Policy Gradient Methods for Control Applications

Novel approaches, review of previous methods, and feasibility studies

by Jan Peters

with support from Stefan Schaal, Sethu Vijaykumar, and Auke Ijspeert.

Contents

1 Introduction **1**

2 Foundations of Policy Gradient Methods **9**
2.1 Expected Return Definition 10
2.2 Value Functions 13
2.3 Trajectories, Transition Kernels and Stationary Distributions 18
2.4 Kernel-based Expected Return 22
2.5 Analytical solution for the value function 24

3 Policy Gradient Theory **27**
3.1 Policy Gradient Theorem 27
3.2 Baselines 28
3.3 Compatible Function Approximation 31
3.4 The Optimal Baseline 35
3.5 All-Action Algorithm 36
3.6 Natural Gradient 38

Bibliography **45**

A Proofs for the Examples **47**
A.1 Linear Quadratic Regulation Examples 47
A.1.1 Expected Return Derivation 47
A.1.2 Value Function Derivation 47
A.1.3 Advantage Function Derivation 51
A.1.4 Compatible Function Approximation 52
A.1.5 All-Action Matrix Derivation 53
A.1.6 53
A.2 Discrete State- and Action Example 53
A.3 All-Action Matrix 53

Chapter 1

Introduction

Reinforcement learning can best be described as the study of stochastic programming methods for (partially) Markovian decision processes without an analytical model of the system. These methods are intended to find the **optimal policy** π^* with the **maximal expected return** $J(\pi^*)$ in a multi-stage Markovian decision problem

Optimal policy

$$\pi^* = \underset{\pi_{\boldsymbol{\theta}} \in \Pi}{\operatorname{argmax}} J(\pi_{\boldsymbol{\theta}}), \tag{1.1}$$

$$J(\pi^*) = \max_{\pi_{\boldsymbol{\theta}} \in \Pi} J(\pi_{\boldsymbol{\theta}}), \tag{1.2}$$

in a goal directed manner[1] (Bertsekas, 2000). In here, Π denotes the space of all admissible policies. $J(\pi_{\boldsymbol{\theta}})$ denotes the expected return of a particular policy $\pi_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$. It can can be defined as

$$J(\pi_{\boldsymbol{\theta}}) = E_{\mathcal{T}}\left\{R\left(\mathcal{T}\right)\right\} = \int_{\mathbb{T}} p\left(\mathcal{T}\left|\pi\right.\right) R\left(\mathcal{T}\right) d\mathcal{T}, \tag{1.3}$$

where $R\left(\mathcal{T}\right)$ denotes the return of a particular trajectory $\mathcal{T}$ having a probability $p\left(\mathcal{T}\left|\pi_{\boldsymbol{\theta}}\right.\right)$ given the current policy $\pi_{\boldsymbol{\theta}}$ (for details see Chapter 2). Clearly, this stochastic programming problem is difficult to solve since there are infinite trajectories $\mathcal{T}$, and neither all rewards $R\left(\mathcal{T}\right)$ nor all probabilities $p\left(\mathcal{T}\left|\pi\right.\right)$ are known to the learning system in the general case.

Stochastic programming offers us two different traditional approaches of solving such problems: (a) **greedy external sampling** approaches, and (b) **parameterized internal sampling** methods (Morton, 2001). Reinforcement learning methods which have been introduced to date can be divided into this scheme as shown in in Figure 1.1, and Table 1.1.

Since the dawn of reinforcement learning, **greedy methods** dominated the field. For applying this approach, researchers focussed on a particular kind of policies, i.e., greedy policies

Greedy methods

$$\pi^*\left(\mathbf{u}\left|\mathbf{x}\right.\right) = \left\{ \begin{array}{ll} 1 & \text{if} \quad \mathbf{u} = \operatorname{argmax}_{\mathbf{u}^* \in \mathbb{U}} Q^*\left(\mathbf{x}, \mathbf{u}^*\right) \\ 0 & \text{if} \quad \mathbf{u} \neq \operatorname{argmax}_{\mathbf{u}^* \in \mathbb{U}} Q^*\left(\mathbf{x}, \mathbf{u}^*\right) \end{array} \right. , \tag{1.4}$$

which are parameterized by value functions such as the state-action value function

$$Q^*\left(\mathbf{x}, \mathbf{u}\right) = E\left\{ \sum_{k=0}^{\infty} \gamma^k r_k \middle| \mathbf{x}_0 = \mathbf{x}, \mathbf{u}_0 = \mathbf{u}, \pi^* \right\}. \tag{1.5}$$

