s) \leq Q^{\pi}(s, \pi'(s)). \tag{2.37}$$

2.2.3 Policy Iteration

Policy iteration (Howard, 1960) is a method to find the optimal policy π^* for any given MDP. Policy iteration consists of two iterative procedures; *policy evaluation*

and *policy improvement* (shown in Figure 2.3). Let k be a integer (not a environmental time step t) and suppose that we have an estimate of the deterministic policy π_k. In policy evaluation step, one successively apply the Bellman operator $\mathcal{T}^{\pi_k}$ to the arbitrary initialized function V or Q until the convergence, respectively. Then, in policy improvement step, one generate the new *greedy* policy π_{k+1} from V^{π_k} or Q^{π_k} by the followings:

$$\pi_{k+1}(s) = \arg\max_{a \in \mathcal{A}} \left(\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^{\pi_k}(s') \right), \tag{2.38}$$

$$= \arg\max_{a \in \mathcal{A}} Q^{\pi_k}(s,a). \tag{2.39}$$

The greedy policy improvement above is guaranteed to generate the deterministic policy π_{k+1} at least as good as π_k. From the definition (2.39), we have

$$\begin{aligned} V^{\pi_k}(s) &= \sum_{a \in \mathcal{A}} \pi_k(a|s) Q^{\pi_k}(s,a) \\ &\leq \max_{a \in \mathcal{A}} Q^{\pi_k}(s,a) \\ &= Q^{\pi_k}(s, \arg\max_{a \in \mathcal{A}} Q^{\pi_k}(s,a)) \\ &= Q^{\pi_k}(s, \pi_{k+1}(s)). \end{aligned}$$

Thus, the greedy policy satisfies (2.37) and therefore from the policy improvement theorem (2.7), it holds that

$$V^{\pi_k}(s) \leq V^{\pi_{k+1}}(s), \quad \text{or} \quad Q^{\pi_k}(s,a) \leq Q^{\pi_{k+1}}(s,a),$$

for all s and a, and thus

$$\eta(\pi_k) \leq \eta(\pi_{k+1}).$$

Therefore, iterating policy evaluation and policy improvement generates the sequence of monotonically improving policies and the policy is optimal if it converged. It was proven that for the discounted reward objective the policy iteration are polynomial (Ye, 2011; Scherrer, 2013).

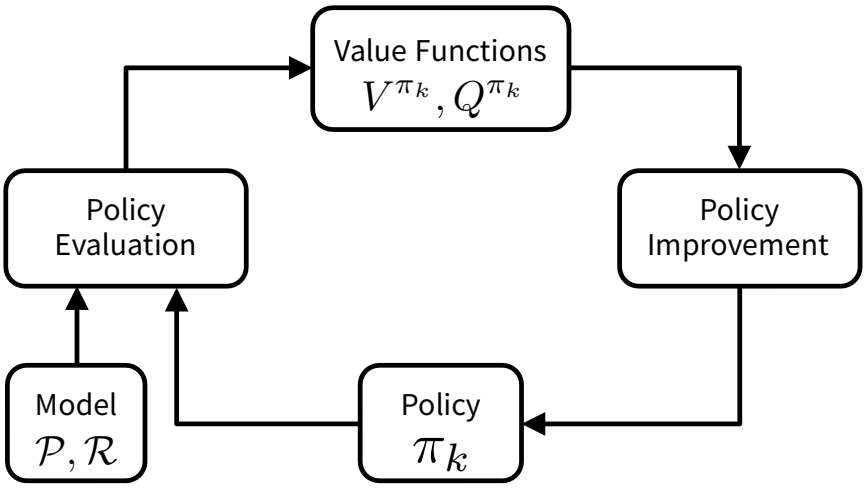


FIGURE 2.3: Framework of Policy Iteration.

2.3 Reinforcement Learning as Approximate Dynamic Programming

The convergence of policy iteration is guaranteed under the two assumptions:

1. the policy and the value functions have tabular representations,

2. the Bellman operators are applied exactly.

However, tabular representations are impractical for large state and action spaces. Furthermore, the model of environment is often unknown and we cannot even obtain the explicit form of Bellman operators in lack of the transition probability $\mathcal{P}$ and the reward function $\mathcal{R}$. Thus, we should introduce the approximations in the policy iteration framework:

1. *Function Approximations*: the policy and the value functions are represented by parametric function approximators,

2. *Model-Free Learning*: the optimization of the policy and the value functions are performed using the *samples*, (s, a, r, s'), collected through the interaction between the agent and the environment.

If function approximations are used in policy iteration framework, it is called *approximate policy iteration*. A comprehensive survey of approximate policy iteration was provided by Bertsekas, 2011. We also note that the algorithms we consider in this

dissertation is *optimistic policy iteration*, where the policy evaluation is truncated in finite number of iterations. In fact, we focus on fully incremental algorithms.

In this dissertation, we focus on the *model-free* reinforcement learning, by which we perform *approximate policy iteration* where the agent aims to optimize the *parameterized* policy and value function through the *trial-and-error* interaction with the environment. The interaction between the agent and the environment consists of the following procedures (shown in Figure 2.4): at each discrete time step t, the agent observes the current state $s_t \in \mathcal{S}$ and chooses the action $a_t \in \mathcal{A}$. The state of the environment transits to the next state s_{t+1} according to $\mathcal{P}(s_{t+1}|s_t, a_t)$, and the agent receives the reward $r_t \in [R_{\min}, R_{\max}]$ according to $\mathcal{R}(s_t, a_t)$.

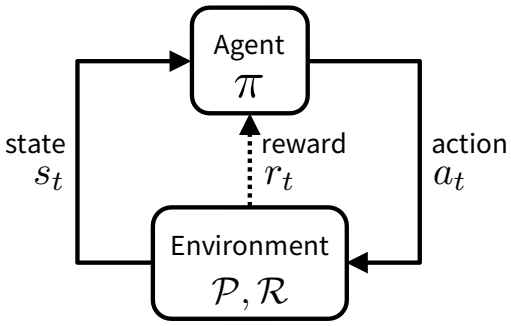


FIGURE 2.4: Diagram of how the agent and the environment interact.

Chapter 3

Natural Policy Gradient and Incremental Natural Actor Critics

In the previous chapter, we formally stated the learning objective in the reinforcement learning and the model-based method called policy iteration. In this chapter, we describe the *natural policy gradient* methods by which the policy iteration is approximately performed. The natural policy gradient methods are model-free and work well with the large function approximator. First, we describe the natural policy gradients in Section 3.1. Then, we introduce the algorithms called *incremental natural actor critics* in Section 3.2, that are the main focus in this dissertation. Importantly, in the last part of this chapter, we provide an motivative example for the safety in natural policy gradient methods.

3.1 Natural Policy Gradients

Now, we consider that the policy is *parameterized*: $\pi(a|s;\theta)$, where $\theta \in \mathbb{R}^d$ is a parameter to be optimized. We allow a shorthand notation $\pi_\theta \triangleq \pi(\cdot|\cdot;\theta)$ when it is convenient. We require the following assumption on the parameterization.

Assumption 3.1. *For any state-action pairs (s,a), $\pi(a|s;\theta)$ is differentiable with respect to θ.*

Now, the purpose of the agent is transformed into finding a (locally) optimal policy parameter θ^* which maximizes the given objective function $\eta(\theta) \triangleq \eta(\pi_\theta)$:

$$\theta^* \in \arg\max_\theta \eta(\theta).$$

Throughout this dissertation, we aim to optimize the policy parameter via *gradient methods*. In the following, we briefly review *vanilla gradients* and *natural gradients*.

3.1.1 Natural Gradient

Assume generally (not restricted to the reinforcement learning) that we aim to maximize an objective function $\eta : \mathbb{R}^d \to \mathbb{R} : \theta \mapsto \eta(\theta)$ by adjusting the parameter vector $\theta \in \mathbb{R}^d$. We search for locally optimal parameters θ^* by iteratively updating θ as follows:

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t \Delta_{\theta,t}, \tag{3.1}$$

where $\Delta_{\theta,t}$ is a *steepest direction* which maximizes $\eta(\theta_t + \Delta_{\theta,t})$ and $\{\alpha_t\}$ is a scalar *step-size* schedule.

The *vanilla gradient*, $\Delta_\theta = \nabla_\theta \eta(\theta)$, is the steepest direction that maximizes $\eta(\theta + \Delta_\theta)$ subject to $\|\Delta_\theta\|^2 < \epsilon^2$ for infinitesimal ϵ, where

$$\nabla_\theta \eta(\theta) = \left(\frac{\partial}{\partial \theta_1} \eta(\theta), \ldots, \frac{\partial}{\partial \theta_d} \eta(\theta) \right)^\top \tag{3.2}$$

and $\| \cdot \|^2$ denotes the Euclidean norm. By the *gradient ascent*,

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t \nabla_\theta \eta(\theta_t), \tag{3.3}$$

θ is guaranteed to converge to a locally optimal parameter provided that $\eta(\theta)$ is continuously differentiable and the step-size schedule satisfies the following (Bertsekas and Tsitsiklis, 2000):

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty. \tag{3.4}$$

Amari, 1998 proposed to use the 'natural' metric for what we aim to optimize. Specifically, the constraint for Δ_θ in parameter spaces is defined by a quadratic form as follows:

$$\Delta_\theta^\top G(\theta) \, \Delta_\theta < \epsilon^2, \tag{3.5}$$

where $d \times d$ positive definite matrix $G(\theta)$ is called the Riemannian metric tensor. Under the constraint (3.5), the steepest direction Δ_θ that maximizes $\eta(\theta + \Delta_\theta)$ is:

$$\Delta_\theta = \tilde{\nabla}_\theta \eta(\theta) \triangleq G^{-1}(\theta) \nabla_\theta \eta(\theta). \tag{3.6}$$

The right hand side of Eq. (3.6) is called the *natural gradient*. If $G(\theta) = I$, natural gradient reduces to the vanilla gradient.

An Example of Natural Gradient

The Riemannian metric $G(\theta)$ should be chosen appropriately to what we optimize. For example, in this section, we consider statistical estimation. Let $x \in \mathcal{X}$ be a random variable generated from a probability distribution $q(x)$, where $\mathcal{X}$ is a set of possible values x can take. We assume a statistical model $p(x|\theta)$ and the problem is to obtain the maximum likelihood estimator θ^* by which the unknown distribution $q(x)$ is approximated in the best way. In this case, the objective $\eta(\theta)$ is defined as follows:

$$\eta(\theta) = \mathop{\mathbb{E}}_{x\sim q}[\log p(x|\theta)] = \sum_{x\in\mathcal{X}} q(z) \log p(x|\theta). \tag{3.7}$$

For the parameter space of a statistical model, the Riemannian structure is defined by the *Fisher information matrix*:

$$\begin{aligned} G(\theta) &= F_x(\theta) \\ &= \mathop{\mathbb{E}}_{x\sim p}\left[\nabla_\theta \log p(x|\theta) \nabla_\theta \log p(x|\theta)^\top\right] \qquad (3.8) \\ &= \sum_{x\in\mathcal{X}} p(x|\theta) \nabla_\theta \log p(x|\theta) \nabla_\theta \log p(x|\theta)^\top \\ &= -\sum_{x\in\mathcal{X}} p(x|\theta) \nabla^2_\theta \log p(x|\theta), \qquad (3.9) \end{aligned}$$

where $\nabla^2_\theta \log p(x|\theta)$ is a Hessian matrix

$$\nabla^2_\theta \log p(x|\theta) = \begin{pmatrix} \partial^2_{\theta_{11}} \log p(x|\theta) & \cdots & \partial^2_{\theta_{1d}} \log p(x|\theta) \\ \vdots & \ddots & \vdots \\ \partial^2_{\theta_{d1}} \log p(x|\theta) & \cdots & \partial^2_{\theta_{dd}} \log p(x|\theta) \end{pmatrix}$$

and we used a shorthand notation

$$\partial^2_{\theta_{ij}} = \frac{\partial^2}{\partial\theta_i \partial\theta_j}.$$

The equality (3.9) follows from the fact that the expected value of a score function is zero. More specifically, we have

$$\sum_{x\in\mathcal{X}} p(x|\theta)\nabla_\theta \log p(x|\theta) = \sum_{x\in\mathcal{X}} \nabla_\theta p(x|\theta) = \nabla_\theta \sum_{x\in\mathcal{X}} p(x|\theta) = \nabla_\theta 1 = 0, \tag{3.10}$$

thus, differentiating above with respect to θ yields

$$\sum_{x\in\mathcal{X}} p(x|\theta)\left(\nabla_\theta \log p(x|\theta)\nabla_\theta \log p(x|\theta)^\top + \nabla^2_\theta \log p(x|\theta)\right) = 0.$$

The constraint (3.5) corresponds to regulating the Kullback-Leibler divergence between the distributions $p(x|\theta)$ and $p(x|\theta+\Delta_\theta)$:

$$\begin{aligned} D_{KL}\left(\theta\|\theta+\Delta_\theta\right) &\triangleq D_{KL}\left(p(\cdot|\theta)\|p(\cdot|\theta+\Delta_\theta)\right) \\ &= \mathop{\mathbb{E}}_{x\sim p}\left[\log \frac{p(x|\theta)}{p(x|\theta+\Delta_\theta)}\right] \\ &= \sum_{x\in\mathcal{X}} p(x|\theta)\log\frac{p(x|\theta)}{p(x|\theta+\Delta_\theta)}. \end{aligned} \tag{3.11}$$

In fact, the 2nd order Taylor expansion of (3.11) yields the Fisher information matrix:

$$\begin{aligned} &D_{KL}\left(\theta\|\theta+\Delta_\theta\right) \\ &\simeq \sum_{x\in\mathcal{X}} p(x|\theta)\left(\log\frac{p(x|\theta)}{p(x|\theta)} - \Delta_\theta^\top \nabla_\theta \log p(x|\theta) - \frac{1}{2}\Delta_\theta^\top\left(\nabla^2_\theta \log p(x|\theta)\right)\Delta_\theta\right) \\ &= -\Delta_\theta^\top \sum_{x\in\mathcal{X}} p(x|\theta)\nabla_\theta \log p(x|\theta) - \frac{1}{2}\Delta_\theta^\top\left(\sum_{x\in\mathcal{X}} p(x|\theta)\nabla^2_\theta \log p(x|\theta)\right)\Delta_\theta \\ &= \frac{1}{2}\Delta_\theta^\top F_x(\theta)\Delta_\theta. \end{aligned} \tag{3.12}$$

Therefore, the constraint (3.5) assures that the distribution after updating the parameter, $p(x|\theta+\Delta_\theta)$, is close to the distribution before updating the parameter, $p(x|\theta)$.

Remarks and Recent Advances

Natural gradient methods are invariant to re-parameterization, that is, their updates produce the same sequence of functions even when their parameterizations are different. Learning rules with this property are called *covariant* (MacKay, 1996). It is known that natural gradient method is covariant is an infinitesimal step-size is used. Thomas, 2014b generalized the natural gradient to allow θ to lie on a semi-Riemannian manifold and established sufficient conditions for the convergence of natural gradient methods. In this case, the pseudo-inverse matrix of Fisher information matrix, $G^{+}(\theta)$, is multiplied to the vanilla gradient. Thomas et al., 2016 generalized the natural gradient to leverage the knowledge of both how the parametric probabilistic model is parameterized and what it is a distribution over.

3.1.2 Natural Policy Gradient

In this section, we describe the *natural policy gradient*. The natural policy gradient is the steepest direction on the Riemannian manifold which maximizes $\eta(\theta + \Delta_\theta)$, where the objective $\eta(\theta)$ is

$$\eta(\theta) = \sum_{s \in \mathcal{S}} \rho^{\pi}(s) \sum_{a \in \mathcal{A}} \pi(a|s;\theta)\mathcal{R}(s,a). \tag{3.13}$$

First, we consider the *vanilla policy gradient*. Partially differentiating (3.13) with respect to θ yields

$$\nabla_\theta \eta(\theta) = \sum_{s \in \mathcal{S}} \rho^{\pi}(s) \sum_{a \in \mathcal{A}} \pi(a|s;\theta) \left(\nabla_\theta \log \rho^{\pi}(s) + \nabla_\theta \log \pi(a|s;\theta)\right) \mathcal{R}(s,a). \tag{3.14}$$

Even though the state distribution $\rho^{\pi}(s)$ is dependent on the policy π and thus indirectly on the parameter θ, deriving $\nabla_\theta \log \rho^{\pi}(s)$ is non-trivial. Morimura et al., 2010a proposed a method for estimating $\nabla_\theta \log \rho^{\pi}(s)$ through *backward* Markov chain formulation in the average reward objective setting and to use (3.14) directly. Morimura et al., 2010a suggested that their method is a dual form of the actor critic algorithm proposed by Konda and Tsitsiklis, 2003, which uses the action value function $Q^{\pi}(s,a)$ instead of $\nabla_\theta \log \rho^{\pi}(s)$ to estimate $\nabla_\theta \eta(\theta)$. Similarly, in the approximate

dynamic programming literature, the dual forms was studied in which the probability distributions over state spaces or state-action spaces are estimated instead of the value functions (Littman, Dean, and Kaelbling, 1995; Wang, Bowling, and Schuurmans, 2007), and the convergence properties of the dual algorithms were investigated (Wang et al., 2008).

In this dissertation, we focus on the primal form, that is, we use the value functions to approximate the policy gradient. The following proposition is one of the fundamentals of the primal form, which relates the vanilla policy gradient $\nabla_\theta \eta(\theta)$ and the action value function $Q^\pi(s,a)$.

Proposition 3.2 (Policy Gradient Theorem (Sutton et al., 1999, Theorem 1))**.** *In either the discounted reward or average reward formulations, the vanilla policy gradient is given as follows:*

$$\nabla_\theta \eta(\theta) = \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - b(s)\right) \nabla_\theta \log \pi(a|s;\theta), \tag{3.15}$$

where $b(s)$ is a state-dependent arbitrary function termed a baseline function.

Proof. First, we consider the discounted reward formulation. From Eqs. (2.4) and (2.9), we have

$$\begin{aligned}
&\nabla_\theta V^\pi(s) && (3.16)\\
&= \nabla_\theta \sum_{a\in\mathcal{A}} \pi(a|s;\theta) Q^\pi(s,a) \\
&= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \nabla_\theta Q^\pi(s,a)\right) \\
&= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \nabla_\theta \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V^\pi(s')\right)\right) \\
&= \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \gamma \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \nabla_\theta \sum_{a'\in\mathcal{A}} \pi(a'|s';\theta) Q^\pi(s',a'). && (3.17)
\end{aligned}$$

Notice the recursive structure between (3.16) and (3.17). Then, unrolling above yields

$$
\begin{aligned}
&\nabla_\theta V^\pi(s) \\
&= \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) \\
&\quad + \gamma \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \times \\
&\qquad \sum_{a'\in\mathcal{A}} \nabla_\theta \pi(a'|s';\theta) Q^\pi(s',a') \\
&\quad + \gamma^2 \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a'\in\mathcal{A}} \pi(a'|s';\theta) \sum_{s''\in\mathcal{S}} \mathcal{P}(s''|s',a') \times \\
&\qquad \sum_{a''\in\mathcal{A}} \nabla_\theta \pi(a''|s'';\theta) Q^\pi(s'',a'') \\
&\quad + \cdots \\
&= \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) \\
&\quad + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}^{\pi,1}(s'|s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s';\theta) Q^\pi(s',a) \\
&\quad + \gamma^2 \sum_{s''\in\mathcal{S}} \mathcal{P}^{\pi,2}(s''|s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s'';\theta) Q^\pi(s'',a) \\
&\quad + \cdots \\
&\overset{(a)}{=} \sum_{s'\in\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \mathcal{P}^{\pi,n}(s'|s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s';\theta) Q^\pi(s',a) \\
&\overset{(b)}{=} \sum_{s'\in\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s'|s_0 = s,\pi) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a|s';\theta) Q^\pi(s',a),
\end{aligned}
$$

where the equality (a) follows from Eq. (2.17) and the equality (b) follows from Eq. (2.18). Summing both sides of the above over the initial state distribution $\rho_0(s)$, and

using Eq. (2.19), it yields that

$$\begin{aligned}
\nabla_\theta \eta(\theta) &= \sum_{s_0 \in \mathcal{S}} \rho_0(s') \nabla_\theta V^\pi(s') \\
&= \sum_{s_0 \in \mathcal{S}} \rho_0(s') \sum_{s \in \mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s | s_0 = s', \pi) \sum_{a \in \mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) \\
&= \sum_{s \in \mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s | \rho_0, \pi) \sum_{a \in \mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) \\
&= \sum_{s \in \mathcal{S}} \rho^\pi(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) \\
&= \sum_{s \in \mathcal{S}} \rho^\pi(s) \sum_{a \in \mathcal{A}} \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta) Q^\pi(s,a),
\end{aligned}$$

where in the last equality we used the fact that

$$\nabla_\theta \log \pi(a|s;\theta) = \frac{\nabla_\theta \pi(a|s;\theta)}{\pi(a|s;\theta)}. \tag{3.18}$$

Next, we consider the average reward formulation. From Eqs. (2.28) and (2.29), we have

$$\begin{aligned}
&\nabla_\theta V^\pi(s) \\
&= \nabla_\theta \sum_{a \in \mathcal{A}} \pi(a|s;\theta) Q^\pi(s,a) \\
&= \sum_{a \in \mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \nabla_\theta Q^\pi(s,a) \right) \\
&= \sum_{a \in \mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \nabla_\theta \left(\mathcal{R}(s,a) - \eta(\theta) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^\pi(s') \right) \right) \\
&= \sum_{a \in \mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \left(-\nabla_\theta \eta(\theta) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \nabla_\theta V^\pi(s') \right) \right).
\end{aligned}$$

Rearranging the above yields

$$\nabla_\theta \eta(\theta) = \sum_{a \in \mathcal{A}} \left(\nabla_\theta \pi(a|s;\theta) Q^\pi(s,a) + \pi(a|s;\theta) \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \nabla_\theta V^\pi(s') \right) - \nabla_\theta V^\pi(s). \tag{3.19}$$

Summing both sides of the above over the stationary distribution $\rho^\pi(s)$,

$$\begin{aligned}\sum_{s\in\mathcal{S}}\rho^\pi(s)\nabla_\theta\eta(\theta) &= \sum_{s\in\mathcal{S}}\rho^\pi(s)\sum_{a\in\mathcal{A}}\nabla_\theta\pi(a|s;\theta)Q^\pi(s,a)\\ &+\sum_{s\in\mathcal{S}}\rho^\pi(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\nabla_\theta V^\pi(s')\\ &-\sum_{s\in\mathcal{S}}\rho^\pi(s)\nabla_\theta V^\pi(s).\end{aligned}$$

Using the balance equation of the stationary distribution (2.22), it holds that

$$\begin{aligned}\sum_{s\in\mathcal{S}}\rho^\pi(s)\nabla_\theta\eta(\theta) &= \sum_{s\in\mathcal{S}}\rho^\pi(s)\sum_{a\in\mathcal{A}}\nabla_\theta\pi(a|s;\theta)Q^\pi(s,a)\\ &+\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s'\in\mathcal{S}}\rho^\pi(s)\pi(a|s;\theta)\mathcal{P}(s'|s,a)\nabla_\theta V^\pi(s')\\ &-\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}}\rho^\pi(s')\pi(a|s';\theta)\mathcal{P}(s|s',a)\nabla_\theta V^\pi(s)\\ &=\sum_{s\in\mathcal{S}}\rho^\pi(s)\sum_{a\in\mathcal{A}}\nabla_\theta\pi(a|s;\theta)Q^\pi(s,a).\end{aligned}$$

Finally, we acquire Eq. (3.15) because it follows from an argument similar to (3.10) that for any state-dependent function $b(s)$,

$$\begin{aligned}&\sum_{s\in\mathcal{S}}\rho^\pi(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta)b(s)\nabla_\theta\log\pi(a|s;\theta)\\ &=\sum_{s\in\mathcal{S}}\rho^\pi(s)b(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta)\nabla_\theta\log\pi(a|s;\theta)\\ &=\sum_{s\in\mathcal{S}}\rho^\pi(s)b(s)\sum_{a\in\mathcal{A}}\nabla_\theta\pi(a|s;\theta)\\ &=\sum_{s\in\mathcal{S}}\rho^\pi(s)b(s)\nabla_\theta\sum_{a\in\mathcal{A}}\pi(a|s;\theta)\\ &=\sum_{s\in\mathcal{S}}\rho^\pi(s)b(s)\nabla_\theta 1=0.\end{aligned}\tag{3.20}$$

■

Importanlty, even though the choice of $b(s)$ dose not affect $\nabla_\theta\eta(\theta)$, the variance in the action-value function estimator is minimized by regulating the baseline $b(s)$ appropriately, as we see later.