Learning would proceed in the fashion that first an arbitrary initial policy π_0^* would be chosen. Then its value function Q_0^* would be estimated, and subsequently the new policy

[1] The goal-directedness makes reinforcement learning different from Genetic algorithms as these are based on a pure Monte-carlo search strategy.

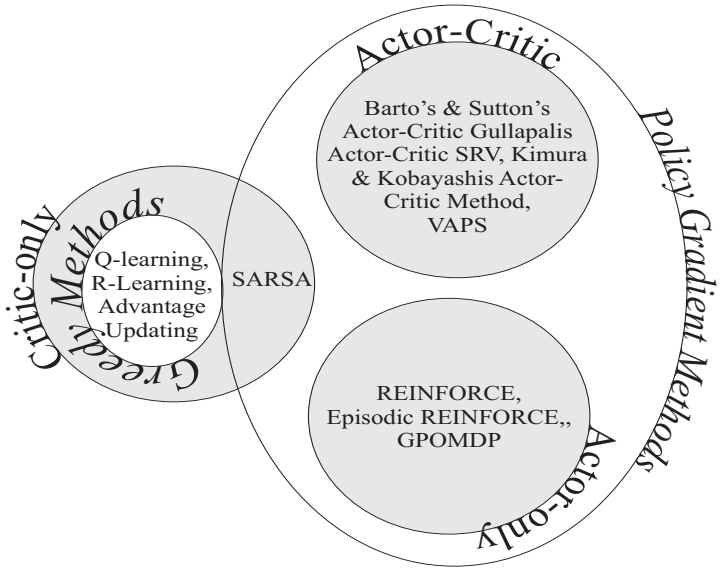


Figure 1.1: This figure shows how traditional reinforcement learning methods mapped onto stochastic programming methods, i.e., greedy and policy gradient methods.

Method	**Actor-only**	**Critic-only**	**Actor-critic**
Greedy Policy Optimization	None	Q-Learning, Advantage Updating, R-Learning, Value-Iteration	None
Parameterized Policy Optimization (= policy gradient methods)	REINFORCE, Episodic REINFORCE, GPOMDP	SARSA with an ε-greedy policy in discrete state and action spaces.	Suttons & Bartos Actor Critic, Kimuras & Kobyashis Actor Critic, VAPS

Table 1.1: Dominant reinforcement learning approaches in the late 1990s. Parameterized policy approaches can be seen as policy gradient methods as explained in Chapter 4.

π_1^* could be computed. After a sufficient amount of iterations $\pi_0^* \overset{\text{learn}}{\longrightarrow} Q_0^* \overset{\text{select}}{\longrightarrow} \pi_1^* \overset{\text{learn}}{\longrightarrow} Q_1^* \overset{\text{select}}{\longrightarrow} \cdots \overset{\text{learn}}{\longrightarrow} Q_\infty^* \overset{\text{select}}{\longrightarrow} \pi_\infty^*$, this approach can converge to the optimal policy just like its model-based counterparts in dynamic programming (Bellman, 1957) such as value iteration (Howard, 1960). However, convergence can **only** be guaranteed for lookup-table value function approximation.

This approach particularly appealed to researchers due to the fact that not only Monte Carlo methods but also **temporal difference (TD) methods** can be applied. These methods use the fact that reward of an action and the value function of the two temporally adjacent state-action pairs $(\mathbf{x}_t, \mathbf{u}_t)$ and $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$ allows the calculation of the TD error δ_t of Q^* using just two samples, e.g., in Q-learning it is given by

$$\delta_t = r_t + \gamma \max_{\mathbf{u}^* \in \mathbb{U}} Q^* \left(\mathbf{x}_{t+1}, \mathbf{u}^*\right) - Q^* \left(\mathbf{x}_t, \mathbf{u}_t\right).$$

Learning of the Q^*-function appeared to be the "key to efficiency" (Sutton, 2000), and an "almost supervised" (Sutton, 2000) learning problem. Furthermore, algorithms like Q-learning were proved to converge for the discrete case with look-up table value function approximation (Sutton & Barto, 1998), and impressive applications such as playing

Backgammon on grandmaster level, and acrobat swing-ups have been presented (Sutton & Barto, 1998).

Nevertheless, lookup table approximations suffer strongly from the **curse of dimensionality** (Bellman, 1957), and it was obvious that a higher order of generalization was needed. However, at the end of the 1990s when researchers turned towards continuous approximations of large-scale value functions, the success of the greedy value function approximators was limited (Sutton & Barto, 1998; Baxter & Bartlett, 1999; Baird, 1998). No theoretical guarantees of the performance could be obtained (Sutton, 2000). All existing greedy methods have been shown to diverge or oscillate for at least one example (Baird, 1998) already when linear function approximation is used to approximate the value function (obviously, more complex function approximation can result into more complex problems).

Greedy methods are biased

The reason for this is two-fold, and lies in the heart of the sequence of $\pi_i^* \xrightarrow{\text{learn}} Q_i^* \xrightarrow{\text{select}} \pi_{i+1}^*$ steps. In the learning step $\pi_i^* \xrightarrow{\text{learn}} Q_i^*$, we basically obtain an estimate $J_n^{\pi_i^*}$ of the expected return $J(\pi_i^*)$ using a finite amount of n trials. The estimate can be calculated as

$$J_n^{\pi_i^*} = \frac{1}{n} \sum_{k=0}^{n} R\left(\mathcal{T}_k\right). \tag{1.6}$$

In here, $R\left(\mathcal{T}_k\right)$ denotes the return of the k-th trajectory $\mathcal{T}_k$. In order to perform the selecting step $Q_i^* \xrightarrow{\text{select}} \pi_{i+1}^*$, we determine the maximal reward from our trial data $J_n^{\max} = \max_{\pi \in \Pi} J_n^{\pi}$, and at the same time we have determined the next policy $\pi_{i+1}^* = \operatorname{argmax}_{\pi \in \Pi} J_n^{\pi}$. We easily see that the expectation of the maximum of the average reward over several trials is equal or greater than the maximum of the expectation of one trial, i.e.,

$$\max_{\pi \in \Pi} J_n^{\pi} = E_{\mathcal{T}} \left\{ \max_{\pi \in \Pi} \frac{1}{n} \sum_{i=0}^{n} R\left(\mathcal{T}_i\right) \right\} \geq E_{\mathcal{T}} \left\{ \max_{\pi \in \Pi} R\left(\mathcal{T}\right) \right\} = \max_{\pi \in \Pi} J(\pi). \tag{1.7}$$

Therefore, **greedy methods are positively biased estimators** of the optimal policy, and the optimization process can be mislead due to the bias. In particular, a policy which performs optimally on a small amount of collected data might perform poor in the complete trajectory space. However, the bias decreases monotonically with the amount of data $\max_{\pi \in \Pi} J_n^{\pi} \geq \max_{\pi \in \Pi} J_{n+1}^{\pi} \geq \max_{\pi \in \Pi} J(\pi)$. This means, we better "wait and see" (Morton, 2001), before we select a new policy. This problem is not severe for typical discrete toy problems as we can oversample every state, and therefore we can minimize the bias for such problems. Nevertheless, it is disastrous for learning value functions with generalization in its function approximation as greedy methods are likely to steer the learning process towards a wrong generalization. Still, greedy optimization methods are strongly consistent (Morton, 2001).

Greedy methods are not sound with value function approximation

Despite this interesting fact, there is also another, more intuitive explanation: the greedy mapping $Q_i^* \xrightarrow{\text{select}} \pi_{i+1}^*$ from the value function Q_i^* onto a policy π_{i+1}^* is non-smooth, and discontinuous. Noise in the value function Q_i^* can cause the policy π_i^* to change fast since a small value function in Q_i^* change can cause a large policy change in the new policy π_{i+1}^* which in turn can cause a large value function change in Q_{i+1}^*. According to Sutton (2000) this can be described as follows:

> "All the **states interact** and must be balanced which causes trade-off between them. [...]. A small change (or error) in the value function estimate can cause a **large, discontinuous change** in the policy which in turn causes a **large change** in the value function estimate."