A straightforward procedure to acquire the natural policy gradient consists of (i)

computing the vanilla policy gradient $\nabla_\theta \eta(\theta)$ from (3.15), (ii) computing an appropriate Riemannian metric tensor $G(\theta)$ and its inverse $G^{-1}(\theta)$, and (iii) computing the product $G^{-1}(\theta)\nabla_\theta \eta(\theta)$. This naive implementation requires high computational complexity, $\mathcal{O}(d^3)$. However, it has been shown that the natural policy gradient can be estimated directly without even requiring the $d \times d$ matrix $G(\theta)$, and the required computational complexity is only $\mathcal{O}(d)$ (Kakade, 2001; Morimura, Uchibe, and Doya, 2005; Bhatnagar et al., 2009). The key factor is to use a *compatible function approximation* (Sutton et al., 1999) as we see in the below.

Now consider the case in which the action value function $Q^\pi(s,a)$ for a fixed policy $\pi(a|s;\theta)$ is approximated by a linear function approximator $f(s,a;w)$ defined by

$$f(s,a;w) \triangleq w^\top \psi(s,a) \triangleq w^\top \nabla_\theta \log \pi(a|s;\theta), \tag{3.21}$$

where $w \in \mathbb{R}^d$ is a parameter vector to be optimized and

$$\psi(s,a) \triangleq \nabla_\theta \log \pi(a|s;\theta) \tag{3.22}$$

is called a compatible feature or a *characteristic eligibility*. The approximator $f(s,a;w)$ is *compatible* in the sense that the following equation holds:

$$\nabla_w f(s,a;w) = \nabla_\theta \log \pi(a|s;\theta). \tag{3.23}$$

Let $\ell^\pi(w)$ be a mean squared error of the function approximation:

$$\ell^\pi(w) \triangleq \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - b(s) - f(s,a;w)\right)^2. \tag{3.24}$$

Note that $\ell^\pi(w)$ is convex with respect to w. Let $w^\pi = \arg\min_w \ell^\pi(w)$ denote the optimal parameter vector under the current policy $\pi(a|s;\theta)$. The optimal parameter w^π satisfies the following equation:

$$\sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - b(s) - f(s,a;w^\pi)\right) \nabla_w f(s,a;w^\pi) = 0. \tag{3.25}$$

Therefore, the compatibility (3.23) yields the following proposition.

Proposition 3.3 (Policy Gradient Theorem with Function Approximation (Sutton et al., 1999, Theorem 2)). *In either the discounted reward or average reward formulations, the vanilla policy gradient is given as follows:*

$$\nabla_\theta \eta(\theta) = \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) f(s,a;w^\pi) \nabla_\theta \log \pi(a|s;\theta). \tag{3.26}$$

Thus, policy gradient can be estimated by approximating $Q^\pi(s,a)$ projected on to the subspace spanned by $\nabla_\theta \log \pi(a|s;\theta)$.

Now, we consider the choice of the baseline $b(s)$. The purpose here is to obtain the minimum variance baseline in the action value function estimator under the current policy π and the given optimal parameter w^π. Formally, we aim to obtain $b^\pi(s) = \arg\min_b \ell^\pi(w^\pi)$. Partially differentiating (3.24) with respect to $b(s)$ under $w = w^\pi$ yields

$$\frac{\partial \ell^\pi(w^\pi)}{\partial b(s)} = -2 \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - b(s) - f(s,a;w^\pi)\right).$$

Equating the above to zero yields

$$\begin{aligned}
\sum_{s\in\mathcal{S}} \rho^\pi(s) b^\pi(s) &= \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) b^\pi(s) \\
&= \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - f(s,a;w^\pi)\right) \\
&= \sum_{s\in\mathcal{S}} \rho^\pi(s) V^\pi(s),
\end{aligned}$$

where the last equality follows from (2.4) or (2.28) and

$$\sum_{a\in\mathcal{A}} \pi(a|s;\theta) f(s,a;w^\pi) = {w^\pi}^\top \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta) = 0. \tag{3.27}$$

Therefore, the variance in the action-value function estimator is minimized when the baseline $b(s)$ is chosen to be the state value function $V^\pi(s)$ for all $s \in \mathcal{S}$. We obtain the following proposition from the above argument.

Proposition 3.4 (Sutton et al., 1999, Bhatnagar et al., 2009, Lemma 2). *For any stationary policy π, the minimum variance baseline $V^\pi(s)$ in the action value function estimator $f(s,a;w^\pi)$ is the state value function $V^\pi(s)$.*

Given Proposition 3.4 and (3.27), it is better to consider $f(s,a;w)$ to be approximation of the *advantage* function:

$$A^\pi(s,a) \triangleq Q^\pi(s,a) - V^\pi(s). \tag{3.28}$$

Note that from the definition (3.28) and (2.4) or (2.28), it holds that

$$\begin{aligned}\sum_{a\in\mathcal{A}} \pi(a|s;\theta) A^\pi(s,a) &= \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(Q^\pi(s,a) - V^\pi(s)\right) \\ &= \sum_{a\in\mathcal{A}} \pi(a|s;\theta) Q^\pi(s,a) - V^\pi(s) = 0.\end{aligned} \tag{3.29}$$

Note also that the definition of the advantage (3.28) is different from the original definition by Baird, 1993:

$$A^\pi(s,a) \triangleq Q^\pi(s,a) - \max_{a\in\mathcal{A}} Q^\pi(s,a).$$

Importantly, substituting Eq. (3.21) into Eq. (3.26) yields

$$\begin{aligned}\nabla_\theta \eta(\theta) &= \sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \left(w^{\pi\top} \nabla_\theta \log \pi(a|s;\theta)\right) \nabla_\theta \log \pi(a|s;\theta) \\ &= \left(\sum_{s\in\mathcal{S}} \rho^\pi(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta)^\top\right) w^\pi \\ &= \left(\sum_{s\in\mathcal{S}} \rho^\pi(s) F_a(s;\theta)\right) w^\pi \\ &\triangleq G(\theta) w^\pi,\end{aligned} \tag{3.30}$$

where

$$F_a(s;\theta) = \sum_{a\in\mathcal{A}} \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta) \nabla_\theta \log \pi(a|s;\theta)^\top \tag{3.31}$$

is the Fisher information matrix of the policy distribution $\pi(a|s;\theta)$ at a state s and thus

$$G(\theta) \triangleq \sum_{s\in\mathcal{S}} \rho^\pi(s) F_a(s;\theta) \tag{3.32}$$

is the Fisher information matrix of the current policy weighted by the state distribution $\rho^{\pi}(s)$. From Eq. (3.30), we obtain the following proposition.

Proposition 3.5 (Kakade, 2001, Theorem 1, Morimura, Uchibe, and Doya, 2005, Bhatnagar et al., 2009, Lemma 1)**.** *In either the discounted reward or average reward formulations, the natural policy gradient is given as follows:*

$$\tilde{\nabla}_{\theta}\eta(\theta) = G^{-1}(\theta)\nabla_{\theta}\eta(\theta) = w^{\pi}, \tag{3.33}$$

where the Riemannian metric $G(\theta)$ is given by (3.32)*.*

Proposition 3.5 indicates that when Eq. (3.25) holds, the parameter w^{π} of the compatible function approximator $f(s, a; w^{\pi})$ directly encodes the natural policy gradient. As stated in Section 2.1, the reinforcement learning problem we deal in this dissertation is equal to the optimization of sample path $\xi_{0:\infty} = (s_0, a_0, s_1, a_1, \dots)$. It was proved by Bagnell and Schneider, 2003; Peters, Vijayakumar, and Schaal, 2003 that the Riemannian metric (3.32) is is equal to the Fisher information matrix of the statistical model of the sample path $\xi_{0:\infty}$ in either the discounted reward or average reward formulations.

Remark 3.6. Given Proposition 3.5, we can update the policy parameter θ using the natural policy gradient by

$$\theta' \leftarrow \theta + \alpha w^{\pi}, \tag{3.34}$$

where α is a step-size. Therefore, we do not even need to estimate $d \times d$ matrix $G(\theta)$ and the required computational complexity to solve (3.34) is only $\mathcal{O}(d)$.

Importantly, it was shown by Kakade, 2001 that the update rule (3.34) approximates the greedy policy improvement step (2.39) in policy iteration.

Proposition 3.7 (Kakade, 2001, Theorem 3)**.** *Let θ' be a parameter obtained by the update* (3.34)*. Then it holds that*

$$\pi(a|s;\theta') = \pi(a|s;\theta)\left(1 + \alpha f(s, a; w^{\pi})\right) + \mathcal{O}(\alpha^2).$$

Recall that

$$f(s,a;w^\pi) \approx A^\pi(s,a) = Q^\pi(s,a) - \mathop{\mathbb{E}}_{a\sim\pi}\left[Q^\pi(s,a)\right]. \tag{3.35}$$

Thus, given Eq. (3.35), Proposition 3.7 indicates that the probabilities to choose the actions of which values are larger get larger, and the probabilities to choose the actions of which values are small get smaller. Therefore, the successive applications of the update (3.34) lead the policy to choose the "best" action instead of the "better" actions and the policy is moving toward the solution of the policy improvement step.

An Example of Natural Policy Gradient

Now, we show an example of the policy gradient and the natural policy gradient using a simple environment which has been used in a literature of the natural policy gradient methods (Kakade, 2001; Bagnell and Schneider, 2003; Morimura, Uchibe, and Doya, 2005). In this section, we choose to optimize the average reward defined by Eq. (2.23).

The state space is $\mathcal{S} = \{s_1, s_2\}$ and the action space is $\mathcal{A} = \{a_1, a_2\}$. The agent can take either of two actions in each state. The state transition probability and the reward function are shown in Tables 3.1 and 3.2, respectively. The policy is

TABLE 3.1: State transition probabilities on the MDP with two states

Current state	Action	Next state	
		s_1	s_2
s_1	a_1	1	0
s_1	a_2	0	1
s_2	a_1	0	1
s_2	a_2	1	0

TABLE 3.2: Reward function on the MDP with two states

Current state	Action	Reward
s_1	a_1	1
s_1	a_2	0
s_2	a_1	2
s_2	a_2	0

characterized by a parameter vector $\theta \in \mathbb{R}^2$, using the sigmoidal function in each

state:

$$\begin{cases} \pi(a_1|s_i;\theta_i) &= \dfrac{1}{1+\exp(-\theta_i)} \triangleq \pi_i \\ \pi(a_2|s_i;\theta_i) &= 1-\pi(a_1|s_i;\theta_i) = 1-\pi_i \end{cases}, \tag{3.36}$$

where $i \in \{1,2\}$. In this small MDP, we can explicitly compute the stationary distribution, $\rho^\pi(s) = (\rho^\pi(s_1), \rho^\pi(s_2))^\top \triangleq (\rho_1^\pi, 1-\rho_1^\pi)^\top$. Recall that $\rho^\pi(s)$ satisfies the balance equation:

$$\begin{aligned} \begin{pmatrix} \rho^\pi(s_1) \\ \rho^\pi(s_2) \end{pmatrix} &= \begin{pmatrix} \mathcal{P}(s_1|s_1,a_1) & \mathcal{P}(s_1|s_1,a_2) & \mathcal{P}(s_1|s_2,a_1) & \mathcal{P}(s_1|s_2,a_2) \\ \mathcal{P}(s_2|s_1,a_1) & \mathcal{P}(s_2|s_1,a_2) & \mathcal{P}(s_2|s_2,a_1) & \mathcal{P}(s_2|s_2,a_2) \end{pmatrix} \\ &\quad \times \begin{pmatrix} \pi(a_1|s_1;\theta_1) & 0 \\ \pi(a_2|s_1;\theta_1) & 0 \\ 0 & \pi(a_1|s_2;\theta_2) \\ 0 & \pi(a_2|s_2;\theta_2) \end{pmatrix} \begin{pmatrix} \rho^\pi(s_1) \\ \rho^\pi(s_2) \end{pmatrix}, \\ \begin{pmatrix} \rho_1^\pi \\ 1-\rho_1^\pi \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \pi_1 & 0 \\ 1-\pi_1 & 0 \\ 0 & \pi_2 \\ 0 & 1-\pi_2 \end{pmatrix} \begin{pmatrix} \rho_1^\pi \\ 1-\rho_1^\pi \end{pmatrix} \\ &= \begin{pmatrix} \pi_1 & 1-\pi_2 \\ 1-\pi_1 & \pi_2 \end{pmatrix} \begin{pmatrix} \rho_1^\pi \\ 1-\rho_1^\pi \end{pmatrix}. \end{aligned}$$

Solving above yields

$$\rho^\pi(s) = \left(\frac{1-\pi_2}{2-\pi_1-\pi_2}, \frac{1-\pi_1}{2-\pi_1-\pi_2} \right)^\top.$$

Therefore, the learning objective is given by

$$\begin{aligned}
\eta(\theta) &= \begin{pmatrix} \mathcal{R}(s_1,a_1) & \mathcal{R}(s_1,a_2) & \mathcal{R}(s_2,a_1) & \mathcal{R}(s_2,a_2) \end{pmatrix} \\
&\quad \times \begin{pmatrix} \pi(a_1|s_1;\theta_1) & 0 \\ \pi(a_2|s_1;\theta_1) & 0 \\ 0 & \pi(a_1|s_2;\theta_2) \\ 0 & \pi(a_2|s_2;\theta_2) \end{pmatrix} \begin{pmatrix} \rho^\pi(s_1) \\ \rho^\pi(s_2) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} \pi_1 & 0 \\ 1-\pi_1 & 0 \\ 0 & \pi_2 \\ 0 & 1-\pi_2 \end{pmatrix} \left(\frac{1-\pi_2}{2-\pi_1-\pi_2}, \frac{1-\pi_1}{2-\pi_1-\pi_2} \right)^\top \\
&= \frac{\pi_1 + 2\pi_2 - 3\pi_1\pi_2}{2-\pi_1-\pi_2}. \qquad (3.37)
\end{aligned}$$

The optimal decision making in this MDP is to choose a_2 in s_1 and a_1 in s_2. Thus, the optimal policy parameter is $\theta^* = (-\infty, \infty)^\top$ and the corresponding average reward is $\eta(\theta^*) = 2$. Even though this MDP is very simple, it is well known that the learning using the vanilla policy gradient suffers from a serious plateau phenomenon. Here we calculate the *analytical flows* of vanilla and natural policy gradient. Deriving (3.37) with respect to θ gives the vanilla policy gradient:

$$\nabla_\theta \eta(\theta) = \frac{(1-\pi_1)(1-\pi_2)}{(2-\pi_1-\pi_2)^2} \begin{pmatrix} \pi_1(2-3\pi_2) \\ \pi_2(4-3\pi_1) \end{pmatrix},$$

where we used the following relations:

$$\begin{aligned}
\frac{\partial}{\partial\theta_i}\pi(a_1|s_i;\theta_i) &= \pi_i(1-\pi_i), \\
\frac{\partial}{\partial\theta_i}\pi(a_2|s_i;\theta_i) &= -\pi_i(1-\pi_i), \\
\frac{\partial}{\partial\theta_i}\log\pi(a_1|s_i;\theta_i) &= \frac{1}{\pi(a_1|s_i;\theta_i)}\frac{\partial}{\partial\theta_i}\pi(a_1|s_i;\theta_i) = 1-\pi_i, \\
\frac{\partial}{\partial\theta_i}\log\pi(a_2|s_i;\theta_i) &= \frac{1}{\pi(a_2|s_i;\theta_i)}\frac{\partial}{\partial\theta_i}\pi(a_2|s_i;\theta_i) = -\pi_i.
\end{aligned}$$

Further, we calculate the weighted Fisher information matrix (3.32) of the policy (3.36) as follows:

$$
\begin{aligned}
&G(\theta) \\
&= \rho^\pi(s_1)\pi(a_1|s_1;\theta)\nabla_\theta \log \pi(a_1|s_1;\theta)\nabla_\theta \log \pi(a_1|s_1;\theta)^\top \\
&\quad + \rho^\pi(s_1)\pi(a_2|s_1;\theta)\nabla_\theta \log \pi(a_2|s_1;\theta)\nabla_\theta \log \pi(a_2|s_1;\theta)^\top \\
&\quad + \rho^\pi(s_2)\pi(a_1|s_2;\theta)\nabla_\theta \log \pi(a_1|s_2;\theta)\nabla_\theta \log \pi(a_1|s_2;\theta)^\top \\
&\quad + \rho^\pi(s_2)\pi(a_2|s_2;\theta)\nabla_\theta \log \pi(a_2|s_2;\theta)\nabla_\theta \log \pi(a_2|s_2;\theta)^\top \\
&= \rho^\pi(s_1)\pi(a_1|s_1;\theta)\begin{pmatrix}(1-\pi_1)^2 & 0\\ 0 & 0\end{pmatrix} + \rho^\pi(s_1)\pi(a_2|s_1;\theta)\begin{pmatrix}(-\pi_1)^2 & 0\\ 0 & 0\end{pmatrix} \\
&\quad + \rho^\pi(s_2)\pi(a_1|s_2;\theta)\begin{pmatrix}0 & 0\\ 0 & (1-\pi_2)^2\end{pmatrix} + \rho^\pi(s_2)\pi(a_2|s_2;\theta)\begin{pmatrix}0 & 0\\ 0 & (-\pi_2)^2\end{pmatrix} \\
&= \frac{(1-\pi_1)(1-\pi_2)}{2-\pi_1-\pi_2}\begin{pmatrix}\pi_1 & 0\\ 0 & \pi_2\end{pmatrix}.
\end{aligned}
$$

The inverse of the above matrix is given as:

$$
G^{-1}(\theta) = \frac{2-\pi_1-\pi_2}{\pi_1\pi_2(1-\pi_1)(1-\pi_2)}\begin{pmatrix}\pi_2 & 0\\ 0 & \pi_1\end{pmatrix}.
$$

Therefore, the natural policy gradient in this MDP is given by

$$
\begin{aligned}
&\tilde{\nabla}_\theta \eta(\theta) \\
&= G^{-1}(\theta)\nabla_\theta \eta(\theta) \\
&= \frac{2-\pi_1-\pi_2}{\pi_1\pi_2(1-\pi_1)(1-\pi_2)}\begin{pmatrix}\pi_2 & 0\\ 0 & \pi_1\end{pmatrix} \times \frac{(1-\pi_1)(1-\pi_2)}{(2-\pi_1-\pi_2)^2}\begin{pmatrix}\pi_1(2-3\pi_2)\\ \pi_2(4-3\pi_1)\end{pmatrix} \\
&= \frac{1}{2-\pi_1-\pi_2}\begin{pmatrix}2-3\pi_2\\ 4-3\pi_1\end{pmatrix}
\end{aligned}
$$

Figure 3.1 shows how much the parameter space is distorted. The red ellipses in the left graph are the same distances from the orange points measured by the Euclidean metrics; $\|\Delta\| = 1$. The blue ellipses in the right graph are also the "same"

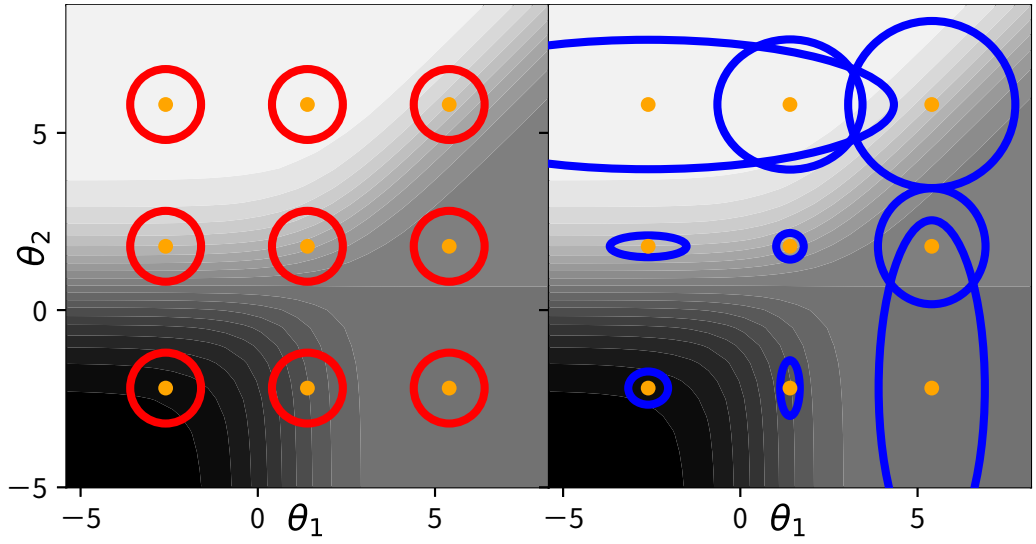


FIGURE 3.1: Visualization of the landscape in the parameter space. The ellipses represent the "same" distances from the orange points measured by the Euclidean metrics $\|\Delta\| = 1$ (left) and the Riemannian metrics $\Delta^\top G(\theta)\Delta = 0.1$ (right). The brightness of the background represents the average reward.

distances from the orange points measured by the Riemannian metrics; $\Delta^\top G(\theta)\Delta = 0.1$, where the Riemannian metric $G(\theta)$ is the weighted Fisher information matrix calculated by (3.31) and (3.32). Figure 3.2 shows the analytical gradient flows. The red arrows in the left graph are the analytical flow of the vanilla policy gradients (3.14). and the blue arrows in the right graph are the analytical flow of the natural policy gradients. The brightness of the background color represents the average reward. Suppose that the policy parameter is initially $\theta_0 = (1.4, -2.2)^\top$, which is shown as an orange dot in each of the left and right graph in the Figure 3.2. Then, using the vanilla policy gradient, the parameter shall steer away rightward in the graph area, that is, the policy remains to choose a_1 in s_1 and a_2 in s_2. This suboptimal policy stays in an extremely flat plateau with average reward $\eta = 1$. On the other hand, using the natural policy gradient, the policy quickly becomes to choose a_1 in s_1 and achieves average reward $\eta \approx 2$ by avoiding the plateau.

Remarks and Recent Advances

It was formally proven that the natural policy gradient is a covariant update rule (Bagnell and Schneider, 2003). Schulman et al., 2015 proposed trust region policy

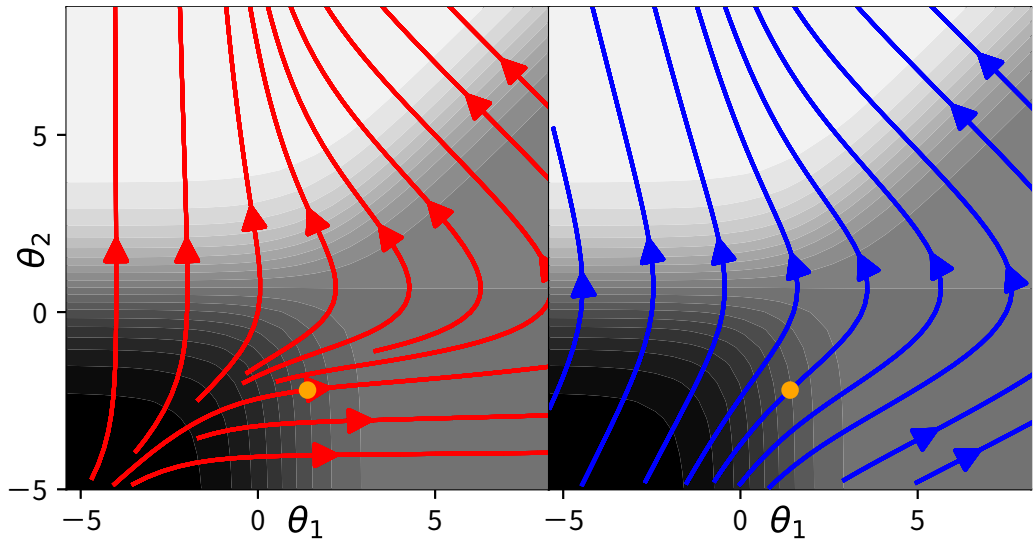


FIGURE 3.2: Analytical flows of the vanilla policy gradients (left) and the natural policy gradients (right) in the parameter space. The brightness of the background represents the average reward.

optimization method, which constraints the update direction of the policy and the step-size by KL divergence, and the gradient it uses is equivalent to the natural policy gradient. The natural policy gradient is widely used in recent policy-gradient-based deep reinforcement learning methods in the name of *trust region constraint* (Schulman et al., 2015; Duan et al., 2016; Wang et al., 2017; Chou, Maturana, and Scherer, 2017; Wu et al., 2017; Schulman et al., 2017; Nachum et al., 2017b; OpenAI, 2018). Importantly, Thomas, Dann, and Brunskill, 2018 generalized Proposition 3.7 to the broad class of gradient methods and gradient-like learning methods such as temporal difference (TD) learning. This generalization is not limited to reinforcement learning methods, but applicable to supervised and unsupervised learning methods.

3.2 Algorithms for the Estimation of Natural Policy Gradients

There are many possible ways to estimate the natural policy gradient. In the most straightforward way (Peters and Schaal, 2006; Peters and Schaal, 2008b), first the vanilla gradient is estimated (e.g. using GPOMDP (Baxter and Bartlett, 2001)) and the natural policy gradient is calculated according to Eq. (3.6) using the empirical

estimate of the Fisher information matrix. In another method (Peters, Vijayakumar, and Schaal, 2003; Peters and Schaal, 2005; Peters and Schaal, 2008a), w^π is estimated based on the least square temporal difference methods (Boyan, 1999; Bradtke and Barto, 1996). Both of above methods require at least $\mathcal{O}(d^2)$ computational complexity and memory. These methods are called *Natural Actor Critics*.

The focus of this dissertation is on the *Incremental Natural Actor Critic* (INAC) algorithms (Bhatnagar et al., 2009; Degris, Pilarski, and Sutton, 2012; Morimura, Uchibe, and Doya, 2005; Thomas, 2014a), which estimate the natural policy gradient (3.33) fully incrementally and require only $\mathcal{O}(d)$ computational complexity and memory. INAC algorithms use the state value function to estimate w^π. In the following, first we review *TD(λ)*, which is a method to estimate $V^\pi(s)$ with linear computational complexity and memory. Then, we present INAC algorithms.

3.2.1 Estimation of the State Value Function

Our goal here is to approximate the true state value function $V^\pi(s)$ of the current policy π. Let $V(s;v)$ be a parameterized state value function where $v \in \mathbb{R}^l$ is a parameter to be learned. We allow a shorthand notation $V_v \triangleq V(\cdot;v)$ when it is convenient. Let $\ell^\pi(v)$ be a squared loss of the function approximation:

$$\ell^\pi(v) \triangleq \sum_{s\in\mathcal{S}} \rho^\pi(s) \left(V^\pi(s) - V(s;v)\right)^2 . \tag{3.38}$$

The purpose is to find a locally optimal parameter v^π which minimizes the loss: $v^\pi = \arg\min_v \ell^\pi(v)$. Note that the true state value function $V^\pi(s)$ appears in the definition of the loss function (3.38) and is not available in the learning. Many methods have been proposed to replace $V^\pi(s)$ to estimated value.

First, suppose that the current time step is t and we have the return:

$$R_t = \sum_{\tau=0}^{\infty} \left(\gamma^\tau r_{t+\tau} - \bar{\eta}_t\right), \tag{3.39}$$

where $\gamma \in [0,1)$ and $\bar{\eta}_t = 0, \forall t$ for the discounted reward objective and $\gamma = 1$ and $\bar{\eta}_t$ is an estimate of the average reward for the average reward objective. Then we can use R_t as an unbiased estimate of V^π and update the parameter v along with the

direction

$$\Delta_{v,t}^{\mathrm{MC}} \propto (R_t - V(s_t; v))\, \nabla_v V(s_t; v). \tag{3.40}$$

The methods which use the return (3.39) as in (3.40) are called the *Monte Carlo* methods.

Unfortunately, the return R_t is not directly available at time step t because R_t depends on the future rewards. The alternative approach is the *bootstrapping*: to use the prediction by the current parameter v as an estimate of V^π. More specifically, we use the quantity $r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v)$ in place of V^π, that is,

$$\Delta_{v,t}^{\mathrm{TD}} \propto (r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v) - V(s_t; v))\, \nabla_v V(s_t; v). \tag{3.41}$$

The term

$$\delta_t \triangleq r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v) - V(s_t; v) \tag{3.42}$$

is called the *Temporal Difference* (TD) error and the methods which use (3.41) are called TD methods.

Next, we present the *TD(λ)*, which is a technique to unify Monte Carlo methods and TD methods (Sutton, 1988; Watkins, 1989). Here, $\lambda \in [0, 1]$ is a parameter that allows to interpolate Monte Carlo and TD: as we see later, $\lambda = 0$ yields TD while $\lambda = 1$ yields a Monte Carlo method. In this sense, TD(0) means the TD learning (3.41) and TD(1) means the Monte Carlo method (3.40).

Forward View: λ Return

TD(λ) is based on the idea of the multi-step look ahead of rewards. Let $R_t^{(n)}$ be the *n-step return* defined by

$$R_t^{(n)} \triangleq \sum_{\tau=0}^{n-1} (\gamma^\tau r_{t+\tau} - \bar{\eta}_t) + \gamma^n V(s_{t+n}; v). \tag{3.43}$$

From the definition, the n-step return consists of the actual rewards observed for n steps and later term of the return is bootstrapped by $\gamma^n V(s_{t+n}; v)$. Therefore we

have $R_t^{(1)} = r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v)$ for $n = 0$ and $R_t^{(n)} \to R_t$ as $n \to \infty$.

Importantly, as we see in the following, the expected value of the n-step return is a better estimate of $V^\pi(s)$ than the approximator $V(s; v)$ is. It holds that

$$\mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[R_t^{(n)}\middle| s_t = s\right] = \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[\sum_{\tau=0}^{n-1}(\gamma^\tau r_{t+\tau} - \bar{\eta}_t)\middle| s_t = s\right] + \sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s)\gamma^n V(s_{t+n} = s'; v).$$

Similarly, it holds that

$$\begin{aligned} V^\pi(s) &= \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[\sum_{\tau=0}^{\infty}(\gamma^\tau r_{t+\tau} - \bar{\eta}_t)\middle| s_t = s\right] \\ &= \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[\sum_{\tau=0}^{n-1}(\gamma^\tau r_{t+\tau} - \bar{\eta}_t)\middle| s_t = s\right] \\ &\quad + \sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s)\gamma^n \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[\sum_{\tau=0}^{\infty}(\gamma^\tau r_{t+n+\tau} - \bar{\eta}_t)\middle| s_{t+n} = s'\right]. \end{aligned}$$

Let $V^{(n)}(s; v) \triangleq \mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[R_t^{(n)}\middle| s_t = s\right]$. Then we have

$$\begin{aligned} \left\|V_v^{(n)} - V^\pi\right\|_\infty &= \sup_{s\in\mathcal{S}}\left|\mathop{\mathbb{E}}_{\pi,\mathcal{P}}\left[R_t^{(n)}\middle| s_t = s\right] - V^\pi(s)\right| \\ &= \gamma^n \sup_{s\in\mathcal{S}}\left|\sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s)\left(V(s'; v) - V^\pi(s')\right)\right| \\ &\le \gamma^n \sup_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s)\left|V(s'; v) - V^\pi(s')\right| \\ &\le \gamma^n \sup_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s)\left\|V_v - V^\pi\right\|_\infty \\ &= \gamma^n \left\|V_v - V^\pi\right\|_\infty. \end{aligned} \tag{3.44}$$

The last equality follows from $\sum_{s'\in\mathcal{S}}\mathcal{P}^{\pi,n}(s'|s) = 1$. The above relation (3.44) is called the *error-reduction property* of the n-step return.

Now, we present the *forward view* of TD(λ). Let R_t^λ be the λ-*return* defined by

$$R_t^\lambda \triangleq (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^{(n)}. \tag{3.45}$$

The λ-return is the weighted average of n-step returns, and decay factor $\lambda \in [0, 1]$ determines how each n-step return is weighted. Given the λ-*return*, the forward

view of TD(λ) is:

$$\Delta_{v,t}^{\mathrm{FW}} \propto \left(R_t^{\lambda} - V(s_t; v)\right) \nabla_v V(s_t; v). \tag{3.46}$$

Note that, analogous to the conventional return (2.5), the λ-return satisfies the following recursion:

$$\begin{aligned}
R_t^{\lambda} &= (1-\lambda)\left(R_t^{(1)} + \lambda R_t^{(2)} + \lambda^2 R_t^{(3)} + \cdots\right) \\
&= (1-\lambda)\left(r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v)\right) \\
&\quad + (1-\lambda)\lambda\left(r_t - \bar{\eta}_t + \gamma R_{t+1}^{(1)}\right) \\
&\quad + (1-\lambda)\lambda^2\left(r_t - \bar{\eta}_t + \gamma R_{t+1}^{(2)}\right) + \cdots \\
&= \left((1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\right)(r_t - \bar{\eta}_t) + \gamma(1-\lambda)V(s_{t+1}; v) + \gamma\lambda(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^{(n)} \\
&= r_t - \bar{\eta}_t + \gamma\left((1-\lambda)V(s_{t+1}; v) + \lambda R_{t+1}^{\lambda}\right)
\end{aligned} \tag{3.47}$$

Therefore, $R_t^{\lambda} = r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v)$ for $\lambda = 0$ and $R_t^{\lambda} = R_t$ for $\lambda = 1$.

Backward View: TD(λ) and Eligibility Traces

Next, we present the *backward view* of TD(λ). Let $e_{v,t} \in \mathbb{R}^l$ be a vector defined by

$$e_{v,t} = \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \nabla_v V(s_t; v). \tag{3.48}$$

The vector $e_{v,t}$ is called an *accumulating trace*, which is one of the variants of the *eligibility traces*. The backward view of TD(λ) is given as follows:

$$\Delta_{v,t}^{\mathrm{BW}} \propto \delta_t e_{v,t}, \tag{3.49}$$

where δ_t is the TD error defined in Eq. (3.42).

Importantly, the following proposition holds.

Proposition 3.8 (Equivalence of Forward and Backward Views (Sutton, 1988; Watkins, 1989))**.** *Provided that the approximated state value $V(s; v)$ is static and the policy $\pi(a|s; \theta)$ is static or converging to an equilibrium point, the overall update produced by the forward*

and back ward views are equivalent (:)

$$\sum_{t=0}^{\infty} \Big(R_t^{\lambda} - V(s_t; v) \Big) \nabla_v V(s_t; v) = \sum_{t=0}^{\infty} \delta_t e_{v,t}. \tag{3.50}$$

Proof. The overall update produced by the backward view is

$$\begin{aligned}
\sum_{t=0}^{\infty} \delta_t e_{v,t} &= \sum_{t=0}^{\infty} \delta_t \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \nabla_v V(s_t; v) \\
&= \delta_0 \nabla_v V(s_0; v) \\
&\quad + \delta_1 \left(\nabla_v V(s_1; v) + \gamma\lambda \nabla_v V(s_0; v) \right) \\
&\quad + \delta_2 \left(\nabla_v V(s_2; v) + \gamma\lambda \nabla_v V(s_1; v) + (\gamma\lambda)^2 \nabla_v V(s_0; v) \right) + \cdots \\
&= \nabla_v V(s_0; v) \left(\delta_0 + \gamma\lambda\delta_1 + (\gamma\lambda)^2 \delta_2 + \cdots \right) \\
&\quad + \nabla_v V(s_1; v) \left(\delta_1 + \gamma\lambda\delta_2 + \cdots \right) \\
&\quad + \nabla_v V(s_2; v) \left(\delta_2 + \cdots \right) + \cdots \\
&= \sum_{t=0}^{\infty} \nabla_v V(s_t; v) \sum_{\tau=t}^{\infty} (\gamma\lambda)^{\tau-t} \delta_\tau.
\end{aligned}$$

Furthermore, it holds that

$$
\begin{aligned}
& R_t^\lambda - V(s_t; v) \\
&= -V(s_t; v) + (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)} \\
&= -V(s_t; v) + (1-\lambda) \left(R_t^{(1)} + \lambda R_t^{(2)} + \lambda^2 R_t^{(3)} + \cdots \right) \\
&= -V(s_t; v) + (1-\lambda) \Big(r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v) \Big) \\
&\qquad + (1-\lambda)\lambda \Big(r_t - \bar{\eta}_t + r_{t+1} - \bar{\eta}_{t+1} + \gamma^2 V(s_{t+2}; v) \Big) \\
&\qquad + (1-\lambda)\lambda^2 \Big(r_t - \bar{\eta}_t + r_{t+1} - \bar{\eta}_{t+1} + r_{t+2} - \bar{\eta}_{t+2} + \gamma^3 V(s_{t+3}; v) \Big) + \cdots \\
&= -V(s_t; v) + (1-\lambda) \left(\left(\sum_{n=1}^{\infty} \lambda^{n-1} \right) (r_t - \bar{\eta}_t) + \gamma V(s_{t+1}; v) \right) \\
&\qquad + (1-\lambda)\gamma\lambda \left(\left(\sum_{n=1}^{\infty} \lambda^{n-1} \right) (r_{t+1} - \bar{\eta}_{t+1}) + \gamma V(s_{t+2}; v) \right) \\
&\qquad + (1-\lambda)(\gamma\lambda)^2 \left(\left(\sum_{n=1}^{\infty} \lambda^{n-1} \right) (r_{t+2} - \bar{\eta}_{t+2}) + \gamma V(s_{t+3}; v) \right) + \cdots \\
&= r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v) - V(s_t; v) \\
&\quad + \gamma\lambda \Big(r_{t+1} - \bar{\eta}_{t+1} + \gamma V(s_{t+2}; v) - V(s_{t+1}; v) \Big) \\
&\quad + (\gamma\lambda)^2 \Big(r_{t+1} - \bar{\eta}_{t+1} + \gamma V(s_{t+3}; v) - V(s_{t+1}; v) \Big) + \cdots \\
&= \sum_{\tau=t}^{\infty} (\gamma\lambda)^{\tau-t} \delta_\tau .
\end{aligned}
$$

■

Even though the equivalence of the forward and backward view breaks, the following heuristics is widely used where the state value parameter v is updated fully incrementally:

$$e_{v,t} = \gamma\lambda e_{v,t-1} + \nabla_v V(s_t; v_t), \tag{3.51}$$

$$\delta_t = r_t - \bar{\eta}_t + \gamma V(s_{t+1}; v_t) - V(s_t; v_t), \tag{3.52}$$

$$v_{t+1} = v_t + \alpha_v \delta_t e_{v,t}, \tag{3.53}$$

where α_v is the positive step-size. The convergence of TD(λ) for the discounted objective was proven by (Tsitsiklis and Roy, 1997) provided that the representation

of the state value is linear with respect to the parameter v and under some technical assumptions. An important extension of TD(λ) was proposed by (van Seijen and Sutton, 2014); they proposed a new forward view called the *n-step truncated λ-return*, and derived a new backward view which is equivalent to the new forward view even when the update of the state value is performed at every time step. The proposed algorithm is called *True Online TD(λ)*. (van Hasselt, Mahmood, and Sutton, 2014) extended the true online method to the off-policy learning.

3.2.2 Incremental Natural Actor Critic Algorithms

A number of algorithms have been proposed to estimate w^π satisfying Eq. (3.25), incrementally (Bhatnagar et al., 2009; Degris, Pilarski, and Sutton, 2012; Morimura, Uchibe, and Doya, 2005; Thomas, 2014a), which we refer to as INAC algorithms. In this section, we derive the general form of INAC iterations, and as special cases, we introduce the existing INAC algorithms.

A key to the derivation of INAC algorithms is the following Proposition 3.9 shown by Morimura, Uchibe, and Doya, 2005; Bhatnagar et al., 2009, which states that the *Temporal Difference (TD) error* can be used as an unbiased estimate for the advantage function:

Proposition 3.9. *The expectation of the TD error in the state-action space is equal to the advantage function:*

$$\mathop{\mathbb{E}}_{\pi,\mathcal{P}}[\delta^\pi|s,a] = A^\pi(s,a).$$

Proof. First, for the average reward objective, let the TD error δ^π be defined by

$$\delta^\pi \triangleq r - \eta(\theta) + V^\pi(s') - V^\pi(s). \tag{3.54}$$

Thus, the expectation of δ^π in the state-action space yields

$$\begin{aligned}\mathop{\mathbb{E}}_{\pi,\mathcal{P}}[\delta^\pi|s,a] &= \mathop{\mathbb{E}}_{\pi,\mathcal{P}}[r-\eta(\theta)+V^\pi(s')-V^\pi(s)|s,a] \\ &= \mathcal{R}(s,a)-\eta(\theta)+\sum_{s'\in\mathcal{S}}\mathcal{P}(s,a,s')V^\pi(s')-V^\pi(s) \\ &= Q^\pi(s,a)-V^\pi(s)=A^\pi(s,a).\end{aligned}$$

Next, for the discounted reward objective, let the TD error δ^π be defined by

$$\delta^\pi \triangleq r+\gamma V^\pi(s')-V^\pi(s). \tag{3.55}$$

Thus, the expectation of δ^π in the state-action space yields

$$\begin{aligned}\mathop{\mathbb{E}}_{\pi,\mathcal{P}}[\delta^\pi|s,a] &= \mathop{\mathbb{E}}_{\pi,\mathcal{P}}[r+\gamma V^\pi(s')-V^\pi(s)|s,a] \\ &= \mathcal{R}(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s,a,s')V^\pi(s')-V^\pi(s) \\ &= Q^\pi(s,a)-V^\pi(s)=A^\pi(s,a).\end{aligned}$$

■

Thus, the update direction Δ_w of w is given as the gradient to minimize the mean-squared error (MSE) of the TD error regression:

$$\begin{aligned}\Delta_w &\propto -\nabla_w \mathop{\mathbb{E}}_{\theta}\left[\frac{1}{2}\left(A^\pi-f(s,a;w)\right)^2\right] \\ &= \mathop{\mathbb{E}}_{\theta}\left[\left(A^\pi-f(s,a;w)\right)\nabla_w f(s,a;w)\right] \\ &= \mathop{\mathbb{E}}_{\theta}\left[\left(\delta^\pi-f(s,a;w)\right)\psi(s,a)\right].\end{aligned}$$

Finally, introducing the *eligibility traces* $e_{\delta,t}$ and $e_{f,t}$, we obtain the *general* form of INAC recursions:

$$w_{t+1}=w_t+\alpha\left(\delta_t e_{\delta,t}-f(s_t,a_t;w_t)e_{f,t}\right), \tag{3.56}$$

where $\alpha\in\mathbb{R}_+$ is the *step-size*. As we show later in Section 5, the values of α and $\|\psi\|$ are the keys for the stability of INAC algorithms. The main aim of this dissertation

is to weaken this sensitivity of INACs to the values of α and $\|\psi\|$. The eligibility traces were introduced to use the multi-step bootstrapping in an online manner. Note that the definitions of the estimated TD error δ_t and the eligibility traces $e_{\delta,t}, e_{f,t}$ are different for each algorithm, as shown in the following.

NTD: The natural policy gradient utilizing the temporal differences (NTD), which was proposed by Morimura, Uchibe, and Doya, 2005, was the first incremental and $\mathcal{O}(d)$ algorithm to estimate the NPG. The NTD algorithm produces a *biased* estimate of NPG for the *average* reward objective. In the NTD algorithm, the TD error and the eligibility traces are defined as follows:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{3.57}$$

$$e_{\delta,t} = e_{f,t} = \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \psi_\tau. \tag{3.58}$$

where V is the approximated state value function, $\lambda \in [0, 1]$ is the decay factor of the trace, and we give a shorthand notation, $\psi_t \triangleq \psi(s_t, a_t)$.

NAC-AP: Natural-gradient actor-critic with advantage parameters (NAC-AP) were proposed by Bhatnagar et al., 2009. The NAC-AP can produce an *unbiased* estimate of the NPG for the *average* reward objective, and its asymptotical convergence is proven. In NAC-AP, the TD error is computed using the estimate of the average reward, $\bar{\eta}$, and does not use the eligibility trace:

$$\delta_t = r_t - \bar{\eta}_t + V(s_{t+1}) - V(s_t), \tag{3.59}$$

$$e_{\delta,t} = e_{f,t} = \psi_t. \tag{3.60}$$

INAC: The INAC (Degris, Pilarski, and Sutton, 2012) is an extension of the NAC-AP, which uses the eligibility trace. The INAC also produces an *unbiased* estimate of the NPG for the *average* reward objective, and a *biased* estimate for the *discounted* reward

objective. The TD error and the eligibility traces are defined as follows:

$$\delta_t = r_t - \bar{\eta}_t + \gamma V(s_{t+1}) - V(s_t), \tag{3.61}$$

$$e_{\delta,t} = \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \psi_\tau, \quad e_{f,t} = \psi_t, \tag{3.62}$$

where $\gamma = 1$ for the average reward setting, and $\bar{\eta} = 0$ for the discounted setting.

NAC-S(λ): The natural actor critic using Sarsa(λ) (NAC-S(λ)) was proposed by Thomas, 2014a. NAC-S(λ) produces an *unbiased* estimate of NPG for the *discounted* reward objective. The TD error and the eligibility traces are defined as follows:

$$\begin{aligned} \delta_t &= r_t + \gamma V(s_{t+1}) - V(s_t), \\ e_{\delta,t} &= e_{f,t} = \sum_{\tau=0}^{t} \gamma^t \lambda^{t-\tau} \psi_\tau. \end{aligned} \tag{3.63}$$

As shown in Eq. (3.63), each ψ_τ term in the eligibility traces for NAC-S(λ) has an additional discount by a factor of γ^t, which enables the sampling from the γ-discounted future state distribution (2.16), thus generating the unbiased estimate for the discounted reward objective.

Algorithm 3.2.1 is a complete algorithms for the general form of INACs.

3.2.3 A Motivative Example for the Safety of INACs

Environment: The pendulum swinging up and the stabilization problem with limited torque (see Figure 3.3) is a well known benchmark in continuous state-action space RL (Doya, 2000; Morimura, Uchibe, and Doya, 2005). The state of the environment consists of an angle $q \in [-\pi, \pi]$ and an angular velocity $\dot{q} \in [-15, 15]$ of the pendulum, that is, $s = (q, \dot{q})^\top$. The action of the agent is applied as a torque to the pendulum after scaling, that is, $5a = \tau \in [-5, 5]$. The dynamics of the pendulum is given by

$$ml^2\ddot{q} = -\mu\dot{q} + mgl\sin(q) + \tau,$$

where $m = l = 1, g = 9.8$ and $\mu = 0.01$, and it is numerically integrated with $\Delta t = 0.02$. An episode lasts for 1000 steps and the initial state in each episode is

Algorithm 3.2.1: Incremental Natural Actor Critic (INAC)

initialization:
  parameterized policy $\pi(\cdot\,|\,\cdot\,;\theta)$;
  parameterized state value $V(\cdot\,;v)$;
  initial parameters θ_0, w_0, v_0;
  step-sizes $\alpha_\theta, \alpha, \alpha_v$;
  eligibility decay rate λ;
  if DISCOUNT **then**
    discount rate γ;
  else
    step-size for average reward $\alpha_{v,c} = c\alpha_v$ with a positive scalar c;
  Initialize eligibility traces: $e_{\delta,-1} = 0, e_{f,-1} = 0, e_{v,-1} = 0$;
  if not DISCOUNT **then**
    $\bar{\eta}_{-1} = 0$;
Draw initial state and action selection: $s_0 \sim \rho_0(\cdot),\ a_0 \sim \pi(\cdot|s_0;\theta_0)$;
for $t = 0, 1, 2, \ldots$ **do**
  Environmental step: $s_{t+1} \sim \mathcal{P}(\cdot\,|s_t, a_t),\ r_t = \mathcal{R}(s_t, a_t)$;
  Action selection: $a_{t+1} \sim \pi(\cdot|s_{t+1};\theta)$;
  if DISCOUNT **then**
    Compute TD error: $\delta_t = r_t + \gamma V(s_{t+1};v_t) - V(s_t;v_t)$;
  else
    Update average reward: $\bar{\eta}_t = (1 - \alpha_{v,c})\bar{\eta}_{t-1} + \alpha_{v,c} r_t$;
    Compute TD error: $\delta_t = r_t - \bar{\eta}_t + V(s_{t+1};v_t) - V(s_t;v_t)$;
  Update eligibility trace for state value: $e_{v,t} = \gamma\lambda e_{v,t-1} + \nabla_v V(s_t;v_t)$;
  Update eligibility traces for compatible function approximators $e_{\delta,t}, e_{f,t}$;
  ▷ `algorithm specific`
  $v_{t+1} = v_t + \alpha_v \delta_{w,t} e_{v,t}$;
  $w_{t+1} = w_t + \alpha\left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right)$;
  $\theta_{t+1} = \theta_t + \alpha_\theta w_t$;
return: locally optimal policy parameter $\theta^\star$;

$s_0 = (q_0, 0)^\top$, where q_0 is determined randomly. The reward function is defined as

$$\mathcal{R}(s) = \cos(q) - (\dot{q}/15\pi)^2,$$

and there is no penalty for over-rotation.

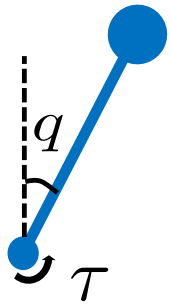


FIGURE 3.3: Pendulum swing up and stabilization with limited torque

Parameterization: The policy is a Gaussian distribution:

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

where the mean $\mu_\theta(s)$ and the standard deviation $\sigma_\theta(s)$ are determined by the output of a fully connected neural network with a single hidden layer. The input vector is $(\cos(q), \sin(q), \dot{q})^\top$, and the hidden layer has 10 sigmoidal units. The output layer consists of two units: the mean unit has a tanh activation and the standard deviation unit has a sigmoidal activation. A small constant value, $\sigma_0 = 0.01$, is added to the output of the standard deviation unit in order to prevent the divergence of ψ_t. The state value function is approximated by the 7th-order Fourier basis (Konidaris, Osentoski, and Thomas, 2011). The eligibility traces for both w and v are reset to zero vectors at the beginning of each episode.