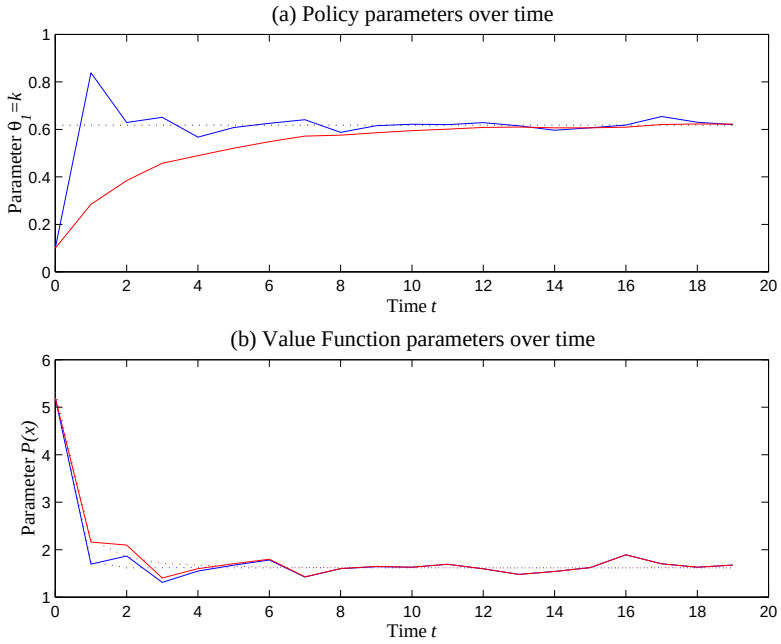


Figure 1.2: This figure shows a simple linear quadratic regulation problem (LQR) when the parameters are estimated from a noisy value function. The noisy value function parameters are shown in (b) with their dotted true values. The true values change over time since they depend on the policy parameters. In (a), we can observe how the resulting greedy and the resulting natural gradient estimator computed from this value function are affected. Obviously, the greedy estimator jumps around wildly while the natural gradient estimator goes into the correct solution smoothly.

Therefore, we often cannot obtain a stable learning system. This is highly related with the biasedness of the greedy approach: if the value function is inaccurate, the greedy approach is biased towards an incorrect solution. This in turn cause a larger change, which results in more noise, and more wrongly directed bias.

Conclusion 1 (Greedy methods) *As we have pointed out in the previous pages, the reason for the difficulties of reinforcement learning with function approximation is not that we have to learn a value function but that we do greedy updates. Greedy methods are characterized by two difficulties: (1) Both Monte Carlo and temporal difference based greedy methods are biased as the greedy optimization step is done on a finite amount of data, and a greedy step is therefore biased. (2) The discontinuous mapping from the value function onto a policy also produces a discontinuous change in the value function. Due to noise in the value function estimate, both value function and greedy policy can jump around wildly not reaching the optimal policy. Therefore, we have to search an alternative to greedy methods.*

Stochastic gradients are unbiased...

The resulting problems in greedy reinforcement learning brought researchers back to the question whether we have taken the right stochastic programming approach (Baxter

& Bartlett, 1999; Baird, 1998; Jaakkola, Jordan, & Singh, 1994; Marbach & Tsitsiklis, 1998)[2]. Clearly, internal sampling methods can provide us with an unbiased estimator of the optimal expected return, and the optimal policy: according to Robbins & Monroe (1951), a stochastic gradient estimator given by

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \left. \frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} \tag{1.8}$$

is unbiased. Furthermore, the effect of noise in the value function is not as drastic as it is illustrated in Figure 1.2 with a simple noisy learning LQR controller[3]. Nevertheless, the gradient $\partial J(\pi_{\boldsymbol{\theta}})/\partial \boldsymbol{\theta}$, defined by

... and noise is not that drastic

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{T}} \frac{\partial p(\mathcal{T}|\pi)}{\partial \boldsymbol{\theta}} R(\mathcal{T})\, d\mathcal{T}, \tag{1.9}$$

is difficult to obtain in practice.