Algorithms: We show the learning results for NTD algorithm. Thus, the TD error and the eligibility traces are defined as Eqs. (3.57) and (3.58), respectively.

Grid Search: We performed a grid search such that

$$\alpha, \alpha_v \in \{10^{-1}, 5 \cdot 10^{-2}, \ldots, 10^{-4}\},$$
$$\alpha_\theta \in \{10^{-4}, 5 \cdot 10^{-5}, \ldots, 10^{-7}\}.$$

The discount factor γ was set to 0.98 and the decay rate λ was set to 0.9. For each setting of the meta-parameters, the learning trials were conducted for 10 independent runs with different random seeds.

Results: Figure 3.4 shows the learning results for all the combinations of step-sizes in the grid search. The horizontal axes indicate the number of the episodes and the vertical axes indicate the average reward. If the estimate of even one run diverged, then the learning curve for that combination is truncated. Obviously, many learning trials resulted in the divergence of estimates and in many cases these divergences followed the improvement of the policy. Quantitatively speaking, 88% sets in the grid search were resulted in the divergence of the estimated parameters. This result highlights the difficulty of the tuning and the potential instability of INACs.

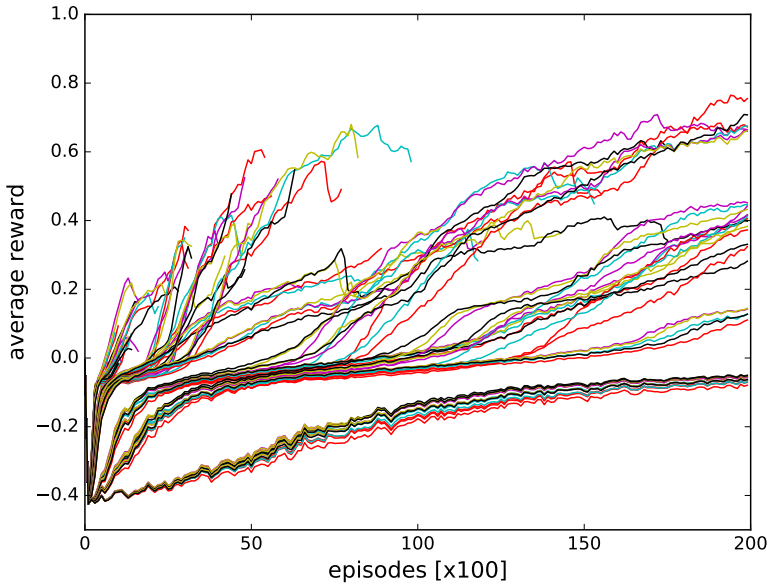


FIGURE 3.4: Learning curves for NTD in inverted pendulum domain with various settings of the step-sizes. The horizontal axis indicates the training episodes, and the vertical axis indicates the average reward. Each curve is the learning result for the different setting in the gris search. and is a mean of 10 independent runs with different random seeds. If the estimate of even one run diverged, then the learning curve for that combination is truncated.

Chapter 4

Adaptive Step-Size via Relative Importance Weighting

The step-size is a parameter of fundamental importance in learning algorithms, particularly for the natural policy gradient (NPG) methods. Recently, RL agents, whose value functions and policies are parameterized using deep neural networks, were shown to give remarkable performance (Mnih et al., 2015; Silver et al., 2016). The optimization of these networks is a highly non-linear problem, which makes *adaptive step-size* methods, such as AdaGrad (Duchi, Hazan, and Singer, 2011) and ADAM (Kingma and Ba, 2015), essential for the learning process. In previous RL studies, various adaptive step-size methods were proposed for policy gradient algorithms (Matsubara, Morimura, and Morimoto, 2010; Pirotta, Restelli, and Bascetta, 2013) and for conventional policy evaluation algorithms (Hutter and Legg, 2007; Dabney and Barto, 2012).

On the other hand, the NPG is estimated using a linear function approximation and a squared loss function. Though general adaptive step-size strategies such as AdaGrad and ADAM are applicable to NPG estimation, performance and stability can be further improved by using an adaptive step-size method that is specific to NPG estimation. To the best of our knowledge, despite its importance, such a special adaptive step-size method for NPG estimation has not been proposed to date.

Theoretically, the *step-sizes*, α_v, α and α_θ, must be decreasing positive values that satisfy $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ to guarantee convergence (Bertsekas and Tsitsiklis, 2000). In practice, however, the step-size is usually a fixed small value because

the RL problem is *not static*. In other words, the change of policy shifts the distribution of the state-action pair. This chapter focuses on the method for determining α.

The purpose of this paper is to propose an adaptive step-size strategy for NPG methods that can stabilize the NPG estimation in INACs. In this chapter, we focus on the Natural policy gradient utilizes the Temporal Differences (NTD) algorithm (Morimura, Uchibe, and Doya, 2005). Therefore, we assume the followings throughout this chapter:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{4.1}$$

$$e_{\delta,t} = e_{f,t} = \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \psi_\tau \triangleq e_t. \tag{4.2}$$

It is straightforward to extend the proposed method here to other INAC methods.

The structure of this chapter and our contributions are summarized as follows. In Section 4.1, we derive an upper bound of the step-size for NTD, α_t^*, which avoids oscillation and divergence of the estimated NPG. In particular, we focus on the NTD algorithm. This upper bound directly encodes a *local* optimum w_{local}^* for a given *single* sample. We also provide the tight upper bound and the lower bound for the step-size which consider the *global* effect of each sample, though they are not suitable for INACs. In Section 4.2, extending the approach of Karampatziakis and Langford, 2011, we propose an adaptive step-size method for general linear regression using the *trace* of the feature vector, which guarantees that an updated parameter does not overshoot the target. This is achieved by weighting the learning samples according to their relative importances. Next, in Section 4.2.1, in order to apply the derived upper bound α_t^*, we applied the derived general adaptive step-size method to NTD> The local optimum w_{local}^* is approached gradually by weighting the learning sample according to its given relative importance. The proposed step-size approaches the derived upper bound α_t^* as the relative importance approaches infinity. In other words, the proposed adaptive step-size determines the "aggressiveness" of the update from a given meta-parameter. Figure 4.1 outlines these two approaches. In Section 4.3, we evaluate the validity of the proposed adaptive step-size using classical benchmarks. Section 4.4 concludes this chapter.

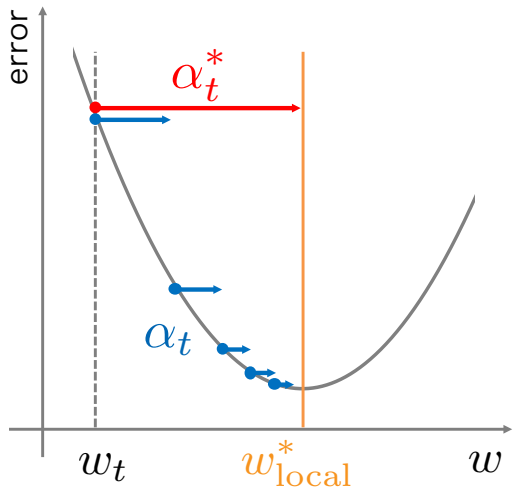


FIGURE 4.1: Overview of the derived upper bound and the proposed step-size method.

4.1 Upper Bound of Step-Size

First, the upper bound of the step-size α is derived. The underlying concept and derivations of the bounds are similar to Dabney and Barto, 2012, but the resulting bound is different. We simply compare the errors of the TD error regression *before* and *after* the update, and derive the upper bound of α to avoid overshooting the target.

Lemma 4.1. *The upper bound for non-negative step-size α is:*

$$\psi_t^\top e_t \leq 0 \quad \Rightarrow \quad \alpha = 0, \tag{4.3}$$

$$\psi_t^\top e_t > 0 \quad \Rightarrow \quad \alpha \leq \frac{1}{\psi_t^\top e_t} \triangleq \alpha_t^*. \tag{4.4}$$

Proof. Let w_t and w_{t+1} be w at times t and $t+1$, respectively. We adopt the shorthand notation $\psi_t \triangleq \psi(s_t, a_t)$. Then, the error of the TD error regression at time t *before* the update is:

$$\delta_{w,t} = \delta_t - w_t^\top \psi_t. \tag{4.5}$$

When $\delta_{w,t} = 0$, w is not updated. Assume that $\delta_{w,t} \neq 0$. Similarly, the error *after* the update is given by

$$\begin{aligned}\delta_{w,t'} &= \delta_t - w_{t+1}^\top \psi_t \\ &= \delta_t - \left(w_t + \alpha \delta_{w,t} e_t\right)^\top \psi_t \\ &= \delta_{w,t} - \alpha \delta_{w,t} \psi_t^\top e_t. \end{aligned} \tag{4.6}$$

When $\psi_t^\top e_t = 0$, the update does not affect $\delta_{w,t}$, and therefore

$$\psi_t^\top e_t = 0 \iff \delta_{w,t'} = \delta_{w,t}.$$

We are interested in the case where $\psi_t^\top e_t \neq 0$. If both $|\delta_{w,t'}| < |\delta_{w,t}|$ and $\text{sign}(\delta_{w,t'}) = \text{sign}(\delta_{w,t})$ are true, the error does not increase and the update does not overshoot the target. The errors before and after the update are then compared:

$$0 \leq \frac{\delta_{w,t'}}{\delta_{w,t}} \leq 1,$$

substituting (4.6) above yields

$$\begin{aligned} 0 &\leq 1 - \alpha \psi_t^\top e_t \leq 1, \\ 0 &\leq \alpha \psi_t^\top e_t \leq 1. \end{aligned} \tag{4.7}$$

When $\psi_t^\top e_t < 0$, the inequality (4.7) yields

$$\frac{1}{\psi_t^\top e_t} \leq \alpha \leq 0. \tag{4.8}$$

Thus, the requirement $\alpha \geq 0$ enforces $\alpha = 0$. Finally, when $\psi_t^\top e_t > 0$, the inequality (4.7) yields

$$0 \leq \alpha \leq \frac{1}{\psi_t^\top e_t} \triangleq \alpha_t^*.$$

■

Using the bound (4.4) as the step-size directly is too aggressive because the upper

bound given by Lemma 4.1 reflects only the *local* effect of the update using a given *single* sample. Thus, the bound (4.4) is safe only locally.

Theorem 4.2, given below, provides a tight upper bound that takes into account the *global* effect for the stationary policy.

Theorem 4.2. *If the policy is stationary,*

$$\alpha^*_{\tau,t} \leq \min_{\tau \in [0,t]} \frac{\delta_{w,\tau,t}}{\delta_{w,t} \psi_\tau^\top e_t} \tag{4.9}$$

is a tight upper bound for a non-negative step-size α, where $\delta_{w,\tau,t}$ is the τth experienced error before the tth update:

$$\delta_{w,\tau,t} = \delta_\tau - w_t^\top \psi_\tau.$$

Proof. We prove the theoren by contradiction. Assume the contrary proposition that there exists some $\alpha_{\mathrm{UB}} \in (0, \alpha^*_{\tau,t})$ such that α_{UB} is an upper bound for α. From this assumption, it follows that (i) using $\alpha^*_{\tau,t}$ at time t causes an error in the past, $\tau \in [0, t]$, to increase the absolute value or change the sign, (ii) while α_{UB} does not cause such an effect. The τth experienced error before the tth update is:

$$\delta_{w,\tau,t} = \delta_\tau - w_t^\top \psi_\tau.$$

Assume that $\delta_{w,\tau,t} \neq 0$. Similarly, the τth experienced error after the tth update using α is given by

$$\begin{aligned}
\delta_{w,\tau',t} &= \delta_\tau - w_{t+1}^\top \psi_\tau \\
&= \delta_\tau - \left(w_t + \alpha \delta_{w,t} e_t\right)^\top \psi_\tau \\
&= \delta_{w,\tau,t} - \alpha \delta_{w,t} \psi_\tau^\top e_t.
\end{aligned}$$

We compare the τth errors before and after the tth update:

$$\frac{\delta_{w,\tau',t}}{\delta_{w,\tau,t}} = 1 - \alpha \frac{\delta_{w,t}}{\delta_{w,\tau,t}} \psi_\tau^\top e_t = 1 - \frac{\alpha}{\eta}, \tag{4.10}$$

where $\eta = \frac{\delta_{w,\tau,t}}{\delta_{w,t} \psi_\tau^\top e_t}$. Then, the contrary assumption implies that there exists some $\tau \in$

$[0, t]$ such that $\delta_{w,\tau',t}/\delta_{w,\tau,t} > 1$ or $\delta_{w,\tau',t}/\delta_{w,\tau,t} < 0$ for $\alpha = \alpha^*_{\tau,t}$, and $0 \leq \delta_{w,\tau',t}/\delta_{w,\tau,t} \leq 1$ for $\alpha = \alpha_{\text{UB}}$.

Let $\alpha = \alpha^*_{\tau,t}$, then if $\delta_{w,\tau',t}/\delta_{w,\tau,t} < 0$, Eq. (4.10) yields $\alpha^*_{\tau,t}/\eta > 1$. This implies that there exists some η such that $\eta \in [0, \alpha^*_{\tau,t}]$, which is a contradiction because η is the element of the set in which $\alpha^*_{\tau,t}$ is the smallest value. If $\delta_{w,\tau',t}/\delta_{w,\tau,t} > 1$, Eq. (4.10) yields $\alpha^*_{\tau,t}/\eta < 0$. In this case $\eta < 0$ because $\alpha^*_{\tau,t} > 0$. Now we consider using $\alpha = \alpha_{\text{UB}}$ in this context:

$$\begin{aligned} 0 \leq & \quad \frac{\delta_{w,\tau',t}}{\delta_{w,\tau,t}} \quad \leq 1, \\ 0 \leq & \quad 1 - \frac{\alpha_{\text{UB}}}{\eta} \quad \leq 1, \\ \eta \leq & \quad \alpha_{\text{UB}} \quad \leq 0. \end{aligned}$$

This is again a contradiction because it was assumed that $\alpha_{\text{UB}} > 0$. Therefore, there exists no lower upper bound such that $\alpha_{\text{UB}} \in (0, \alpha^*_{\tau,t})$. ■

The lower bound is also given by Theorem 4.3.

Theorem 4.3. *If the policy is stationary,*

$$\alpha^*_{\tau,t} \geq \frac{1}{\|\delta_{w,t} e_t\| \max_{\tau \in [0,t]} \left\| \frac{\psi_\tau}{\delta_{w,\tau,t}} \right\|} \tag{4.11}$$

is a lower bound for a non-negative step-size α.

Proof. The least upper bound given by Theorem 4.2 yields

$$\begin{aligned} \alpha^*_{\tau,t} &= \min_{\tau \in [0,t]} \frac{\delta_{w,\tau,t}}{\delta_{w,t} \psi_\tau^\top e_t} \\ &= \left(\max_{\tau \in [0,t]} \frac{\delta_{w,t} \psi_\tau^\top e_t}{\delta_{w,\tau,t}} \right)^{-1} \\ &= \left(\max_{\tau \in [0,t]} \left| \left\langle \delta_{w,t} e_t, \frac{\psi_\tau}{\delta_{w,\tau,t}} \right\rangle \right| \right)^{-1}, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Then, the Cauchy-Schwarz inequality yields

$$\begin{aligned}\frac{1}{\alpha^*_{\tau,t}} &= \max_{\tau \in [0,t]} \left| \left\langle \delta_{w,t} e_t, \frac{\psi_\tau}{\delta_{w,\tau,t}} \right\rangle \right| \\ &\leq \max_{\tau \in [0,t]} \|\delta_{w,t} e_t\| \cdot \left\| \frac{\psi_\tau}{\delta_{w,\tau,t}} \right\| \\ &= \|\delta_{w,t} e_t\| \max_{\tau \in [0,t]} \left\| \frac{\psi_\tau}{\delta_{w,\tau,t}} \right\|.\end{aligned}$$

■

Note that the straightforward applications of both the upper (4.9) and lower bounds (4.11) require the computation and memory storage for all $\tau \in [0, t]$. One of the possible heuristics for applying the upper bound (4.9) with the memory and the computation of $\mathcal{O}(d)$ per time step is to take the minimum upper bound:

$$\alpha_t = \min \left[\alpha_{t-1}, \frac{1}{|\psi_t^\top e_t|} \right], \tag{4.12}$$

where $\alpha_0 = 1.0$, similar to the application of alpha bound (Dabney and Barto, 2012). However, we assume that the policy is updated at each iteration, and therefore our policy is *not* stationary. Thus, decreasing the step-size simply may be too conservative.

4.2 Adaptive Step-Size via Relative Importance Weighting

4.2.1 Adaptive Step-Size for the Linear Function Approximator with the Trace via Relative Importance Weighting

Next, we derive an adaptive step-size for the general linear-function approximation using the *trace* of a given feature vector. The aim is to propose the adaptive step-size that avoids an overshoot, and that can interpolate a very aggressive update and a less aggressive update according to given relative importance. The derivation is similar to Karampatziakis and Langford, 2011, but the resulting step-size is different because we consider the trace.

Let $x \in \mathbb{R}^d$ be a feature vector and $y \in \mathbb{R}$ be a target signal. We focus on a linear function approximator, $\hat{y} = w^\top x$, where $w \in \mathbb{R}^d$ is a parameter. Let $\ell(\hat{y}, y)$ be a

loss function. The goal is to find w^* such that $w^* = \arg\min_w \sum_t \ell(w^\top x_t, y_t)$, using stochastic gradient descent. Further, let $z_t \triangleq \sum_{\tau=0}^{t} \gamma^{t-\tau} x_\tau$ be a *trace* of the feature vector. Notice that though the arguments in this section hold for any bounded feature vector z_t, in this dissertation we consider the trace only. The update of w at time t using the standard stochastic gradient descent is

$$\begin{aligned} w_{t+1} &= w_t - \alpha \nabla_w \ell(w^\top x_t, y_t) \\ &= w_t - \alpha \frac{\partial \ell}{\partial \hat{y}}\bigg|_{\hat{y}=w^\top x_t} x_t, \end{aligned}$$

where α is again a small positive step-size. In this dissertation, however, we focus on the update using the trace of the feature vector:

$$w_{t+1} = w_t - \alpha \frac{\partial \ell}{\partial \hat{y}}\bigg|_{\hat{y}=w^\top x_t} z_t.$$

The derivation of the adaptive step-size begins with the following lemma where the relative importance is an integer.

Lemma 4.4. *Let $h \in \mathbb{N}_0$ be a relative importance. Updating w h times using a sample (x_t, y_t) is equivalent to the following update:*

$$w_{t+1} = w_t - \varsigma(h) z_t,$$

where the scaling factor $\varsigma(h)$ has the recursive form:

$$\begin{aligned} \varsigma(h+1) &= \varsigma(h) + \alpha \frac{\partial \ell}{\partial \hat{y}}\bigg|_{\hat{y}=(w_t - \varsigma(h) z_t)^\top x_t}, \\ \varsigma(0) &= 0. \end{aligned} \tag{4.13}$$

Proof. We prove the lemma by induction with respect to h. The initial case $h = 0$ is self-evidently true. The intermediate parameter updated h times is

$$w_{t'} = w_t - \varsigma(h) z_t.$$

Thus, after $h+1$ updates, the result can be written using $w_{t'}$ as follows:

$$\begin{aligned} w_{t+1} &= w_{t'} - \alpha \left.\frac{\partial \ell}{\partial \hat{y}}\right|_{\hat{y}=w_{t'}^\top x_t} z_t \\ &= w_t - \left(\varsigma(h) + \alpha \left.\frac{\partial \ell}{\partial \hat{y}}\right|_{\hat{y}=(w_t - \varsigma(h) z_t)^\top x_t}\right) z_t. \end{aligned}$$

■

This dissertation considers the squared loss, $\ell(\hat{y}, y_t) = \frac{1}{2}(y_t - \hat{y})^2$. Substituting this value into (4.13) yields

$$\begin{aligned} \varsigma(h+1) &= \varsigma(h) + \alpha \left(\left(w_t - \varsigma(h) z_t\right)^\top x_t - y_t \right) \\ &= \varsigma(h) \left(1 - \alpha x_t^\top z_t\right) + \alpha \left(w_t^\top x_t - y_t\right). \end{aligned}$$

Note that a recurrence relation of the form

$$\varsigma(h+1) = \varsigma(h) p + q,\ \varsigma(0) = 0$$

is solved by

$$\varsigma(h) = \frac{1 - p^h}{1 - p} q.$$

Therefore, we have

$$\begin{aligned} \varsigma(h) &= \frac{1 - \left(1 - \alpha x_t^\top z_t\right)^h}{1 - \left(1 - \alpha x_t^\top z_t\right)} \alpha \left(w_t^\top x_t - y_t\right) \\ &= \frac{1 - \left(1 - \alpha x_t^\top z_t\right)^h}{x_t^\top z_t} \left(w_t^\top x_t - y_t\right) \\ &= \frac{w_t^\top x_t - y_t}{x_t^\top z_t} \left(1 - \left(1 - \alpha x_t^\top z_t\right)^h\right). \end{aligned}$$

Furthermore, we fix h and consider updating w hK times, each with step-size α / K, where $K \in \mathbb{N}$:

$$\varsigma(hK) = \frac{w_t^\top x_t - y_t}{x_t^\top z_t} \left(1 - \left(1 - \frac{\alpha}{K} x_t^\top z_t\right)^{hK}\right). \tag{4.14}$$

In the limit $K \to \infty$,

$$\begin{aligned}\lim_{K\to\infty} \varsigma(hK) &= \lim_{K\to\infty} \frac{w_t^\top x_t - y_t}{x_t^\top z_t} \left(1 - \left(1 - \frac{h\alpha}{hK} x_t^\top z_t\right)^{hK}\right) \\ &= \frac{w_t^\top x_t - y_t}{x_t^\top z_t} \left(1 - \exp\left(-h\alpha x_t^\top z_t\right)\right).\end{aligned}$$

Thus, the update of w_t can be expressed as

$$w_{t+1} = w_t + \frac{1 - \exp(-h\alpha x_t^\top z_t)}{x_t^\top z_t} \left(y_t - w_t^\top x_t\right) z_t. \tag{4.15}$$

Next, we generalize Lemma 4.4 to the case $h \in \mathbb{R}_{\geq 0}$, as in Karampatziakis and Langford, 2011. By analogy with Eq. (4.14), the key idea is to perform K times more updates, each with a step-size smaller by a factor of K.

Theorem 4.5. *The limit of the gradient descent process with the trace for one sample using an infinitesimal step-size and a relative importance $h \in \mathbb{R}_{\geq 0}$ equals the update*

$$w_{t+1} = w_t - \varsigma(h) z_t,$$

where $\varsigma(h)$ satisfies the differential equation

$$\begin{aligned}\varsigma'(h) &= \alpha \frac{\partial \ell}{\partial \hat{y}}\bigg|_{\hat{y}=(w_t - \varsigma(h) z_t)^\top x_t}, \\ \varsigma(0) &= 0.\end{aligned} \tag{4.16}$$

Proof. First, Lemma 4.4 holds for a step-size α/K. The dependence on K is indicated explicitly by writing $\varsigma_{\alpha/K}(h)$ instead of $\varsigma(h)$:

$$\begin{aligned}\varsigma_{\alpha/K}(h+1) &= \varsigma_{\alpha/K}(h) + \frac{\alpha}{K} \Delta(\varsigma_{\alpha/K}(h)), \\ \varsigma_{\alpha/K}(0) &= 0,\end{aligned}$$

where

$$\Delta(\varsigma_{\alpha/K}(h)) = \frac{\partial \ell}{\partial \hat{y}}\bigg|_{\hat{y}=(w_t - \varsigma_{\alpha/K}(h) z_t)^\top x_t}.$$

Second, let $\varsigma_{\mathbb{Q}}(i) = \varsigma_{\alpha/K}(Ki)$ be a function whose argument is a non-negative rational number $i = h/K \in \mathbb{Q}_{\geq 0}$. Then, it follows that

$$\begin{aligned}\varsigma_{\mathbb{Q}}(i + 1/K) &= \varsigma_{\alpha/K}(Ki + 1) \\ &= \varsigma_{\alpha/K}(Ki) + \frac{\alpha}{K}\Delta(\varsigma_{\alpha/K}(Ki)) \\ &= \varsigma_{\mathbb{Q}}(i) + \frac{\alpha}{K}\Delta(\varsigma_{\mathbb{Q}}(i)). \end{aligned} \tag{4.17}$$

Note that the recurrence (4.17) corresponds to an K-fold multiple of updates, each with a step-size smaller by a factor of K. By rearranging (4.17), we have

$$\frac{\varsigma_{\mathbb{Q}}(i + 1/K) - \varsigma_{\mathbb{Q}}(i)}{1/K} = \alpha\Delta(s_{\mathbb{Q}}(i)). \tag{4.18}$$

Therefore, taking the limit $K \to \infty$ of Eq. (4.18) and changing the notation yields (4.16). ■

Again, in this dissertation, we focus on the squared loss $\ell(\hat{y}, y) = \frac{1}{2}(y - \hat{y})^2$. The differential equation (4.16) reduces to

$$\varsigma'(h) = \alpha\left((w_t - \varsigma(h)z_t)^\top x_t - y_t\right),$$

and is solved by

$$\varsigma(h) = \frac{w_t^\top x_t - y_t}{x_t^\top z_t}\left(1 - \exp(-h\alpha x_t^\top z_t)\right).$$

Thus, the update of w can be written

$$w_{t+1} = w_t + \frac{1 - \exp(-h\alpha x_t^\top z_t)}{x_t^\top z_t}\left(y_t - w_t^\top x_t\right) z_t, \tag{4.19}$$

which is equivalent to (4.15), except that h is a non-negative real number in Eq. (4.19).

Finally, we compare the errors before and after the update using Eq. (4.19). Assume that $y_t - w_t^\top x_t \neq 0$. The error after the update is:

$$\begin{aligned} & y_t - w_{t+1}^\top x_t \\ &= y_t - w_t^\top x_t - \frac{1 - \exp(-h\alpha x_t^\top z_t)}{x_t^\top z_t} \left(y_t - w_t^\top x_t\right) z_t^\top x_t \\ &= \left(y_t - w_t^\top x_t\right) \exp(-h\alpha x_t^\top z_t). \end{aligned}$$

Thus, the update (4.19) does not overshoot the given target:

$$\frac{y_t - w_{t+1}^\top x_t}{y_t - w_t^\top x_t} = \exp(-h\alpha x_t^\top z_t) > 0.$$

If $x_t^\top z_t > 0$, the update decreases the error. The *meta-step-size* α and the relative importance h determine the aggressiveness of the update. On the other hand, if $x_t^\top z_t < 0$, then the error after the update can become very large and can even diverge. Therefore, rejecting the sample becomes an option.

4.2.2 Adaptive Step-Size via Relative Importance Weighting for INAC

We apply the derived update (4.19) to INAC iteration (5.1) with (4.1) and . Substituting x_t, z_t and y_t for ψ_t, e_t and δ_t, respectively, Eq. (4.19) yields

$$\begin{aligned} & w \leftarrow w + \alpha_t \left(\delta_t - w^\top \psi_t\right) e_t, \\ & \alpha_t = \begin{cases} \dfrac{1 - \exp(-h\alpha \psi_t^\top e_t)}{\psi_t^\top e_t} & (\psi_t^\top e_t > 0) \\ 0 & (\psi_t^\top e_t \leq 0) \end{cases}. \end{aligned} \tag{4.20}$$

Remark 4.6. The derived step-size (4.20) lies within the upper bound (4.3,4.4). Furthermore, in the limit $h \to \infty$, the adaptive step-size (4.20) reduces to the upper bound (4.3,4.4). Thus, we can choose any relative importance $h \in [0, \infty)$ to each learning sample. Notice that for any choice of h, the proposed adaptive step-size is at least locally safe.

We refer the proposed adaptive step-size via relative importance weighting as RIW. Algorithm 4.2.1 shows a naive implementation of RIW. As explained in the

previous subsection, if $\psi_t^\top e_t < 0$, the update can cause a divergence of the estimated NPG. However, rejecting all these samples becomes very inefficient. In order to use all the samples, we can choose the conservative step-size strategy (4.12) if $\psi_t^\top e_t \leq 0$. Algorithm 4.2.2 shows the aggressive & conservative step-size strategy, where $\alpha_t = \frac{1-\exp(-h\alpha\psi_t^\top e_t)}{\psi_t^\top e_t}$ is used instead of $\frac{1}{|\psi_t^\top e_t|}$ in this implementation. Algorithm 4.2.3 show a procedure of NTD with RIW.

Algorithm 4.2.1: RIW with Obeying Bound (RIW-OB)

initialization:
  h, α, ψ_t, e_t;
if $\psi_t^\top e_t > 0$ **then**
  $\alpha_t = (1 - \exp(-h\alpha\psi_t^\top e_t))/\psi_t^\top e_t$;
else
  $\alpha_t = 0$;
return: adaptive step-size α_t;

Algorithm 4.2.2: RIW with Aggressive & Conservative (RIW-AC)

initialization:
  h, α, ψ_t, e_t;
  if FIRSTCALL **then**
    $\beta = 1$;
if $\psi_t^\top e_t > 0$ **then**
  $\alpha_t = (1 - \exp(-h\alpha\psi_t^\top e_t))/\psi_t^\top e_t$; ▷ `aggressive`
  if $\alpha_t < \beta$ **then**
    $\beta = \alpha_t$;
else
  $\alpha_t = \beta$; ▷ `conservative`
return: adaptive step-size α_t;

4.3 Numerical Experiment

In this section, we evaluate the validity of the proposed adaptive step-size method using classical benchmarks.

4.3.1 MDP with Two States

First, we confirm that the proposed method can estimate the NPG even though it does not use all the samples, that is, $\alpha_t = 0$ for $\psi_t^\top e_t \leq 0$. The task is an MDP with

Algorithm 4.2.3: NTD-RIW

initialization:
  parameterized policy $\pi(\cdot\,|\,\cdot\,;\theta)$;
  parameterized state value $V(\cdot\,;v)$;
  initial parameters θ_0, w_0, v_0;
  (meta-)step-sizes $\alpha_\theta, \alpha, \alpha_v$;
  eligibility decay rate λ, discount rate γ;
  RIW$\in$ {RIW-OB, RIW-AC};
  Initialize eligibility traces: $e_{\delta,-1} = 0, e_{f,-1} = 0, e_{v,-1} = 0$;
Draw initial state and action selection: $s_0 \sim \rho_0(\cdot)$, $a_0 \sim \pi(\cdot|s_0;\theta_0)$;
for $t = 0, 1, 2, \ldots$ **do**
  Environmental step: $s_{t+1} \sim \mathcal{P}(\cdot\,|s_t, a_t)$, $r_t = \mathcal{R}(s_t, a_t)$;
  Action selection: $a_{t+1} \sim \pi(\cdot|s_{t+1};\theta)$;
  Compute TD error: $\delta_t = r_t + \gamma V(s_{t+1};v_t) - V(s_t;v_t)$;
  Update eligibility trace for state value: $e_{v,t} = \gamma\lambda e_{v,t-1} + \nabla_v V(s_t;v_t)$;
  Update eligibility traces for compatible function approximators:
   $e_t = \gamma\lambda e_{t-1} + \psi_t$;
  Choose relative importance h;
  α_t = RIW(h,α, ψ_t, e_t);
  $v_{t+1} = v_t + \alpha_v \delta_{w,t} e_{v,t}$;
  $w_{t+1} = w_t + \alpha_t \left(\delta_t - f(s_t, a_t; w_t)\right) e_t$;
  $\theta_{t+1} = \theta_t + \alpha_\theta w_t$;
return: locally optimal policy parameter $\theta^\star$;

two states (Kakade, 2001; Bagnell and Schneider, 2003; Morimura, Uchibe, and Doya, 2005), which was already introduced in Section 3.1.2. The state transition law and reward function is summarized in Figure 4.2. Recall that the policy is characterized by $\theta \in \mathbb{R}^2$, using the sigmoidal function:

$$\begin{cases} \pi(a_1|s_i;\theta_i) & = 1/\left(1+\exp(-\theta_i)\right) \\ \pi(a_2|s_i;\theta_i) & = 1 - \pi(a_1|s_i;\theta_i) \end{cases}, \tag{4.21}$$

where $i \in \{1, 2\}$. The optimal decision making in this MDP involves choosing a_2 in s_1 and a_1 in s_2, and thus the optimal policy parameter is $\theta^* = (-\infty, \infty)^\top$. We choose the initial parameter $\theta_0 = (1.4, -2.2)^\top$, which will be trapped into a plateau without natural gradients. By using this environment, we can evaluate how fast the agent can estimate the NPG. In this experiment, we compare the proposed step-size with the fixed step-size. Figure 4.3 shows the learning results for the best meta-parameter values which yield the fastest learning without divergence. We used Algorithm 4.2.1 with fixed $h = 1$. Performance was evaluated by averaging over 100 runs.

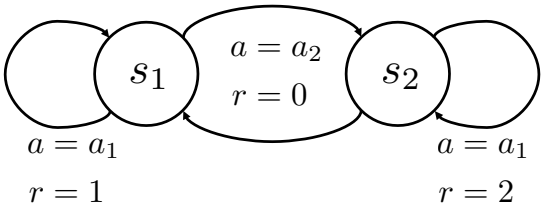


FIGURE 4.2: MDP with two states

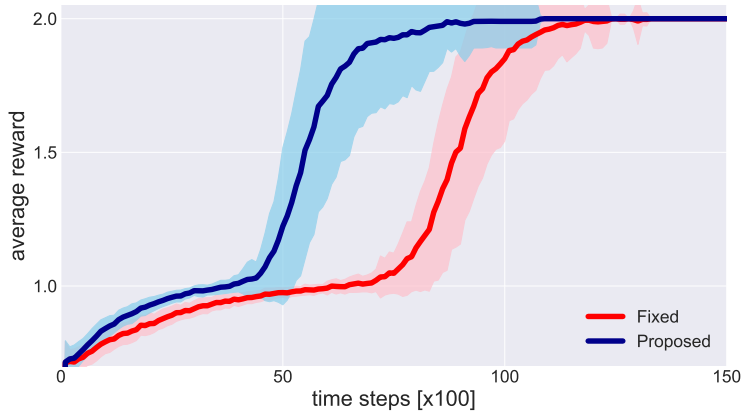


FIGURE 4.3: Learning curves in the MDP with two states. Mean and standard deviation of the average reward over 100 runs. The horizontal axis indicates the time steps and the vertical axis indicates the mean of the average reward.

4.3.2 Pendulum Swing Up and Stabilizing with Limited Torque

In the next experiment, we evaluated the robustness with regard to step-size tuning.

Setups

Environment and Parameterization: See Section 3.2.3.

Algorithms: The base algorithm we use is NTD, that is, the TD error and the eligibility traces are defined as Eqs. (4.1) and (4.2.2), respectively. Fixed step-size and the proposed adaptive step-size is applied. We use Algorithm 4.2.1 and $h = 1/(\|w_t\| + \|v_t\|)$. This choice of relative importance is justified because it satisfies $h \in [0, \infty)$. For comparison, we also applied AdaGrad (Duchi, Hazan, and Singer, 2011) and Adam (Kingma and Ba, 2015) to the estimation of NPG. More specifically,

we used the following equations for the update of w_t as AdaGrad and Adam, respectively:

$$\begin{aligned} \hat{g}_t &= \hat{g}_{t-1} + g_t^2, \\ w_{t+1} &= w_t + \alpha \frac{g_t}{\sqrt{\hat{g}_t} + \epsilon} \end{aligned} \tag{4.22}$$

and

$$\begin{aligned} \hat{g}_{m,t} &= \beta_1 \hat{g}_{m,t-1} + (1-\beta_1) g_t, \\ \hat{g}_{v,t} &= \beta_2 \hat{g}_{v,t-1} + (1-\beta_2) g_t^2, \\ w_{t+1} &= w_t + \alpha \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \frac{\hat{g}_{m,t}}{\sqrt{\hat{g}_{v,t}} + \epsilon}, \end{aligned} \tag{4.23}$$

where $\epsilon = 10^{-8}, \beta_1 = 0.9, \beta_2 = 0.999, \hat{g}_{-1} = \hat{g}_{m,-1} = \hat{g}_{v,-1} = e_{-1} = 0$ and

$$\begin{aligned} e_t &= \gamma \lambda e_{t-1} + \psi_t, \\ g_t &= \left(\delta_t - f(s_t, a_t; w_t)\right) e_t. \end{aligned}$$

Note that all of the operations above for the vectors are element-wise. In order to avoid the incomplete estimate of the NPG, the policy parameter is not updated within the first 100 episodes. The state value function is estimated by TD(λ) (Sutton, 1988).

Grid Search: We performed a grid search such that

$$\begin{aligned} \lambda &\in \{0, 0.5, 0.7, 0.9, 0.95, 0.99, 1.0\}, \\ \alpha, \alpha_v &\in \{10^{-1}, 5 \cdot 10^{-2}, \ldots, 10^{-4}\}, \\ \alpha_\theta &\in \{10^{-4}, 5 \cdot 10^{-5}, \ldots, 10^{-7}\}. \end{aligned}$$

The discount factor γ was set to 0.98. For each setting of the meta-parameters, the learning trials were conducted for 10 independent runs with different random seeds.

Results

Table 4.1 summarized the learning results for all the combinations of the meta-parameters. The divergence of NPG can avoided if AdaGrad of the proposed RIW is used. The column "< 0.0" indicates that the final performance (average reward in the last episode) is less than zero, which corresponds to the case that the pendulum keeps over rotating. The column "failures" indicates the sum of rates of divergence and "< 0.0", and means the rate that the learning completely failed. Table 4.1 indicates that RIW is more robust with regard to the value of meta-parameters than NTD, AdaGrad, and ADAM.

TABLE 4.1: Summary of Learning Results.

step-size	non-divergent	< 0.0	v diverged	w diverged	failures
Fixed	19.4%	5.6%	34.5%	46.1%	86.2%
AdaGrad	80.0%	33.8%	20.0%	0.0%	53.8%
Adam	50.6%	11.7%	48.8%	0.6%	61.1%
RIW	76.1%	17.7%	23.9%	0.0%	41.6%

4.4 Closing Remarks

In this chapter, we derived the upper bound of the step-size used for the INAC algorithms, especially for NTD. and proposed an adaptive step-size method that weights the learning samples according to their relative importance, in order to implement the derived upper bound. The proposed adaptive step-size determines its aggressiveness from the given meta-parameter and is guaranteed that it is safe locally. Numerical experiments validated the proposed method. We also provided the tight upper and lower bound for the step-size, though they are not suitable for the incremental learning where the policy keeps changing in each step. To the best of our knowledge, this is the first adaptive step-size method used for NPG estimation.

An interesting extension to this work is to seek for another criterion to determine h. One possible criterion for choosing h is the derivative of the logarithmic stationary distribution, $\nabla_\theta \ln d^\pi(s)$, which can be estimated as in Morimura et al., 2010a.

By regulating h adaptively depending on the state distribution even indirectly, the approximation of the expectation would be accelerated.

Chapter 5

Implicit Incremental Natural Actor Critic

In the previous chapter, we proposed an adaptive step-size method called RIW and we confirmed by the numerical experimental that by using the proposed method the divergence of the estimated NPG can avoided. However, though RIW satisfies the upper bound and safety is guaranteed, RIW has a big drawback; RIW is very sample inefficient because it does not update NPG if $\psi_t^\top e_t \leq 0$. Further, notice that there still remains a open question: **why are INACs unstable?** The main research aim of this chapter is to answer this question.

The goal of this chapter is to determine the reason for which existing INAC algorithms are unstable, and to propose an incremental and stable algorithm for the NPG estimation. The proposed method, which we refer to as an *implicit incremental natural actor critic* (I2NAC), is motivated by the ideas of the implicit stochastic gradient descent (Toulis, Rennie, and Airoldi, 2014; Toulis and Airoldi, 2015; Toulis, Tran, and Airoldi, 2016), and especially of the implicit temporal differences (Tamar et al., 2014). Theoretical analysis indicates the stability of I2NAC and the instability of the existing INAC methods. In a classical benchmark task, it is shown that I2NAC is less sensitive to the values of the meta-parameters, including the step-size for NPG estimation.

The structure of this chapter and our contributions are summarized as follows. In Section 5.1, we derive I2NAC, which is a new incremental and stable algorithm, to estimate the NPG. The existing INAC algorithms can be generally extended to I2NAC. After the asymptotic convergence analysis of I2NAC is shown in Section

5.2, we compare the stabilities of I2NAC and INACs theoretically in Section 5.3, where we highlight the reason for which INACs are unstable, and the stability of I2NAC iterations is shown. In Section 5.4, we present numerical experiments using the classical benchmark to verify the stability of I2NAC. The experimental result suggests that I2NAC can avoid the divergence of the estimated NPG and the policy degradation phenomenon. Finally, Section 5.5 concludes the paper.

5.1 Implicit Incremental Natural Actor Critic

In this section, we present our proposed method, i.e., the *implicit incremental natural actor critic* (I2NAC), which is based on the ideas of the implicit stochastic gradient descent (Toulis, Rennie, and Airoldi, 2014; Toulis and Airoldi, 2015; Toulis, Tran, and Airoldi, 2016; Toulis and Airoldi, 2015; Toulis, Tran, and Airoldi, 2016), and especially the implicit temporal differences (Tamar et al., 2014). Recall that INAC iteration is given as

$$w_{t+1} = w_t + \alpha \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right). \tag{5.1}$$

First, we add two terms to INACs' update, which sum to zero:

$$w_{t+1} = w_t + \alpha \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right) + \iota \left(w_t^\top e_t - w_t^\top e_t\right) e_t,$$

where $\iota \geq \alpha$ and an eligibility trace e_t can be $e_{\delta,t}$ or $e_{f,t}$. Here, we introduce the *implicit* update:

$$w_{t+1} = w_t + \alpha \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right) + \iota \left(w_t^\top e_t - w_{t+1}^\top e_t\right) e_t. \tag{5.2}$$

Eq. (5.2) is implicit in the sense that the parameter after the update, that is, w_{t+1}, appears on both sides of the equation. We call ι as an *implicit rate*. Note that the fixed point of Eq. (5.2) is the same as the fixed point of (5.1). It follows that

$$\left(I + \iota e_t e_t^\top\right) w_{t+1} = \left(I + \iota e_t e_t^\top\right) w_t + \alpha \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right),$$

where the matrix $I + \iota e_t e_t^\top$ is positive definite. Finally, using the Sherman-Morrison formula (A.2), we have the *implicit incremental natural actor critic* (I2NAC):

$$w_{t+1} = w_t + \alpha \left(I + \iota e_t e_t^\top\right)^{-1} \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right) \tag{5.3}$$

$$= w_t + \alpha \left(I - \frac{\iota}{1 + \iota \|e_t\|^2} e_t e_t^\top\right) \left(\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right). \tag{5.4}$$

The difference between I2NAC and INACs is only the multiplication of the matrix $I - \frac{\iota}{1+\iota\|e_t\|^2} e_t e_t^\top$. All of the INAC algorithms of the form (5.1) can be converted into I2NAC. Note that the required memory and computation to solve (5.4) remain $\mathcal{O}(d)$ because (5.4) can be solved only by computing the inner products, scalar multiplication and summation or subtraction of vectors. Algorithm 5.1.1 gives the overall procedure for I2NAC.

5.2 Asymptotic Convergence Analysis

In this section, we present the convergence analysis for I2NAC. The convergence is provided in a manner that is similar to that done by Bhatnagar et al., 2009. Here, we introduce the additional restriction on the problem formulation, and show that I2NAC generates a locally optimal policy when $e_{f,t} = \psi_t$. This convergence proof is based on the two-timescale method (Borkar, 1997) and the ODE method (Borkar and Meyn, 2000) for a stochastic approximation.

First, for the purpose of the analysis, we transform the problem from the maximization of the reward into the minimization of the cost. In the recursion (5.4), this is accomplished by simply taking the negative of the TD error:

$$w_{t+1} = w_t + \alpha \left(I - \frac{\iota}{1 + \iota \|e_t\|^2} e_t e_t^\top\right) \left(-\delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}\right). \tag{5.5}$$

Let the step-size α_t be time dependent, and

$$G_t^{\iota,e} \triangleq I - \frac{\iota}{1 + \iota \|e_t\|^2} e_t e_t^\top,$$

Algorithm 5.1.1: Implicit Incremental Natural Actor Critic (I2NAC)

initialization:
  parameterized policy $\pi(\cdot\,|\,\cdot\,;\theta)$;
  parameterized state value $V(\,\cdot\,;v)$;
  initial parameters θ_0, w_0, v_0;
  step-sizes $\alpha_\theta, \alpha, \alpha_v$;
  eligibility decay rate λ;
  if DISCOUNT **then**
    discount rate γ;
  else
    step-size for average reward $\alpha_{v,c} = c\alpha_v$ with a positive scalar c;
  implicit rate $\iota(\geq \alpha)$; Initialize eligibility traces:
  $e_{\delta,-1} = 0, e_{f,-1} = 0, e_{v,-1} = 0$;
  if not DISCOUNT **then**
    $\bar{\eta}_{-1} = 0$;
Draw initial state and action selection: $s_0 \sim \rho_0(\cdot)$, $a_0 \sim \pi(\cdot|s_0;\theta_0)$;
for $t = 0, 1, 2, \ldots$ **do**
  Environmental step: $s_{t+1} \sim \mathcal{P}(\,\cdot\,|s_t, a_t)$, $r_t = \mathcal{R}(s_t, a_t)$;
  Action selection: $a_{t+1} \sim \pi(\cdot|s_{t+1};\theta)$;
  if DISCOUNT **then**
    Compute TD error: $\delta_t = r_t + \gamma V(s_{t+1};v_t) - V(s_t;v_t)$;
  else
    Update average reward: $\bar{\eta}_t = (1 - \alpha_{v,c})\bar{\eta}_{t-1} + \alpha_{v,c} r_t$;
    Compute TD error: $\delta_t = r_t - \bar{\eta}_t + V(s_{t+1};v_t) - V(s_t;v_t)$;
  Update eligibility trace for state value: $e_{v,t} = \gamma\lambda e_{v,t-1} + \nabla_v V(s_t;v_t)$;
  Update eligibility traces for compatible function approximators $e_{\delta,t}, e_{f,t}$;
  ▷ `algorithm specific`
  Let $e_t \in \{e_{\delta,t}, e_{f,t}\}$;
  $v_{t+1} = v_t + \alpha_v \delta_{w,t} e_{v,t}$;
  $\Delta = \delta_t e_{\delta,t} - f(s_t, a_t; w_t) e_{f,t}$;
  $w_{t+1} = w_t + \alpha(\Delta - \iota/(1 + \iota\|e_t\|^2) e_t e_t^\top \Delta)$;
  ▷ `calculate from right to left`
  $\theta_{t+1} = \theta_t + \alpha_\theta w_t$;
return: locally optimal policy parameter $\theta^\star$;

which is positive definite. We choose $e_{f,t} = \psi_t$. Then, Eq. (5.5) can be rewritten as follows:

$$w_{t+1} = \left(I - \alpha_t G_t^{\iota,e} \psi_t \psi_t^\top\right) w_t - \alpha_t \delta_t G_t^{\iota,e} e_{\delta,t}. \tag{5.6}$$

Next, let C^π be a compact set such that $C^\pi = \{\theta | g_i(\theta) \leq 0, i = 1, \ldots, m\} \subset \mathbb{R}^d$, where the functions $g_i(\cdot), i = 1, \ldots, m$ represent the constraints that C^π satisfies. Let Γ be a projection $\Gamma : \mathbb{R}^d \to C^\pi$, such that for any $\theta \in \mathbb{R}^d$, $\Gamma(\theta) \in C^\pi$ and for $\theta \in C^\pi$, $\Gamma(\theta) = \theta$. We assume that the policy parameter is updated as follows:

$$\theta_{t+1} = \Gamma\left(\theta_t + \alpha_{\theta,t} w_t\right), \tag{5.7}$$

where $\alpha_{\theta,t}$ is the time-dependent step-size for the policy parameter. We require the following assumptions.

Assumption 5.1. *Approximated TD error δ_t is uniformly bounded by a constant C_δ.*

Assumption 5.2. *There exists a constant C_{e^δ} such that $\|e_{\delta,t}\| < C_{e^\delta} \|\psi_t\|$.*

The traces (3.58), (3.60), (3.62), and (3.63) satisfy Assumption 5.2.

Assumption 5.3. *The eligibility ψ_t is uniformly bounded, and the matrix $\psi_t \psi_t^\top$ is positive definite.*

Assumption 5.4. $\mathop{\mathbb{E}}_{\theta}\left[\delta_t^\pi e_{\delta,t}\right] = \nabla_\theta \eta(\theta)$ *holds.*

Note that, for example, the conventional cumulative trace such as Eq. (3.58) satisfies Assumption 5.4.