Early policy gradients methods suffered from the large variance

Early methods of obtaining $\partial J(\pi_{\boldsymbol{\theta}})/\partial \boldsymbol{\theta}$ have been studied in the reinforcement learning community at the end of the 1980s when Ronald J. Williams (Williams & Peng, 1991; Williams, 1992) first introduced the class of "REward Increment = nonnegative Factor × Offset Reinforcement × Characteristic Eligibility" (REINFORCE) algorithms and Vijaykumar Gullapalli (Gullapalli, 1991; Gullapalli, Franklin, & Benbrahim, 1994) showed how an instance of this algorithm class, the Stochastic Real Valued (SRV) algorithm could be applied for several problems. However, while greedy methods are problematic due to the bias, these early gradient methods were problematic due to their large variance in the gradient estimate. This was due to the fact that they estimated the gradients from the immediate reward of a state, i.e., neglecting most terms in equation (1.9), or from **single roll-outs**. Such policy gradients estimated from immidiate rewards or single roll-outs will hardly ever be close to the true gradient, i.e., have a large variance. Due to the large variance in the gradient, the convergence rate of these methods was low. This in turn is the reason why they were abandoned in the 1990s.

Low convergence rate but unaffected by dimensionality

Nevertheless, when using multiple roll-outs, the policy gradient estimate improves significantly. In fact, roll-out gradient estimation from $n \gg 1$ trials (i.e., trajectories) can work quite well since it is just a Monte Carlo integration algorithm over n trajectories. Monte carlo integration **converges with a rate of** $\propto 1/\sqrt{n}$ (Morton, 2001). While this appears a slow for low-dimensional state spaces ($d \approx 1$), it turns out that it is a superb result for high-dimensional spaces ($d \gg 1$) as the convergence rate is **not affected by the dimensionality** d (Morton, 2001). However, the variance in the estimate grows (linearly?) with d (Morton, 2001). Clearly, these methods are capable of overcoming the curse of dimensionality unlike lookup-table based greedy methods.

During the revival of policy gradient methods, Sutton et al. (2000) and Konda & Tsitsiklis (2001, 2000, 2002) presented a general policy gradient theorem which unified previous approaches, and comes with a compatible value function approximation. Building on their results and Suttons unfinished paper (Sutton, McAllester, Singh, & Mansour, 2001), we will present methods of obtaining minimum-variance estimates of the policy gradients, i.e., optimal baselines, and the All-Action algorithm. Furthermore, it appears that all non-greedy reinforcement learning methods shown in Table 1.1 are to some extend policy gradient methods. Surprisingly, this includes SARSA as well as the actor-critic methods of the beginning 1980s, we will study this in Chapter 4.

Natural gradients are efficient

Natural gradient methods as presented by Amari (2000) have the large advantage over

[2]Many of the researchers did not attribute the problems associated with the greedy approach to the policy but to the fact that we have to learn a value function.

[3]The analysis of this example as a whole is given in chapters 2-3. Here, we just give it as an illustration of greedy value function based optimization.