Assumption 5.5. *The two schedules for the step-sizes satisfy*

$$\sum_t \alpha_t = \sum_t \alpha_{\theta,t} = \infty, \quad \sum_t \alpha_t^2 < \infty, \sum_t \alpha_{\theta,t}^2 < \infty, \quad \alpha_{\theta,t} = o(\alpha_t). \tag{5.8}$$

Assumption 5.6. *The sequence $(s_t, s_{t+1}, r_t)_{t \geq 0}$ is i.i.d., and has uniformly bounded second moments.*

Even though it is not practical, the assumption that the sequence $(s_t, s_{t+1}, r_t)_{t \geq 0}$ is i.i.d. has been used in the literature on RL (Sutton, Szepesvári, and Maei, 2008;

Degris, White, and Sutton, 2012). The extension for the Markov noise case is outside of the scope of this study.

As we see later, given Assumptions 3.1 and 5.1 - 5.6, it is possible to show that w_t obtained by I2NAC converges to some $\tilde{w}$ almost surely. However, in order to ensure that $\tilde{w} = w^\pi$, we require the implicit rate ι to be time dependent, and the following mild assumption for the sequence $\{\iota_t\}$. Under Assumption 5.7 below, ι_t decreases much slower than α_t. Furthermore, it holds that $G_t^{\iota,e} \to I$ as $t \to \infty$.

Assumption 5.7. *The sequence of implicit rates $\{\iota_t\}$ satisfies the following properties:* (1) $\iota_t \to 0$ *as* $t \to \infty$, (2) $\iota_t \geq \alpha_t, \forall t$, *and* (3) $\alpha_t = o(\iota_t)$.

We have the following theorem, which states that the I2NAC iteration converges to w^π almost surely. Note that this result is also generally applicable to INAC algorithms by simply setting $G_t^{\iota,e} = I$.

Theorem 5.8. *Under a given parameter θ, w_t obtained from the recursion (5.6) satisfies $w_t \to -G^{-1}(\theta)\nabla_\theta \eta(\theta)$ as $t \to \infty$ almost surely.*

Proof. First, we consider the following ordinary differential equation (ODE) that is associated with (5.6) for a given θ:

$$\dot{w} = \mathop{\mathbb{E}}_{\theta}\left[-G_t^{\iota,e}\psi_t\psi_t^\top w - \delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right] \triangleq h(w). \tag{5.9}$$

The function $h(w)$ is Lipschitz continuous with respect to w owing to the boundedness of the reward and the eligibility. Let $h_\infty(w) = \lim_{c\to\infty} h(cw)/c$. The function $h_\infty(w)$ exists and satisfies $h_\infty(w) = -\mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right] w$. For the ODE $\dot{w} = -\mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right] w$, the origin is an asymptotically stable equilibrium with $L_1(w) = \|w\|^2/2$ as the associated Lyapunov function because $\mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right]$ is positive definite.

Next, let $\{M_t\}$ be a martingale sequence,

$$\begin{aligned} M_t = &\left(-G_t^{\iota,e}\psi_t\psi_t^\top w_t - \delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right) \\ &+ \mathop{\mathbb{E}}_{\theta}\left[-G_t^{\iota,e}\psi_t\psi_t^\top w_t - \delta_t^\pi G_t^{\iota,e} e_{\delta,t} | \mathcal{F}_t\right], \end{aligned}$$

where $\mathcal{F}_t = \sigma(w_\tau, M_\tau, \tau \leq t)$. Under the assumptions pertaining to the boundedness, it is easy to verify that there exists a constant $C < \infty$ such that

$$\mathop{\mathbb{E}}_{\theta}\left[\|M_t\|^2 | \mathcal{F}_t\right] \leq C(1 + \|w_t\|^2), \quad t \geq 0.$$

Finally, for the ODE (5.9), consider the following function

$$L_2(w) = \frac{1}{2}\|w - \tilde{w}\|^2,$$

where

$$\tilde{w} = -\mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right]^{-1}\mathop{\mathbb{E}}_{\theta}\left[\delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right].$$

Then, we have

$$\begin{aligned}
\frac{\mathrm{d}L_2(w)}{\mathrm{d}t} &= \nabla_w L_2(w)^\top \dot{w} \\
&= (w - \tilde{w})^\top \mathop{\mathbb{E}}_{\theta}\left[-G_t^{\iota,e}\psi_t\psi_t^\top w - \delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right] \\
&= -(w - \tilde{w})^\top \mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right]\left(w + \mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right]^{-1}\mathop{\mathbb{E}}_{\theta}\left[\delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right]\right) \\
&= -(w - \tilde{w})^\top \mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right](w - \tilde{w}) \\
&< 0, \quad \forall w \neq \tilde{w}.
\end{aligned}$$

Therefore, an asymptotically stable equilibrium for the recursion (5.9) is $\tilde{w}$. Now, from the arguments above, Theorem 2.2 in Borkar and Meyn, 2000 holds, and the recursion (5.6) converges to $\tilde{w}$ almost surely. Furthermore, under Assumption 5.7, $G_t^{\iota,e} \to I$ as $t \to \infty$. Therefore, we have

$$\begin{aligned}
\tilde{w} &= -\mathop{\mathbb{E}}_{\theta}\left[G_t^{\iota,e}\psi_t\psi_t^\top\right]^{-1}\mathop{\mathbb{E}}_{\theta}\left[\delta_t^\pi G_t^{\iota,e} e_{\delta,t}\right] \\
&= -\mathop{\mathbb{E}}_{\theta}\left[\psi_t\psi_t^\top\right]^{-1}\mathop{\mathbb{E}}_{\theta}\left[\delta_t^\pi e_{\delta,t}\right] \\
&= -G^{-1}(\theta)\nabla_\theta\eta(\theta) = w^\pi.
\end{aligned}$$

Thus, the recursion (5.6) converges to w^π almost surely. ■

Furthermore, the following corollary holds immediately because under Assumptions 3.1 and 5.1 - 5.7 and Theorem 5.8, Theorem 4 provided by Bhatnagar et al., 2009 holds.

Corollary 5.9. *The parameter of the policy, θ_t, converges to the local optimum almost surely as $t \to \infty$.*

5.3 Stability Analysis

In this section, we analyze the stability of INACs and I2NAC. We mainly analyze the stability of the I2NAC iteration. Note that the argument is also generally applicable to INAC algorithms by simply setting $G_t^{\iota,e} = I$. A similar analysis was conducted by Tamar et al., 2014.

First, the recursion for update (5.4) can be rewritten as follows:

$$w_{t+1} = \left(I - \alpha G_t^{\iota,e} e_{f,t} \psi_t^\top\right) w_t + \alpha \delta_t G_t^{\iota,e} e_{\delta,t}.$$

Let w_0 denote the initial value of the NPG. The estimate of the NPG at time $T+1 \in \mathbb{N}_{\geq 0}$ obtained by I2NAC can be rewritten as

$$w_{T+1} = \prod_{t=0}^{T} A_t w_0 + \sum_{t=0}^{T-1} \prod_{\tau=t+1}^{T} A_\tau b_t + b_T,$$

where

$$A_t = I - \alpha G_t^{\iota,e} e_{f,t} \psi_t^\top, \quad b_t = \alpha \delta_t G_t^{\iota,e} e_{\delta,t}.$$

Thus, the L2 norm of the NPG estimated by I2NAC is upper bounded as follows:

$$\|w_{T+1}\| \leq \prod_{t=0}^{T} \|A_t\| \|w_0\| + \sum_{t=0}^{T-1} \prod_{\tau=t+1}^{T} \|A_\tau\| \|b_t\| + \|b_T\|.$$

When Assumptions 5.1 and 5.2 hold, there is a constant C_b such that $\|b_t\| < C_b \|\psi_t\|$. Therefore, $\|A_t\|$ governs the boundedness of the norm of the estimated NPG; if $\|I - \alpha G_t^{\iota,e} e_{f,t} \psi_t^\top\| \leq 1$ for all t, $\|w_t\|$ stays bounded. Else it may diverge. The

same argument holds for INACs. The following theorem gives $\|I - \alpha e_{f,t}\psi_t^\top\|$ and $\|I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\|$.

Theorem 5.10. *$\|I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\|$ is given as follows:*

$$\begin{aligned}&\|I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\| \\ &= \max\{1, \sqrt{1 + \frac{\alpha^2\kappa_t^2 - 2\alpha\nu_t + \alpha\kappa_t\sqrt{\alpha^2\kappa_t^2 - 4\alpha\nu_t + 4}}{2}}\},\end{aligned} \tag{5.10}$$

where

$$\kappa_t \triangleq \left\|G_t^{\iota,e} e_{f,t}\right\| \|\psi_t\| \quad \textit{and} \quad \nu_t \triangleq \left(G_t^{\iota,e} e_{f,t}\right)^\top \psi_t.$$

Setting $G_t^{\iota,e} = I$ in Eq. (5.10) yields $\|I - \alpha e_{f,t}\psi_t^\top\|$.

In the proof of Theorem 5.10, we use the following lemma (Lemma 2 in (Tamar et al., 2014). See (Tamar et al., 2014) or Appendix A.1 for the proof).

Lemma 5.11. *Let $X = x_1 y_1^\top + x_2 y_2^\top \in \mathbb{R}^{d\times d}$, then the matrix X has $d-2$ eigenvalues that are equal to 0, and the remaining two eigenvalues are given by*

$$\frac{x_1^\top y_1 + x_2^\top y_2 \pm \sqrt{(x_1^\top y_1 - x_2^\top y_2)^2 + 4(x_1^\top y_2)(x_2^\top y_1)}}{2}.$$

Proof of Theorem 5.10. We consider I2NAC. For INAC algorithms, $G_t^{\iota,e} = I$ gives $\|I - \alpha e_{f,t}\psi_t^\top\|$ from the definition. The norm of a real-valued matrix A is given by the square root of the maximum eigenvalue of $A^\top A$. Thus,

$$\begin{aligned}&\left(I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\right)^\top \left(I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\right) \\ &= I - \alpha\psi_t \left(G_t^{\iota,e} e_{f,t}\right)^\top - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top + \alpha^2\psi_t \left(G_t^{\iota,e} e_{f,t}\right)^\top G_t^{\iota,e} e_{f,t}\psi_t^\top \\ &= I + \psi_t \left(\alpha^2 \left\|G_t^{\iota,e} e_{f,t}\right\|^2 \psi_t - \alpha G_t^{\iota,e} e_{f,t}\right)^\top - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top \\ &\triangleq I + X.\end{aligned} \tag{5.11}$$

Note that $G_t^{\iota,e} e_{f,t}$ is a vector. Thus, from Lemma 5.11, the matrix X on the right-hand side of Eq. (5.11) has $d-2$ eigenvalues equal to 0, and the remaining two eigenvalues

are given by

$$\frac{\alpha^2\kappa_t^2 - 2\alpha\nu_t \pm \alpha\kappa_t\sqrt{\alpha^2\kappa_t^2 - 4\alpha\nu_t + 4}}{2},$$

where

$$\kappa_t \triangleq \left\|G_t^{\iota,e} e_{f,t}\right\| \|\psi_t\|, \quad \nu_t \triangleq \left(G_t^{\iota,e} e_{f,t}\right)^\top \psi_t.$$

Therefore, the matrix on the right-hand side of Eq. (5.11) has $d-2$ eigenvalues equal to 1, and the remaining two eigenvalues are given by

$$1 + \frac{\alpha^2\kappa_t^2 - 2\alpha\nu_t \pm \alpha\kappa_t\sqrt{\alpha^2\kappa_t^2 - 4\alpha\nu_t + 4}}{2}.$$

Taking the square root of the above gives $\|I - \alpha G_t^{\iota,e} e_{f,t}\psi_t^\top\|$. ■

Theorem 5.10 enables us to compare the stability of I2NAC with INACs. In particular, by setting $e_{\delta,t} = e_{f,t} = e_t = \psi_t$, the difference between the stabilities of I2NAC and INACs becomes apparent.

Remark 5.12. Setting $e_{\delta,t} = e_{f,t} = e_t = \psi_t$, it holds that

$$G_t^{\iota,e} e_{f,t} = G_t^{\iota,\psi}\psi_t = \frac{1}{1+\iota\|\psi_t\|^2}\psi_t.$$

Then, for INAC algorithms,

$$\begin{aligned}\|A_t\| &= \|I - \alpha\psi_t\psi_t^\top\| \\ &= \max\{1, |\alpha\|\psi_t\|^2 - 1|\} \geq 1, \qquad (5.12)\\ \|b_t\| &= \alpha\|\delta_t\psi_t\|, \qquad (5.13)\end{aligned}$$

and for I2NAC,

$$\begin{aligned}\|A_t\| &= \|I - \alpha G_t^{\iota,e} \psi_t \psi_t^\top\| \\ &= \max\{1, |1 - \frac{\alpha\|\psi_t\|^2}{1+\iota\|\psi_t\|^2}|\} = 1, \qquad (5.14)\\ \|b_t\| &= \frac{\alpha}{1+\iota\|\psi_t\|^2}\|\delta_t\psi_t\|. \qquad (5.15)\end{aligned}$$

The last equality in Eq. (5.14) holds because $\iota \geq \alpha$. Here, we assume that the policy is Gaussian, $\mathcal{N}(\mu, \sigma)$. Then, the eligibility is given by

$$\psi_\mu = (a-\mu)/\sigma^2, \quad \psi_\sigma = ((a-\mu)^2 - \sigma^2)/\sigma^3.$$

In MDP, the optimal policy is deterministic. Thus, if the learning progresses successfully, $\sigma \to 0$ and $\|\psi\| \to \infty$. Therefore, Eqs. (5.12) -(5.15) indicate that the iterations by INACs may diverge regardless of the value of the step-size α, even if the learning successes, while the iteration by I2NAC stays bounded. In other words, I2NAC is guaranteed to be safe.

Note that by applying the decreasing step-size $\alpha \to 0$, $\|A_t\| \to 1$ and $\|b_t\| \to 0$ as $t \to \infty$. However, for INACs, the norm $\|A_t\|$ may take a value larger than 1 in finite t depending on the values of α and ψ_t. Therefore, the product $\Pi_t\|A_t\|$ may become large, and the learning becomes unstable in finite t.

5.4 Numerical Experiment

In Section 5.3, we analyzed the stabilities of I2NAC and INACs, especially for the case where $e_{\delta,t} = e_{f,t} = e_t = \psi_t$. In this section, we empirically evaluate the learning stability and speed for the case where the eligibility trace is used.

5.4.1 Setups

Environment and Parameterization: See Section 3.2.3.

Algorithms: For the estimation of NPG, NTD and I2NAC based on NTD iteration are applied. Thus, the TD error and the eligibility traces are defined as Eqs. (3.57)

and (3.58), respectively. Both constant step-sizes and decreasing step-size schedules are applied. The schedules that we utilized, which satisfy Assumption 5.5, are as follows (Bhatnagar et al., 2009):

$$\alpha_t = \frac{\alpha}{1 + t^{2/3}/\tau}, \quad \alpha_{v,t} = \frac{\alpha_v}{1 + t^{2/3}/\tau} \quad \text{and} \quad \alpha_{\theta,t} = \frac{\alpha_\theta}{1 + t/\tau}, \tag{5.16}$$

where we chose $\tau = 10^6$. For comparison, we again applied AdaGrad (4.22) and Adam (4.23) to the estimation of NPG. In a similar way in Section 3.2.3, the policy parameter is not updated within the first 100 episodes. and the state value function is estimated by TD(λ) (Sutton, 1988).

Grid Search: We performed a same grid search as done in Section 4.3.2. For I2NAC, we chose $\iota = \alpha$.

5.4.2 Results and Discussions

Stabilities: First, we evaluate the learning stability of I2NAC. There are three possible causes that make the learning system divergent: (i) the policy parameter θ_t diverges; (ii) the state value parameter v_t first diverges, resulting in a divergent learning signal δ_t for NPG estimation; and (iii) NPG w_t first diverges, resulting in a divergent learning signal for the policy. Because our grid search does not include too large a value of α_θ, the case (i) above was not observed.

Table 5.1 shows the percentages of the non-divergent sets and the cause of divergences in the grid search. The main causes of divergence for NTD were the divergence of w. I2NAC was far more stable than NTD, and its main causes of divergence were v. For both NTD and I2NAC, the use of decreasing step-sizes (DS) decreases the divergence. NTD with AdaGrad was the least divergent algorithm, and very few divergences were caused by w. Thus, Table 5.1 contains the value 0.0. The divergences of NTD with Adam were also less caused by w, but more caused by v.

TABLE 5.1: Percentages of non-divergent sets and the cause of divergences in the grid search.

	non-divergent	v diverged	w diverged
NTD	19.4%	34.5%	46.1%
NTD with DS	34.1%	24.8%	41.1%
I2NAC	65.5%	29.0%	5.5%
I2NAC with DS	77.9%	16.5%	5.6%
NTD with AdaGrad	80.0%	20.0%	0.0%
NTD with Adam	50.6%	48.8%	0.6%

Performances for Best Meta-Parameter Values: Second, we evaluate the learning speed of I2NAC. Figure 5.1 shows the learning curves for the best meta-parameter values. Each learning curve is an average of 10 independent runs with different random seeds. Table 5.2 shows the best meta-parameter values and the final performances with standard deviations. "Best" indicates that the acquired average rewards were largest and the estimates did not diverge. NTD could improve the policy initially, but the performance of the policy degraded in the later stage of the learning. For both NTD and I2NAC, the application of the DS schedules resulted in the degradation of the learning speed. The learning of AdaGrad was the fastest. The final performance of I2NAC was comparable to that of AdaGrad, and slightly better than that of Adam.

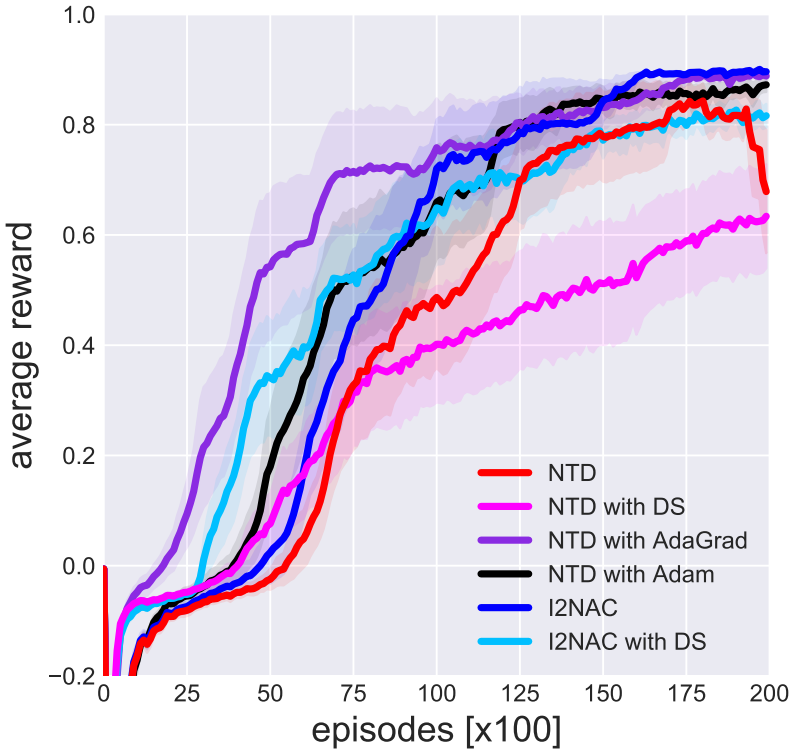


FIGURE 5.1: Learning curves for the best meta-parameter values in the sense that the acquired average reward were largest and the estimates did not diverge. The horizontal axis indicates the training episodes, and the vertical axis indicates the average reward. Each learning curve is a mean of 10 independent runs with different random seeds. The shaded areas indicate standard errors.

TABLE 5.2: Best meta-parameter values found in the grid search. The right-most column shows the mean of the final performances with standard deviations over 10 independent runs.

	λ	α_θ	α	α_v	final performance
NTD	1.0	10^{-6}	$5 \cdot 10^{-4}$	10^{-4}	0.679 ± 0.353
NTD with DS	0.95	10^{-5}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-3}$	0.634 ± 0.298
NTD with AdaGrad	0.9	$5 \cdot 10^{-6}$	10^{-1}	10^{-3}	0.888 ± 0.025
NTD with Adam	0.99	10^{-6}	10^{-3}	$5 \cdot 10^{-4}$	0.872 ± 0.041
I2NAC	1.0	10^{-6}	10^{-3}	10^{-4}	0.896 ± 0.007
I2NAC with DS	1.0	$5 \cdot 10^{-6}$	$5 \cdot 10^{-3}$	10^{-3}	0.816 ± 0.068

Comparison with Adaptive Step-size Strategies: Now, we compare the overall performances of I2NAC with those of AdaGrad and Adam. Table 5.3 shows the percentages of the meta-parameter values in the grid search by which the learning resulted in poor final performances (less than 0; the pendulum keeps rotating) . Table 5.3 indicates that the percentage of AdaGrad that results in the poor final performances are relatively higher than those of I2NAC and Adam.

TABLE 5.3: Percentages of the final performances in the grid search. Only the percentages for poor performance sets and good performance sets are shown. For each set of meta-parameter values, the result is averaged over 10 iterations for different random seeds.

	poor final performances
NTD with AdaGrad	33.9%
NTD with Adam	11.7%
I2NAC	14.0%

As stated above, we also observed that (i) I2NAC was more stable than NTD with Adam, while applying AdaGrad was the least divergent, (ii) for the best meta-parameter values, NTD with AdaGrad exhibited the fastest initial improvement of the policy, and (iii) the final performance of I2NAC was comparable to that of AdaGrad and slightly better than that of Adam. Note that from Eqs. (5.4), (4.22), and (4.23), the adaptive values of the step-sizes at each update for these methods are dependent on the following quantities:

$$\alpha_t^{\mathrm{ADAPT}} \triangleq \begin{cases} \left\| \dfrac{\alpha}{\sqrt{\hat{g}_t} + \epsilon} \right\| & \text{for AdaGrad,} \\ \left\| \dfrac{\alpha}{\sqrt{\hat{g}_{v,t}} + \epsilon} \dfrac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \right\| & \text{for Adam,} \\ \left\| \alpha \left(I - \dfrac{\iota}{1+\iota\|e_t\|^2} e_t e_t^\top \right) \right\| & \text{for I2NAC.} \end{cases} \tag{5.17}$$

From the above expressions, it is apparent that $\alpha_t^{\mathrm{ADAPT}}$ for I2NAC and Adam in Eqs. (5.17) can either increase or decrease during the learning process, while for AdaGrad, $\alpha_t^{\mathrm{ADAPT}}$ is always decreasing. Thus, it is possible that $\alpha_t^{\mathrm{ADAPT}}$ for AdaGrad

in Eqs. (5.17) decreases too fast, and that the learning results in premature convergence. Indeed, for small values of α in the grid search, we observed that $\alpha_t^{\mathrm{ADAPT}}$ for AdaGrad decayed in the early learning stages and the amounts of updates of w, that is, the estimate of NPG, became very small. As a result, the policy did not improve that much. Figure 5.2 provides examples of this phenomena, where the meta-parameter values that are used are the same as those in Table 5.2, with the exception of α. For AdaGrad, $\alpha_t^{\mathrm{ADAPT}}$ decays to a very small value, especially when α is small. Therefore, it appears that the learning of AdaGrad was less divergent, while many more learning trials resulted in premature convergence compared to the case for I2NAC and Adam for small α, as is also indicated by Table 5.3. Even though such a premature convergence can be avoided by the much larger value of α compared to I2NAC and Adam (see Figure 5.1 and Table 5.2), this issue would be much more serious in tasks that require many more training episodes. For Adam, $\alpha_t^{\mathrm{ADAPT}}$ tends to increase, and when α is set to the large value, $\alpha_t^{\mathrm{ADAPT}}$ grows large leading to the divergence of the estimates. Combined with Tables 5.1, 5.3 and Figure 5.2, it appears that the percentage of Adam that results in a poor performance is the lowest among the three methods at the cost of divergence. For I2NAC, although $\alpha_t^{\mathrm{ADAPT}}$ appears to be constant, it in fact keeps changing dramatically over a small range. Furthermore, Table 5.3 and Figure 5.2 indicate that I2NAC is the least sensitive to the value of α.

Note also that the update direction of Adam is determined by $\hat{g}_{m,t}$ and $\hat{g}_{v,t}$. Both of these values are the cumulative of g_t, which already contains the information pertaining to e_t. Thus, the application of Adam to INACs completely breaks the equivalence between the forward and backward views, which is established by the eligibility traces (Sutton, 1988; Morimura, Uchibe, and Doya, 2005). In contrast, as we showed in the previous sections, at least the fixed point of I2NAC is equivalent to the fixed point of the base INAC algorithm. This may be one of the reasons for which the final performance of Adam is slightly worse than that of I2NAC.

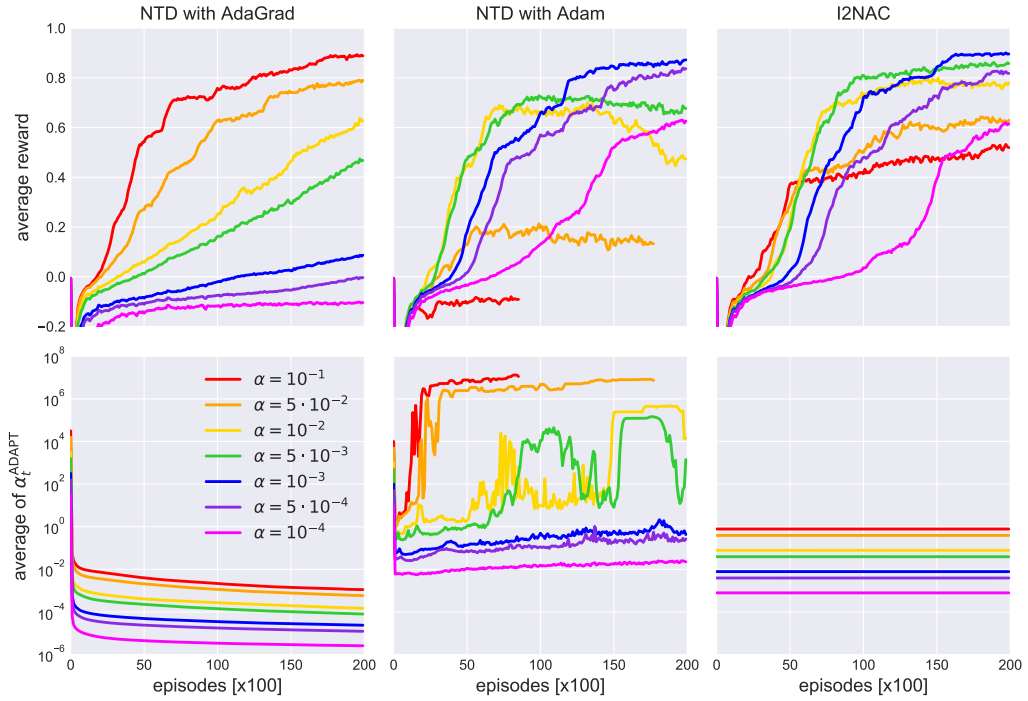


FIGURE 5.2: Learning curves for various values of α. The meta-parameter values were the same as in Table 5.2, except for α. The horizontal axis indicates the training episodes. Top: the vertical axis indicates the average reward. Bottom: the vertical axis indicates the average value of $\alpha_t^{\mathrm{ADAPT}}$ (defined in Eq. (5.17)) in each episode. Each result is a mean of 10 independent runs with different random seeds. For clarity, only the means are shown. When $\alpha = 10^{-1}$ and $\alpha = 5 \cdot 10^{-2}$, the learning curves for NTD with Adam are truncated because the estimates of NPG diverged.

Effect of ι on the learning stability and speed: Finally, we evaluate the effect of implicit rate ι on the learning stability and speed of I2NAC. In order to evaluate the learning stability, we conducted an additional grid search with a fixed $\lambda = 1$ and $\iota \in \{\alpha, 2\alpha, 5\alpha, 10\alpha, 1\}$. Table 5.4 shows the percentages of the non-divergent sets and the cause of divergences in terms of ι. The result indicates that as ι increases, the learning process gradually becomes more stable.

TABLE 5.4: Percentages of the non-divergent sets and the cause of divergences for the various values of ι.

ι	non-divergent	v diverged	w diverged
α	55.5%	4.6%	39.9%
2α	55.6%	4.5%	39.9%
5α	57.3%	4.4%	38.3%
10α	58.4%	4.6%	37.0%
1	61.5%	4.8%	33.7%

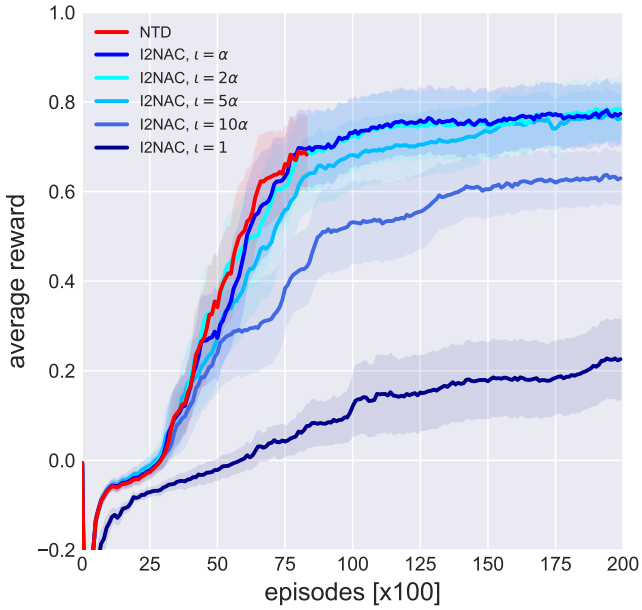
In order to evaluate the learning speed, we considered the following two cases for NTD: (A) the estimate of NPG diverged, or (B) the performance of the policy degraded at the later stage of the learning. Table 5.5 shows the meta-parameters by which the divergence and the degradation occurred. We applied I2NAC with $\iota \in \{\alpha, 2\alpha, 5\alpha, 10\alpha, 1\}$ to both of the above cases. We used identical meta-parameter values and random seeds for I2NAC. Figures 5.3 (A) and 5.3 (B) show the learning results for the divergent and the degradation cases, respectively. Each learning curve is a mean of 10 independent runs with different random seeds. In Figure 5.3 (A), the estimate of NPG by NTD diverged, where the learning curve is truncated. In Figure 5.3 (B), the performance of the policy acquired by NTD degraded at the later stage of the learning, even though the estimates did not diverge. For both cases, I2NAC improved the policy without divergence or degradation. When $\iota = \alpha$ and $\iota = 2\alpha$, the learning speed decreased slightly. However, as ι became large, the learning speed decreased substantially. These results indicate that I2NAC can improve the stability at a small cost of the learning speed when the value of ι is small. Although the performance of I2NAC was less sensitive to the value of ι compared to other meta-parameters such as the step-size, these results indicate that $\iota = \alpha$ is a good choice for the implicit rate.

TABLE 5.5: Meta-parameter values used to evaluate the effect of ι.

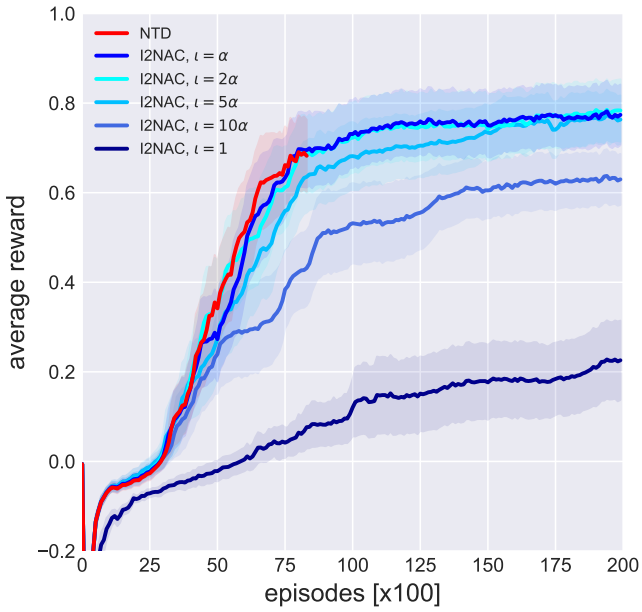
case	λ	α_θ	α	α_v
divergence	0.9	$5.0 \cdot 10^{-6}$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$
degradation	1.0	$1.0 \cdot 10^{-6}$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$

5.5 Closing Remarks

In this chapter, we proposed I2NAC, which is an incremental estimation algorithm of the NPG based on the implicit update. Existing INAC algorithms can be generally extended to I2NAC. The convergence of I2NAC iteration was proven under conventional and mild assumptions. Theoretical analysis results showed why the existing INAC methods are unstable, and how the use of I2NAC overcomes the instability and results in the safe incremental estimation of NPG. Using the classical benchmark task, it was shown that I2NAC is less sensitive to the values of the step-sizes. While the adaptive step-size methods for policy (Matsubara, Morimura, and Morimoto, 2010; Pirotta, Restelli, and Bascetta, 2013) or state value function (Dabney and Barto, 2012) would further stabilize the whole learning process, this issue is outside of the scope of this study.



(A) w diverged for NTD.



(B) Policy degraded for NTD.

FIGURE 5.3: Effect of ι on the learning speed. The horizontal axis indicates the training episodes, and the vertical axis indicates the average reward. With the exception of ι for I2NAC and random seeds, the same values of the meta-parameters (shown in Table 5.5) were used. Each learning curve is a mean of 10 independent runs with different random seeds. The shaded areas indicate the standard errors. (A) the estimate of NPG by NTD diverged, where the learning curve is truncated. (B) the performance of the policy acquired by NTD degraded at the later stage of the learning, even though the estimates did not diverge.

Chapter 6

Conclusion and Future Work

In this dissertation, we proposed two methods which aim at safe estimation of the NPG. Our main contributions are summarized as follows:

- In Chapter 4, we proposed an adaptive step-size strategy for INAC methods. Though the analysis and proposal in this chapter were focused on NTD algorithm, they can be easily extended for any other INAC methods. We first derived an upper bound of the step-size for local update, so that each update of the parameter vector does not overshoot the given target signal. We also provided the tight upper and lower bounds. Then, we proposed an adaptive step-size strategy to implement the derived upper bound, which takes into account the global effect. The adaptive step-size was derived by considering the infinite-times-update with infinitesimal step-size. The proposed adaptive step-size strategy is guaranteed that it does not exceed the derived upper bound. We evaluated the usefulness of the proposed method in classical benchmark problems.

- In Chapter 5, we proposed a new incremental and stable algorithm for the NPG estimation based on the idea of the implicit update of the parameter vector. We named the proposed algorithm the *implicit incremental natural actor critic* (I2NAC), Existing INAC methods can be generally extended to I2NAC. We provided the asymptotic convergence analysis of I2NAC. Then, we compared the learning stability of the I2NAC and INACs by calculating the upper bound of the norm of the estimated NPGs. Theoretical analysis results showed the stability of I2NAC and that the conventional INACs have a element of danger

which leads the estimated NPG to divergence even when the learning procedure successes. We evaluated the usefulness of the proposed method in classical benchmark and confirmed its usefulness. The experimental results showed that I2NAC is less sensitive to the values of the meta-parameters, including the step-size for the NPG update, compared to the existing incremental NPG method. Though I2NAC has an additional meta-parameter ι, it was shown that the learning performance is less sensitive to ι compared to other meta-parameters. Importantly, it was suggested that even in the meta-parameter settings where the conventional incremental NPG method resulted in the policy degradation or the divergence of the estimated parameter, I2NAC could still improve the policy.

6.1 Future Work

Although we proposed two algorithms to assure the safety of incremental NPG estimation, recall that our notion of **afety** has only restricted meaning and there are many open problems.

Off-policy Learning

As described in §1.2.4 Off-policy learning is an important element of deep reinforcement learning methods (Mnih et al., 2015; Lillicrap et al., 2016; Gu et al., 2017b; Gu et al., 2017a; Haarnoja et al., 2017; Haarnoja et al., 2018). The proposed methods in this dissertation are *on-policy* methods, mainly because the base INAC algorithms are on-policy. Extending the proposed methods to off-policy policy gradient algorithms (Degris, White, and Sutton, 2012; Silver et al., 2014) is a promising future direction.

Monotonicity of Performance Improvement

TRPO proposed by Schulman et al., 2015 is based on a theoretical analysis in which they derived a lower bound for the performance difference of two policies. The derived bound indicates that by constraining the update of the policy with respect to the KL divergence between the policies before and after update, it is theoretically

possible to acquire a sequence of policies with non-decreasing performances. Similar analyses were performed by Kakade and Langford, 2002; Pirotta et al., 2013. Extending their analyses to compatible function approximation and combining proposed methods will result in safe algorithms in two sense; the estimate of the natural policy gradient is safely obtained without diverging and the policy is safely updated with monotonic improvement. This future direction is worth to investigate.

Extension to Supervised and Unsupervised Learning Methods

As we have already stated in §3.1.2, Thomas, Dann, and Brunskill, 2018 generalized Proposition 3.7 to the broad class of gradient methods and gradient-like learning methods such as TD learning. This generalization is not limited to reinforcement learning methods, but applicable to supervised and unsupervised learning methods. Their proposal is named as a *naturalization* of learning rules. By using their naturalization techniques, it is possible to estimate the natural gradients for general gradient-like learning rules requiring only $\mathcal{O}(d)$ computational complexity and memory. An important future work is to extend the proposed methods in this dissertation so that it is possible to apply them to the naturalization learning method. This extension will enable the safe and incremental estimation of natural gradient for general gradient-like learning rules.

Appendix A

Mathematical Background

In this chapter, mathematical background we use is summarized.

A.1 Linear Algebra

This section provides background on linear algebra. See (Meyer, 2000) for complete explanation. Notice that throughout this dissertation, all the vectors are column vectors.

We use the following property for a matrix defined by the outer product of vectors:

Property A.1. *Let $n \leq d$ and let $x_i, y_i \in \mathbb{R}^d, i \in \{1, \ldots, n\}$. The rank of the following matrix is n:*

$$A = \sum_{i=1}^{n} x_i y_i^\top.$$

Definition A.2 (Trace of Matrix)**.** Let a_{ij} denote the element in i-th row and j-th column of a matrix $A \in \mathbb{R}^{d \times d}$. A *trace* of matrix A, $\mathrm{tr}(A)$, is defined as follows:

$$\mathrm{tr}(A) = \sum_{i=1}^{d} a_{ii}.$$

We use the following properties of the trace of matrix.

Property A.3. *The trace of matrix has the following properties:*

1. *Let $A, B \in \mathbb{R}^{d \times d}$. Then it holds that*

$$\mathrm{tr}(A + B) = \mathrm{tr}(A) + \mathrm{tr}(B). \tag{A.1a}$$

2. *Let $\lambda_i, i \in \{1, \ldots, d\}$ denote the eigenvalues of a matrix $A \in \mathbb{R}^{d \times d}$. Then it holds that*

$$\mathrm{tr}(A) = \sum_{i=1}^{d} \lambda_i. \tag{A.1b}$$

More generally, letting $k \in \mathbb{N}_0$, it holds that

$$\mathrm{tr}(A^k) = \sum_{i=1}^{d} \lambda_i^k. \tag{A.1c}$$

3. *Let $xy^\top \in \mathbb{R}^{d \times d}$, where $x, y \in \mathbb{R}^d$. Then it holds that*

$$\mathrm{tr}(xy^\top) = x^\top y. \tag{A.1d}$$

Sherman-Morrison

Let $A \in \mathbb{R}^{d \times d}$ be a non-singular matrix and $b, c \in \mathbb{R}^d$ be vectors such that $1 + c^\top A^{-1} b \neq 0$. Then, the sum $A + bc^\top$ is non-singular and it holds that

$$(A + bc^\top)^{-1} = A^{-1} - \frac{A^{-1} bc^\top A^{-1}}{1 + c^\top A^{-1} b}. \tag{A.2}$$

The equation above is called *Sherman-Morrison formula*.

Matrix Norm

There are several definitions of matrix norms. In this dissertation, we used a matrix norm induced by the euclidean vector norm. Let $\|x\|$ denote the euclidean norm for a vector x.

Definition A.4 (Matrix Norm Induced by the Euclidean Norm)**.** Let $A \in \mathbb{R}^{l \times d}$ and $x \in \mathbb{R}^d$. The norm of A is given by

$$\|A\| = \max_{\|x\|=1} \|Ax\|. \tag{A.3}$$

Property A.5. *Let $A \in \mathbb{R}^{l\times d}$. Let $\lambda_{\max}$ be the largest eigenvalue of $A^\top A$. Then, it holds that*

$$\|A\| = \sqrt{\lambda_{\max}}. \tag{A.4}$$

The following lemma (Lemma 2 in (Tamar et al., 2014)) gives the norm of a matrix defined by $X = x_1 y_1^\top + x_2 y_2^\top \in \mathbb{R}^{d\times d}$, where $x_1, y_1, x_2, y_2 \in \mathbb{R}^d$.

Lemma A.6. *The matrix $X = x_1 y_1^\top + x_2 y_2^\top \in \mathbb{R}^{d\times d}$ has $d-2$ eigenvalues that are equal to 0, and the remaining two eigenvalues are given by*

$$\frac{x_1^\top y_1 + x_2^\top y_2 \pm \sqrt{(x_1^\top y_1 - x_2^\top y_2)^2 + 4(x_1^\top y_2)(x_2^\top y_1)}}{2}.$$

Proof. The rank of X is 2 from Property A.1. We calculate the remaining two eigenvalues, λ_1, λ_2. Using Property A.3, we have:

$$\begin{aligned}\lambda_1 + \lambda_2 &= \mathrm{tr}(X)\\ &= \mathrm{tr}(x_1 y_1^\top) + \mathrm{tr}(x_2 y_2^\top)\\ &= x_1^\top y_1 + x_2^\top y_2.\end{aligned} \tag{A.5}$$

Similarly, it holds that

$$\begin{aligned}\lambda_1^2 + \lambda_2^2 &= \mathrm{tr}(X^2)\\ &= \mathrm{tr}(x_1 y_1^\top x_1 y_1^\top) + \mathrm{tr}(2 x_1 y_1^\top x_2 y_2^\top) + \mathrm{tr}(x_2 y_2^\top x_2 y_2^\top)\\ &= (x_1^\top y_1)^2 + 2(x_1^\top y_2)(x_2^\top y_1) + (x_2^\top y_2)^2.\end{aligned}$$

Thus, we have

$$\begin{aligned}\lambda_1\lambda_2 &= \left((\lambda_1+\lambda_2)^2 - (\lambda_1^2+\lambda_2^2)\right)/2\\ &= (x_1^\top y_1)(x_2^\top y_2) - (x_1^\top y_2)(x_2^\top y_1).\end{aligned} \tag{A.6}$$

Combining (A.5) and (A.6) yields

$$\lambda_i^2 - \left(x_1^\top y_1 + x_2^\top y_2\right)\lambda_i + (x_1^\top y_1)(x_2^\top y_2) - (x_1^\top y_2)(x_2^\top y_1) = 0, \quad i \in \{1,2\}.$$

Solving above equation completes the proof. ■

A.2 Stochastic Approximation

The purpose of this section is to present a *two-timescale stochastic approximation*, which is a theoretical fundamental of the incremental natural actor critic algorithms.

Definition A.7 (Almost Sure Convergence)**.** A sequence of random variables $\{X_n\}$ converges *almost surely* to X if

$$\mathbb{P}\left(\lim_{n\to\infty} X_n = X\right) = 1. \tag{A.7}$$

Two-timescale Approach

Now, we proceed to the two-timescale stochastic approximation. Let $X_t \in \mathbb{R}^d, Y_t \in \mathbb{R}^l, t \in \mathbb{N}_0$ be two sequences of parameters governed by the following difference equations:

$$X_{t+1} = X_t + \alpha_t \left(f(X_t, Y_t) + M_{X,t+1}\right), \tag{A.8}$$

$$Y_{t+1} = Y_t + \beta_t \left(g(X_t, Y_t) + M_{Y,t+1}\right), \tag{A.9}$$

where $f : \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^d$ and $g : \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^l$ are Lipschitz continuous functions and $M_{X,t+1}$ and $M_{Y,t+1}$ are martingale difference sequences w.r.t. the σ-algebras $\mathcal{F}_t = \sigma(X_\tau, Y_\tau, M_{X,\tau}, M_{Y,\tau}, \tau \le t)$. $\mathcal{F}_t$ is a filtration which satisfy $\mathcal{F}_{t_1} \subset \mathcal{F}_{t_2}$ whenever $t_1 \le t_2$ and

$$\mathbb{E}\left[\|M_{z,t+1}\|^2 \middle| \mathcal{F}_t\right] \le C(1 + \|X_t\|^2 + \|Y_t\|^2), \quad z \in \{X, Y\}, t \ge 0, \tag{A.10}$$

for some constant $C < \infty$. We require the following assumptions for the two step-size schedules $\{\alpha_t\}$ and $\{\beta_t\}$.

Assumption A.8. *The step-size schedules* $\{\alpha_t\}$ *and* $\{\beta_t\}$ *satisfy the following relations:*

$$\sum_t \alpha_t = \sum_t \beta_t = \infty, \quad \sum_t \alpha_t^2 = \sum_t \beta_t^2 < \infty, \tag{A.11}$$

$$\beta_t = o(\alpha_t). \tag{A.12}$$

Under the condition (A.12), $\beta_t \to 0$ faster than $\{\alpha_t\}$. Therefore, the iteration (A.9) proceeds at a "slower rate" than (A.8). Intuitively, the faster component $X(t)$ sees the slower component $Y(t)$ as quasi-static. Now, consider the following coupled ODEs:

$$\dot{X} = f(X(t), Y(t)), \tag{A.13}$$

$$\dot{Y} = 0, \tag{A.14}$$

or

$$\dot{X} = f(X(t), Y), \tag{A.15}$$

where Y is a constant because of (A.14).

Suppose that following assumptions hold.

Assumption A.9. $\sup \|X_t\| < \infty, \sup \|Y_t\| < \infty$.

Assumption A.10. *The ODE* (A.15) *has a globally asymptotically stable equilibrium* $\mu(Y)$ *where* $\mu : \mathbb{R}^l \to \mathbb{R}^d$ *is a Lipschitz continuous function.*

Assumption A.11. *The following ODE*

$$\dot{Y} = g(\mu(Y(t)), Y(t)). \tag{A.16}$$

has a globally asymptotically stable equilibrium Y^*.

Let $\{\bar{\alpha}_t\}$ and $\{\bar{\beta}_t\}$ be two real-valued sequences such that

$$\bar{\alpha}_t = \sum_{\tau=0}^{t-1} \alpha_\tau \quad \text{and} \quad \bar{\beta}_t = \sum_{\tau=0}^{t-1} \beta_\tau,$$

respectively. Note that (A.11) implies $(\bar{\alpha}_t - \bar{\alpha}_{t-1}) \to 0$ and $(\bar{\beta}_t - \bar{\beta}_{t-1}) \to 0$ as $t \to \infty$. Let $\bar{X}(\bar{\alpha})$ and $\bar{Y}(\bar{\alpha})$ denote continuous time processes such that $\bar{X}(\bar{\alpha}_t) = X_t$ and $\bar{Y}(\bar{\alpha}_t) = Y_t$, respectively, with $\bar{\alpha} \in \mathbb{R}_{\geq 0}$ and linear interpolations in between. For $\bar{\beta} \geq 0$, let $X^{\bar{\beta}}(\bar{\alpha})$ and $Y^{\bar{\beta}}(\bar{\alpha})$ be the solutions of (A.13) and (A.14), respectively, with $X^{\bar{\beta}}(\bar{\beta}) = \bar{X}(\bar{\beta})$ and $Y^{\bar{\beta}}(\bar{\beta}) = \bar{Y}(\bar{\beta})$, where $\bar{\alpha} \geq \bar{\beta}$. Note that (A.14) implies $Y^{\bar{\beta}}(\bar{\alpha}) = \bar{Y}(\bar{\beta})$ where $\bar{\alpha} \geq \bar{\beta}$.

Now, one can write (A.9) as

$$Y_{t+1} = Y_t + \alpha_t \left(\frac{\beta_t}{\alpha_t} \left(g(X, Y) + M_{Y,t+1} \right) \right) \tag{A.17}$$

Notice that (A.12) implies the term multiplying α_t on the r.h.s above vanishes in the limit. Therefore, (A.8) and (A.9) can be considered as Euler discretizations of the ODEs (A.13) and (A.14) with noise terms $M_{X,t+1}$ and $M_{Y,t+1}$ and time discretization $\{\bar{\alpha}_t\}$. Borkar, 1997 showed that for any given $T > 0$,

$$\sup_{\bar{\alpha} \in [\bar{\beta}, \bar{\beta}+T]} \|\bar{X}(\bar{\alpha}) - X^{\bar{\beta}}(\bar{\alpha})\| \to 0 \quad \text{and} \quad \sup_{\bar{\alpha} \in [\bar{\beta}, \bar{\beta}+T]} \|\bar{Y}(\bar{\alpha}) - Y^{\bar{\beta}}(\bar{\alpha})\| \to 0$$

almost surely as $\bar{\beta} \to \infty$. Similar argument holds for the iteration (A.9) with the ODE (A.16) and the time discretization $\{\bar{\beta}_t\}$.