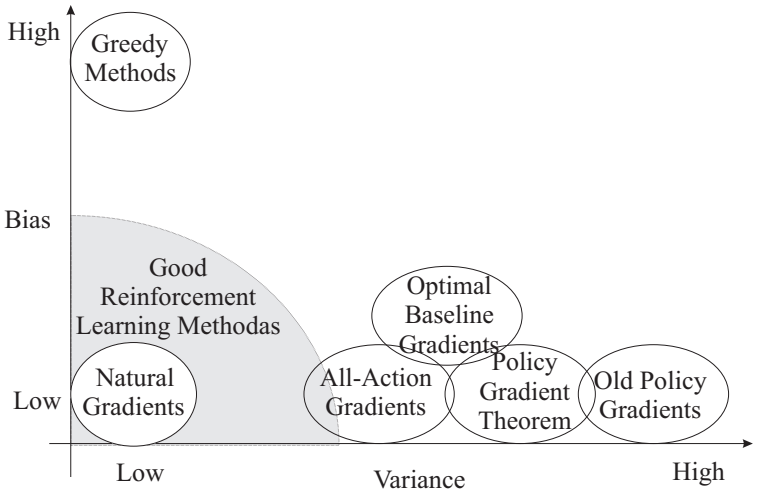


Figure 1.3: This figure shows different reinforcement learning methods in a bias variance diagram. Greedy methods have little variance but a large bias. The old policy gradient methods, i.e., REINFORCE, and episodic REINFORCE, have a huge variance in expectation. The policy gradient theorem reduces the variance in the gradients significantly due to the usage of value functions instead of the actual return. However, in practical application, it is used only with single actions, and states which have occured. The variance can be improved using a an optimal baselines which reduces the variance but introduces bias. It can further be reduced using the all-action algorithm - which still has a variance due to the fact that the stationary distribution has to be estimated. Only the natural gradients deliver us an unbiased minimum variance estimator.

normal gradient methods, that they are not only unbiased but also consistent, efficient, minimum variance estimators. Kakade (2001) presented an average natural gradient for policy gradient methods. Building on this we will relate the results of Kakade (2001) to the all action algorithm, and extend his proof from the "average natural gradient" towards the true natural gradient.

Conclusion 2 (Policy gradient methods) *Policy gradient methods avoid the problems of greedy approach as gradient estimators are unbiased, and noise in the value function does not have such drastic effects. Nevertheless, early gradient methods did not become popular due to the large variance in the gradient. As we will see in this report, optimal baselines, All-Action policy gradients, and natural gradients allow us to obtain a minimum variance estimate of the policy gradient. Furthermore, the natural gradient estimator is unbiased, consistent, and efficient.*
The second problem of gradient methods are plateaus and local minima. While the latter cannot be solved by current methods, we will see that the natural gradient can indeed solve the problems of plateaus.

While supervised learning as well as unsupervised learning methods have matured to highly efficient techniques, reinforcement learning has remained a method for solving toy-problems despite few impressive applications. Policy gradient methods currently appear

the only choice to speed-up reinforcement learning so that it can be applied in large-scale applications. We intend to analyze policy gradient approaches in order to bring them into an applicable framework. We test this framework on two case studies and will HOPEFULLY show that it outperforms greedy value function based methods[4].

In this research paper, we will proceed as follows. In the second chapter, we will discuss the foundations of policy gradient methods, i.e., the expected return of the policy, probability distributions related to policy gradient methods, and value function. In particular, we will contribute two important results here, i.e., (i) the transformation from the trajectory view used in this introduction onto a sample based view, and (ii) the condition for value function approximations to be able to represent the value function of a reinforcement learning problem. Fact (i) is particular powerful as it reduces a controlled Markov decision process onto an uncontrolled Markov chain.

In the third chapter, we discuss policy gradient theory based on the works of Sutton et al. (2000), Konda & Tsitsiklis (2001, 2000, 2002), and Kakade (2001). We will attempt to provide the missing links in their work, and present sufficient examples to undermine our claims.

In the fourth chapter, we will discuss several non-greedy reinforcement learning methods, and show that all of these methods are to some extend policy gradient methods. We have stressed this point already in the introduction, and most of the methods listed in Table 1.1 will be discussed here.

In the fifth chapter, we will study the learning of the policy compatible value function approximation. In the sixth chapter, we will present a case study on cart-pole and other control problems. Throughout this report, we assume the reader to be familiar with the book "Reinforcement Learning" by Sutton and Barto (1998).

[4]We do not count methods based upon ϵ-greedy policies as greedy methods. In fact, many ϵ-greedy methods presented in Sutton & Barto (1998) are policy gradient methods.