The following proposition establishes the convergence of two-timescale stochastic approximations:

Proposition A.12 (Borkar, 1997, Theorem 1.1)**.** *Under Assumptions A.8 - A.11, $(X_t, Y_t) \to (\mu(Y^*), Y^*)$ as $t \to \infty$ almost surely.*

List of Symbols

$\mathbb{N}$	natural numbers
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{R}$	real numbers
$\mathbb{R}_{\geq 0}$	non-negative real numbers
$\mathbb{Q}_{\geq 0}$	non-negative rational numbers
$\mathbb{P}$	probability
$\mathbb{E}$	expectation
$\mathbb{E}_p$	expectation with respect to a distribution p
$\mathcal{S}$	finite state space
$\mathcal{A}$	finite action space
$\mathcal{P}$	state transition probability, state transition law or dynamics
$\mathcal{R}$	reward function
s	state
a	action
r	reward
R	return
ξ	history, trajectory or sample path
t, τ	time steps
η	learning objective function
π	policy
π^*	optimal policy

ρ_0	initial state distribution
ρ^π	future state distribution under the policy π
γ	discount factor
V^π	state value function under the policy π
Q^π	action value function under the policy π
A^π	advantage function under the policy π
V^*, Q^*	optimal state value and action value
$\mathcal{T}^\pi$	Bellman operator under policy π
$\mathcal{T}$	Bellman optimality operator
δ	temporal difference error
e	eligibility trace
λ	decay factor of eligibility trace
θ	parameter of policy
$\pi(\cdot\|\cdot;\theta), \pi_\theta$	parameterized policy
w	parameter of compatible function approximator for advantage
$f(\cdot,\cdot;w)$	compatible function approximator for advantage
ψ	basis function for compatible function approximator
v	parameter of state value function
$V(\cdot;v)$	parameterized state value function
α	step-size
$G(\theta)$	Riemannian metric

$F(\theta)$	Fisher information matrix
D_{KL}	Kullback-Leibler divergence
h	relative importance
ς	scaling factor
ι	implicit rate

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Professor Minoru Asada. He not only has been my supervisor since my undergraduate, but also motivated me to be a researcher when I was a high school student. Though my current research field is the reinforcement learning, the idea of the cognitive developmental robotics has always been attracting my interest since that day. Furthermore, he provided me a freedom of research throughout my days in the emergent robotics laboratory.

I acknowledge my thesis committee, Professor Koh Hosoda and Professor Koichi Osuka. Their insightful comments and hard questions helped me to deepen my thinking and gave me a chance to reaffirm my stance on the research.

I greatly thank Dr. Hiroki Yokoyama for inspiring ideas, many discussions, and other efforts to help me. He was my advisor in the last half of my master course. Most importantly, he gave me a chance to conduct the research on the reinforcement learning field, and made me realize how interesting to work on theoretical researches and mathematics.

My sincere thanks also goes to Dr. Eiji Uchibe. He readily accepted to review my master thesis and thenceforth he has been the most familiar researcher to me in the reinforcement learning field. He gave me the opportunity to enter the community of reinforcement learning researchers.

I also thank the members and alumni of the emergent robotics laboratory, especially for Dr. Jimmy Baraglia, Dr. Yuji Kawai, Dr. Takato Horii, Mr. Jihoon Park, Dr. Takumi Kawasetsu and Dr. Hisashi Ishihara.

Last, but not least, I thank my family for their love and support.

Bibliography

Abbasi-Yadkori, Yasin, Peter L. Bartlett, and Stephen J. Wright (2016). "A Fast and Reliable Policy Improvement Algorithm". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 1338–1346.

Abbeel, Pieter and Andrew Y. Ng (2005). "Exploration and Apprenticeship Learning in Reinforcement Learning". In: *Proceedings of the 22nd international conference on Machine learning*, pp. 1–8.

Abbeel, Pieter et al. (2007). "An Application of Reinforcement Learning to Aerobatic Helicopter Flight". In: *Advances in Neural Information Processing Systems 19*, pp. 1–8.

Abe, Naoki et al. (2010). "Optimizing debt collections using constrained reinforcement learning". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 75–84.

Amari, Shun-ichi (1998). "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2, pp. 251–276.

Amari, Shun-ichi, Hyeyoung Park, and Kenji Fukumizu (2000). "Adaptive method of realizing natural gradient learning for multilayer perceptrons". In: *Neural Computation* 12.2, pp. 1399–1409.

Asada, Minoru et al. (1996). "Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning". In: *Machine Learning* 23, pp. 279–303.

Bagnell, J. Andrew and Jeff Schneider (2003). "Covariant policy search". In: *Proceedings of 18th the International Joint Conference on Artificial Intelligence*, pp. 1019–1024.

Baird, Leemon C. (1993). *Advantage Updating*. Tech. rep. WL-TR-93-1146. Wright-Patterson Air Force Base Ohio: Wright Laboratory.

Baxter, Jonathan and Peter L. Bartlett (2001). "Infinite-Horizon Policy-Gradient Estimation". In: *Journal of Artificial Intelligence Research* 15, pp. 319–350.

Bertsekas, Dimitri P. (2010). "Pathologies of Temporal Difference Methods in Approximate Dynamic Programming". In: *49th IEEE Conference on Decision and Control*, pp. 3034–3039.

— (2011). "Approximate policy iteration: A survey and some new methods". In: *Journal of Control Theory and Applications* 9.3.

Bertsekas, Dimitri P. and John N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Athena Scientific.

— (2000). "Gradient convergence in gradient methods with errors". In: *SIAM Journal on Optimization* 10.3, pp. 627–642.

Bhatnagar, S. et al. (2009). "Natural Actor-Critic Algorithms". In: *Automatica* 45.11, pp. 2471–2482.

Borkar, Vivek S. (1997). "Stochastic approximation with two time scales". In: *Systems & Control Letters* 29, pp. 291–294.

— (2001). "A sensitivity formula for risk-sensitive cost and the actor–critic algorithm". In: *Systems & Control Letters* 44.2, pp. 339–346.

Borkar, Vivek S. and Sean P. Meyn (2000). "The O.D.E. method for convergence of stochastic approximation and reinforcement learning". In: *SIAM Journal of Control and Optimization* 38.2, pp. 447–469.

Boyan, Justin A. (1999). "Least-squares temporal difference learning". In: *Proceedings of The 16th International Conference on Machine Learning*, pp. 49–56.

Bradtke, Steven J. and Andrew G. Barto (1996). "Linear least-squares algorithms for temporal difference learning". In: *Machine Learning* 22, pp. 33–57.

Chou, Po-Wei, Daniel Maturana, and Sebastian Scherer (2017). "Improving Stochastic Policy Gradients in Continuous Control with Deep Reinforcement Learning using the Beta Distribution". In: *Proceedings of The 34th International Conference on Machine Learning*, pp. 834–843.

Clouse, Jeffery A. and Paul E. Utgoff (1992). "A teaching method for reinforcement learning". In: *Machine Learning Proceedings*, pp. 92–101.

Dabney, William and Andrew G. Barto (2012). "Adaptive Step-Size for Online Temporal Difference Learning". In: *Proceedings of the 26th Conference on Artificial Intelligence (AAAI-12)*, pp. 872–878.

Degris, Thomas, Patrick M. Pilarski, and Richard S. Sutton (2012). "Model-free reinforcement learning with continuous action in practice". In: *Proceedings of the 2012 American Control Conference*.

Degris, Thomas, Martha White, and Richard S. Sutton (2012). "Off-policy actor-critic". In: *Proceedings of The 29th International Conference on Machine Learning*, pp. 457–464.

Doya, Kenji (2000). "Reinforcement Learning in Continuous Time and Space". In: *Neural Computation* 12.1, pp. 219–245.

Duan, Yan et al. (2016). "Benchmarking Deep Reinforcement Learning for Continuous Control". In: *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1329–1338.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization". In: *The Journal of Machine Learning Research* 12, pp. 2121–2159.

Gao, Yuanxiang, Li Chen, and Baochun Li (2018). "Post: Device Placement with Cross-Entropy Minimization and Proximal Policy Optimization". In: *Advances in Neural Information Processing Systems 31*, pp. 9992–10001.

García, Javier and Fernando Fernández (2015). "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16, pp. 1437–1480.

Gehring, Clement and Doina Precup (2013). "Smart Exploration in Reinforcement Learning using Absolute Temporal Difference Errors". In: *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1037–1044.

Geibel, Peter (2006). "Reinforcement learning for mdps with constraints". In: *Proceedings of the 17th European Conference on Machine Learning*, pp. 646–653.

Gordon, Geoffrey J. (2000). "Reinforcement learning with function approximation converges to a region". In: *Advances in Neural Information Processing Systems 13*, pp. 1040–1046.

Gosavi, Abhijit (2009). "Reinforcement learning for model building and variance-penalized control". In: *Proceedings of the Winter Simulation Conference*, pp. 373–379.

Greensmith, Evan, Peter L. Bartlett, and Jonathan Baxter (2004). "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning". In: *Journal of Machine Learning Research* 5, pp. 1471–1530.

Gu, Shixiang et al. (2017a). "Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning". In: *Advances in Neural Information Processing Systems 30*, pp. 3846–3855.

Gu, Shixiang et al. (2017b). "Q-prop: Sample-efficient policy gradient with an off-policy critic". In: *The 5th International Conference on Learning Representations*.

Gullapalli, Vijaykumar (1990). "A stochastic reinforcement learning algorithm for learning real-valued functions". In: *Neural Networks* 3.6, pp. 671–692.

Haarnoja, Tuomas et al. (2017). "Reinforcement Learning with Deep Energy-Based Policies". In: *Proceedings of The 34th International Conference on Machine Learning*, pp. 1352–1361.

Haarnoja, Tuomas et al. (2018). "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of The 35th International Conference on Machine Learning*, pp. 1861–1870.

Harutyunyan, Anna et al. (2016). "Q(λ) with Off-Policy Corrections". In: *Proceedings of The 27th International Conference on Algorithmic Learning Theory*, pp. 305–320.

Heger, Matthias (1994). "Consideration of risk in reinforcement learning". In: *Proceedings of The 11th International Conference on Machine Learning*, pp. 105–111.

Howard, Ronald A. (1960). *Dynamic Programming and Markov Processes*. MIT Press.

Howard, Ronald A. and James E. Matheson (1972). "Risk-sensitive Markov decision processes". In: *Management Science*. Vol. 18. 7, pp. 356–369.

Hutter, Marcus and Shane Legg (2007). "Temporal difference updating without a learning rate". In: *Advances in Neural Information Processing Systems 20*, pp. 705–712.

Imani, Ehsan, Eric Graves, and Martha White (2018). "An Off-policy Policy Gradient Theorem Using Emphatic Weightings". In: *Advances in Neural Information Processing Systems 31*, pp. 96–106.

Kaelbling, Leslie Pack, Michael L. Littman, and Andrew W. Moore (1996). "Reinforcement Learning: A Survey". In: *Journal of Artificial Intelligence Research* 4, pp. 319–350.

Kakade, Sham (2001). "A natural policy gradient". In: *Advances in Neural Information Processing Systems 14*, pp. 227–242.

Kakade, Sham and John Langford (2002). "Approximately optimal approximate reinforcement learning". In: *Proceedings of The 19th International Conference on Machine Learning*.

Kaliszyk, Cezary et al. (2018). "Reinforcement Learning of Theorem Proving". In: *Advances in Neural Information Processing Systems 31*, pp. 8835–8846.

Karampatziakis, Nikos and John Langford (2011). "Online Importance Weight Aware Updates". In: *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pp. 392–399.

Kimura, Hajime and Shigenobu Kobayashi (1998). "An Analysis of Actor/Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Function". In: *Proceedings of The 15th International Conference on Machine Learning*, pp. 278–286.

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *International Conference for Learning Representations*.

Konda, Vijay R. and John N. Tsitsiklis (2003). "On actor critic algorithms". In: *SIAM Journal of Control and Optimization* 42.4, pp. 1143–1166.

Konidaris, George, Sarah Osentoski, and Philip Thomas (2011). "Value function approximation in reinforcement learning using the Fourier basis". In: *Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11)*, pp. 380–385.

Lillicrap, Timothy P. et al. (2016). "Continuous control with deep reinforcement learning". In: *The 4th International Conference on Learning Representations*.

Littman, Michael L., Thomas L. Dean, and Leslie P. Kaelbling (1995). "On the complexity of solving Markov decision problems". In: *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 394–402.

MacKay, David J.C. (1996). *Maximum Likelihood and Covariant Algorithms for Independent Component Analysis*. Tech. rep. University of Cambridge.

Maei, Hamid Reza et al. (2009). "Convergent Temporal-Difference Learning with Arbitrary Smooth Function Approximation". In: *Advances in Neural Information Processing Systems 22*, pp. 1204–1212.

Maei, Hamid Reza et al. (2010). "Toward off-policy learning control with function approximation". In: *Proceedings of The 27th International Conference on Machine Learning*, pp. 719–726.

Maire, Frederic (2005). "Apprenticeship learning for initial value functions in reinforcement learning". In: *Proceedings of the IJCAI'05 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*, pp. 23–28.

Mannor, Shie, Reuven Rubinstein, and Yohai Gat (2003). "The Cross Entropy method for Fast Policy Search". In: *Proceedings of the 20th International Conference on Machine Learning*, pp. 512–519.

Martens, James and Roger Grosse (2015). "Optimizing Neural Networks with Kronecker-factored Approximate Curvature". In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2408–2417.

Matsubara, Takamitsu, Tetsuro Morimura, and Jun Morimoto (2010). "Adaptive step-size policy gradients with average reward metric". In: *Proceedings of 2nd Asian Conference on Machine Learning*, pp. 285–298.

Meyer, Carl D. (2000). *Matrix Analysis and Applied Linear Algebra*. Siam.

Mihatsch, Oliver and Ralph Neuneier (2002). "Risk-Sensitive Reinforcement Learning". In: *Machine Learning* 49.2-3, pp. 267–290.

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–533.

Moriarty, David E., Alan C. Schultz, and John J. Grefenstette (1999). "Evolutionary algorithms for reinforcement learning". In: *Journal of Artificial Intelligence Research* 11, pp. 241–276.

Morimura, Tetsuro, Eiji Uchibe, and Kenji Doya (2005). "Utilizing natural gradient in temporal difference reinforcement learning with eligibility traces". In: *Proceedings of the 2nd International Symposium on Information Geometry and Its Applications*, pp. 256–263.

Morimura, Tetsuro et al. (2010a). "Derivatives of logarithmic stationary distributions for policy gradient reinforcement learning". In: *Neural Computation* 22.2, pp. 342–376.

Morimura, Tetsuro et al. (2010b). "Nonparametric return distribution approximation for reinforcement learning". In: *Proceedings of The 27th International Conference on Machine Learning*, pp. 799–806.

— (2010c). "Parametric return density estimation for reinforcement learning". In: *Proceedings of The 26th conference on Uncertainty in Artificial Intelligence*, pp. 8–11.

Munos, Rémi et al. (2016). "Safe and efficient off-policy reinforcement learning". In: *Advances in Neural Information Processing Systems 29*, pp. 1054–1062.

Nachum, Ofir et al. (2017a). "Bridging the Gap Between Value and Policy Based Reinforcement Learning". In: *Advances in Neural Information Processing Systems 30*, pp. 2775–2785.

— (2017b). "Trust-PCL: An Off-Policy Trust Region Method for Continuous Control". In: *The 5th International Conference on Learning Representations*.

Ng, Andrew Y. et al. (2003). "Autonomous helicopter flight via reinforcement learning". In: *Advances in Neural Information Processing Systems 16*, pp. 799–806.

OpenAI (2018). "Learning Dexterous In-Hand Manipulation". In: *arXiv*. Vol. 1808.00177.

Park, Hyeyoung, Shun-ichi Amari, and Kenji Fukumizu (2000). "Adaptive natural gradient learning algorithms for various stochastic models". In: *Neural Networks* 13, pp. 755–764.

Perkins, Theodore J. and Andrew G. Barto (2001). "Lyapunov-Constrained Action Sets for Reinforcement Learning". In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 409–416.

— (2002). "Lyapunov Design for Safe Reinforcement Learning". In: *Journal of Machine Learning Research* 3.

Peters, Jan and Stefan Schaal (2005). "Natural actor-critic". In: *European Conference on Machine Learning*, pp. 280–291.

— (2006). "Policy gradient methods for robotics". In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 2219–2225.

— (2008a). "Natural actor-critic". In: *Neurocomputing* 71.7, pp. 1180–1190.

— (2008b). "Reinforcement learning of motor skills with policy gradients". In: *Neural Networks* 21.4, pp. 682–697.

Peters, Jan, Sethu Vijayakumar, and Stefan Schaal (2003). "Reinforcement learning for humanoid robotics". In: *Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots*, pp. 103–123.

Pirotta, Matteo, Marcello Restelli, and Luca Bascetta (2013). "Adaptive step-size for policy gradient methods". In: *Advances in Neural Information Processing Systems*, pp. 1394–1402.

Pirotta, Matteo et al. (2013). "Safe Policy Iteration". In: *Proceedings of the 30th International Conference on Machine Learning*, pp. 307–315.

Precup, Doina, Richard S. Sutton, and Sanjoy Dasgupta (2001). "Off-policy temporal-difference learning with function approximation". In: *Proceedings of The 18th International Conference on Machine Learning*, pp. 417–424.

Precup, Doina, Richard S. Sutton, and Satinder Singh (2000). "Eligibility traces for off-policy policy evaluation". In: *Proceedings of The 17th International Conference on Machine Learning*, pp. 759–766.

Puterman, Martin L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

Rummery, Gavin and Mahesan Niranjan (1994). *On-line Q-learning using connectionist systems*. Tech. rep. CUED/F-INFENG/TR 166. Cambridge University.

Scherrer, Bruno (2013). "Improved and Generalized Upper Bounds on the Complexity of Policy Iteration". In: *Advances in Neural Information Processing Systems 26*, pp. 386–394.

Schulman, John et al. (2015). "Trust region policy optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897.

Schulman, John et al. (2017). "Proximal Policy Optimization Algorithms". In: *arXiv*. Vol. 1707.06347.

Silver, David et al. (2014). "Deterministic policy gradient algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 387–395.

Silver, David et al. (2016). "Mastering the game of go with deep neural networks and tree search". In: *Nature* 529.4, pp. 484–489.

Silver, David et al. (2017). "Mastering the game of Go without human knowledge". In: *Nature* 550, pp. 354–359.

Singh, Satinder et al. (2000). "Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms". In: *Machine Learning* 38.3, pp. 287–308.

Sugiyama, Masashi (2015). *Statistical reinforcement learning: modern machine learning approaches*. Chapman and Hall/CRC.

Sutton, Richard S. (1988). "Learning to predict by the method of temporal differences". In: *Machine Learning* 3, pp. 9–44.

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

— (2017). *Reinforcement Learning: An introduction, second edition*. MIT Press. ISBN: http://incompleteideas.net/book/the-book-2nd.html.

Sutton, Richard S., Csaba Szepesvári, and Hamid Reza Maei (2008). "A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation". In: *Advances in Neural Information Processing Systems 21*, pp. 1609–1616.

Sutton, Richard S. et al. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12*, pp. 1057–1063.

Sutton, Richard S. et al. (2009). "Fast gradient-descent methods for temporal-difference learning with linear function approximation". In: *Proceedings of The 26th International Conference on Machine Learning*, pp. 993–1000.

Szepesvári, Csaba (2010). "Algorithms for reinforcement learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Vol. 4. 1. Morgan and Claypool, pp. 1–103.

Tamar, Aviv, Dotan Di Castro, and Shie Mannor (2013). "Temporal Difference Methods for the Variance of the Reward To Go". In: *Proceedings of the 30th International Conference on Machine Learning*, pp. 495–503.

— (2016). "Learning the Variance of the Reward-To-Go". In: *Journal of Machine Learning Research* 17.

Tamar, Aviv et al. (2014). "Implicit temporal differences". In: *Neural Information Processing Systems, Workshop on Large-Scale Reinforcement Learning*.

Tesauro, Gerald (1995). "Temporal Difference Learning and TD-Gammon". In: *Communications of the ACM* 38.3, pp. 58–67.

Thomas, Philip S. (2014a). "Bias in Natural Actor-Critic Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 441–448.

— (2014b). "GeNGA: A generalization of natural gradient ascent with positive and negative convergence results". In: *Proceedings of the 31st International Conference on Machine Learning*.

— (2015). "Safe reinforcement learning". PhD thesis. University of Massachusetts Amherst.

Thomas, Philip S., Christoph Dann, and Emma Brunskill (2018). "Decoupling Gradient-Like Learning Rules from Representations". In: *Proceedings of The 35nd International Conference on Machine Learning*, pp. 4917–4925.

Thomas, Philip S., Georgios Theocharous, and Mohammad Ghavamzadeh (2015a). "High Confidence Off-Policy Evaluation". In: *Proceedings of the 29th Conference on Artificial Intelligence (AAAI-15)*, pp. 3000–3006.

— (2015b). "High Confidence Policy Improvement". In: *Proceedings of The 32nd International Conference on Machine Learning*, pp. 2380–2388.

Thomas, Philip S. et al. (2013). "Projected Natural Actor-Critic". In: *Advances in Neural Information Processing Systems 26*, pp. 2337–2345.

Thomas, Philip S. et al. (2016). "Energetic natural gradient descent". In: *Proceedings of The 33nd International Conference on Machine Learning*, pp. 2887–2895.

Thomaz, Andrea L. and Cynthia Breazeal (2006). "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance". In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 1000–1005.

Toulis, Panos and Edoardo M Airoldi (2015). "Scalable estimation strategies based on stochastic approximations: classical results and new insights". In: *Statistics and computing* 25.4, pp. 781–795.

Toulis, Panos, Jason Rennie, and Edoardo M. Airoldi (2014). "Statistical analysis of stochastic gradient methods for generalized linear models". In: *Proceedings of The 31st International Conference on Machine Learning* 32.1, pp. 667–675.

Toulis, Panos, Dustin Tran, and Edo Airoldi (2016). "Towards stability and optimality in stochastic gradient descent". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 1290–1298.

Tsitsiklis, John N. and Benjamin Van Roy (1997). "An analysis of temporal-difference learning with function approximation". In: *IEEE Transactions on Automatic Control* 42.5, pp. 674–690.

van Hasselt, Hado, A. Rupam Mahmood, and Richard S. Sutton (2014). "Off-policy TD(λ) with a true online equivalence". In: *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pp. 330–339.

van Seijen, Harm and Richard S. Sutton (2014). "True Online TD(λ)". In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 692–700.

Wagner, Paul (2011). "A reinterpretation of the policy oscillation phenomenon in approximate policy iteration". In: *Advances in Neural Information Processing Systems 24*, pp. 2573–2581.

— (2013). "Optimistic policy iteration and natural actor-critic: A unifying view and a non-optimality result". In: *Advances in Neural Information Processing Systems 26*, pp. 1592–1600.

— (2014). "Policy oscillation is overshooting". In: *Neural Networks* 52, pp. 43–61.

Wang, Tao, Michael Bowling, and Dale Schuurmans (2007). "Dual Representations for Dynamic Programming and Reinforcement Learning". In: *Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 44–51.

Wang, Tao et al. (2008). "Stable dual dynamic programming". In: *Advances in neural information processing systems 20*, pp. 1569–1576.

Wang, Ziyu et al. (2017). "Sample Efficient Actor-Critic with Experience Replay". In: *The 5th International Conference on Learning Representations*.

Watkins, Christopher John Cornish Hellaby (1989). "Learning from Delayed Rewards". PhD thesis. Cambridge University.

Watkins, Christopher John Cornish Hellaby and Peter Dayan (1992). "Q-learning". In: *Machine Learning* 8, pp. 279–292.

Wu, Yuhuai et al. (2017). "Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation". In: *Advances in Neural Information Processing Systems 30*, pp. 5279–5288.

Ye, Yinyu (2011). "The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate". In: *Mathematics of Operations Research* 36.4, pp. 593–603.

List of Publications

Journal Articles

- Ryo Iwaki, Hiroki Yokoyama and Minoru Asada. "Incremental Estimation of Natural Policy Gradient with Relative Importance Weighting." IEICE Transactions on Information and Systems, Vol. E101-D, No.9, pp. 2346-2355, 2018.
- Ryo Iwaki and Minoru Asada. "Implicit incremental natural actor critic algorithm." Neural Networks, Vol. 109, pp. 103-112, 2019.

Conference Proceedings (refereed)

- Ryo Iwaki and Minoru Asada. "Incremental, Scalable and Stable Algorithms for Natural Policy Gradient Estimation." The 3rd Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM), 2017.
- Ryo Iwaki and Minoru Asada. "Implicit Incremental Natural Actor Critic." The 24th International Conference on Neural Information Processing (ICONIP), 2017.
- Ryo Iwaki, Hideyuki Takahashi and Minoru Asada. "Adaptive Leader-Follower Role Switching Based on Rhythm Stability: Toward Modeling of Dynamic Infant-Caregiver Interaction." In Proceedings of the Fourth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics, Vol.USB, pp.100–101, 2014.

Awards

- JSAI Annual Conference Student Incentive Award, 2016.