Chapter 2

Foundations of Policy Gradient Methods

In reinforcement learning problems, we generally have an **actor** which is in a **state** $\mathbf{x} \in \mathbb{X}$, and takes an **action** $\mathbf{u} \in \mathbb{U}$ according to a **policy** or probability distribution $\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) = p(\mathbf{u}|\mathbf{x}, \boldsymbol{\theta}) \in [0, 1]$. This policy has the **internal parameters** $\boldsymbol{\theta} \in \mathbb{R}^n$, and $p(\mathbf{u}|\mathbf{x}, \boldsymbol{\theta})$ denotes the probability of taking action $\mathbf{u}$ in state $\mathbf{x}$. Its actions modify the state $\mathbf{x}$ in accordance to the **environment** which is presented in form of a probability distribution of the next state $p(\mathbf{x}'|\mathbf{x}, \mathbf{u})$, and is usually not known to the actor. For each action it receives a **reward** $r(\mathbf{x}, \mathbf{u})$. Throughout this report we will deal with discrete time t but continuous states $\mathbf{x}$ and actions $\mathbf{u}$. Furthermore, throughout this report we assume that $\mathbf{x}$ is fully observable and $p(\mathbf{x}'|\mathbf{x}, \mathbf{u})$ depends only on $\mathbf{x}$, $\mathbf{u}$ (and not the whole preceding state-action sequence). Therefore, we are dealing only with **Markov decision problems** (MDPs). The whole setup is shown in Figure 2.1, and examples are given in Figure 2.2.

Reinforcement learning problems

Example 1 (Linear quadratic control) *Throughout this report, we will deal with the example of controlling a linear system with a* ***linear quadratic regulator*** *(LQR). For simplicity, we will just consider scalar actions, i.e., $\mathbb{U} = \mathbb{R}$, but high-dimensional state spaces $\mathbb{X} = \mathbb{R}^n$. This system is a Markovian decision problem and has state-transition probability distribution of the system given by*

LQR problems

$$p(\mathbf{x}'|\mathbf{x}, u) = \begin{cases} 1 & \text{if} \quad \mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}u, \\ 0 & \text{if} \quad \mathbf{x}' \neq \mathbf{A}\mathbf{x} + \mathbf{b}u, \end{cases}$$

in the noise-free case. The system matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, and the input vector $\mathbf{b} \in \mathbb{R}^n$ can be arbitrary. Its rewards are defined as

$$r(\mathbf{x}, u) = -\frac{1}{2}\mathbf{x}^T \mathbf{Q}\mathbf{x} - \frac{1}{2}u^T R u,$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$, and $R \in \mathbb{R}$ are positive definite. A simple example for such a problem is given in Figure 2.2 (a-b). This system can be controlled optimally by a linear controller $u = \mathbf{k}_{opt}^T \mathbf{x}$ (Dorato, Abdallah, & Cerone, 1995). This controller can be represented by a ***Gaussian policy***

Gaussian policy

$$\pi_{\boldsymbol{\theta}}(u|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(u - \mathbf{k}^T\mathbf{x}\right)^2\right),$$

with the policy parameters $\boldsymbol{\theta} = \left[\mathbf{k}^T, \sigma\right]^T \in \mathbb{R}^n \times \mathbb{R}_+$. In here, we have the controller $\mathbf{k} \in \mathbb{R}^n$, and the exploration rate $0 \leq \sigma \in \mathbb{R}$. For $\sigma \to 0$, and $\mathbf{k} \to \mathbf{k}_{opt}$, the Gaussian

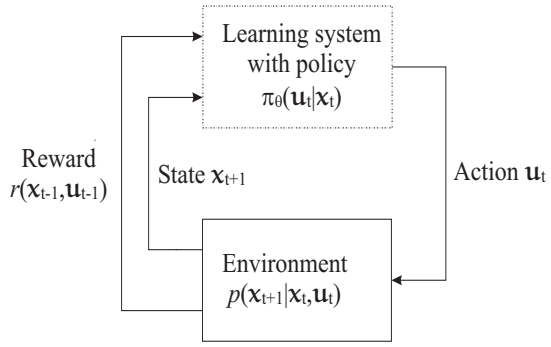


Figure 2.1: The standard Markov decision reinforcement learning problem consisting of a learning system, and the environment. The learning system is based on a policy $\pi_{\boldsymbol{\theta}}(\mathbf{u}_t|\mathbf{x}_t)$ which denotes the probability of taking action $\mathbf{u}_t$ in state $\mathbf{x}_t$. Its parameter vector $\boldsymbol{\theta}$ is adapted during learning. The environment consits of the state transition probabilities, and a reward. The transition probabilities $p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{u}_t)$ denote the probability of the next state $\mathbf{x}_{t+1}$ given the current state $\mathbf{x}_t$, and action $\mathbf{u}_t$. The environment also yields a reward $r(\mathbf{x}_{t-1},\mathbf{u}_{t-1})$ for the previous step. This figure is similar to the one of Sutton and Barto (1998).

policy becomes the optimal controller. The parameters $\mathbf{A}$, $\mathbf{b}$, $\mathbf{Q}$, *and* R *are fixed parameters of the environment – not of the policy. The optimal controller* $\mathbf{k}_{opt}$ *is determined by the environment.*

Discrete problems

Example 2 (Discrete state and action problems) *Another, more common example, is a discrete action, discrete state Markov decision problem with* n *states, and* m *actions. The state space is given by* $\mathbb{X} = \{x_1, \ldots, x_n\}$, *and the action space by* $\mathbb{U} = \{u_1, \ldots, u_m\}$. *Here we are given a table of state transitions probabilities* $\mathcal{P}^u_{xx'}$ *to describe* $p(x'|x,u) = \mathcal{P}^u_{xx'}$. *Similarly, the rewards are given in tabular form by* $r(x,u) = \mathcal{R}^u_x$. *An example for such a problem is given in Figure 2.2 (c-e). A common policy for such problems is the* ***Gibbs policy***

Gibbs policy

$$\pi(u|x) = \frac{\exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}_{xu}\right)}{\sum_{i=1}^m \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}_{xu_i}\right)},$$

with internal parameters $\boldsymbol{\theta} \in \mathbb{R}^{n\times m}$, *and features vectors* $\boldsymbol{\phi}_{xu} \in \mathbb{R}^{n\times m}$. *This policy is known to be able to represent the optimal policy for such problems. However, its parameters might have to take infinite values in order to generate a deterministic policy. Therefore, another policy, i.e., a* ***decision border policy****, might be more appropriate*

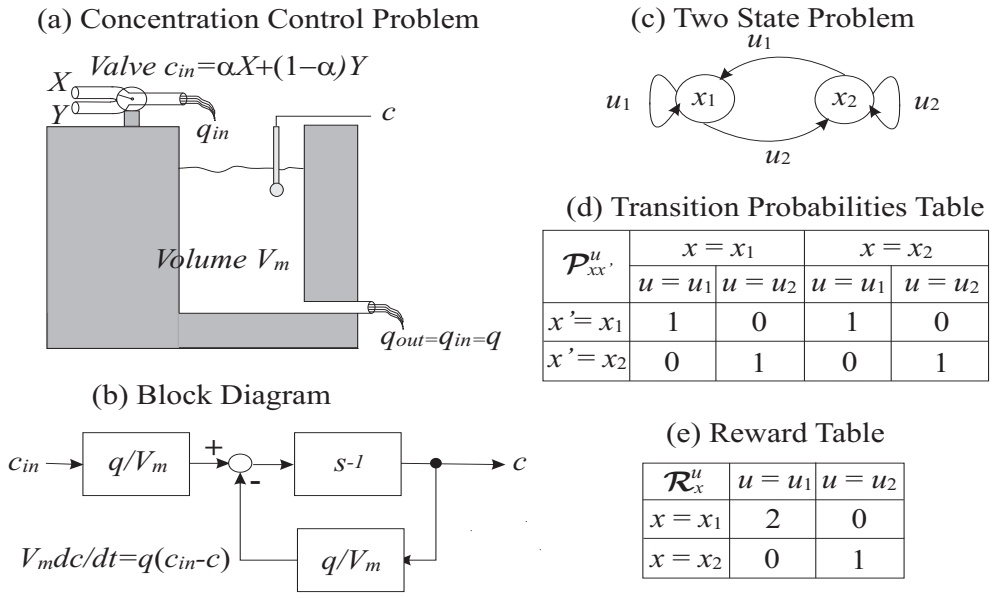
Decision border policy

$$\pi(u_j|x_i) = \begin{cases} \theta_{x_i u_j} & \text{if} \quad j < m, \\ 1 - \sum_{k=1}^{m-1} \theta_{x_i u_k} & \text{if} \quad j = m. \end{cases}$$

2.1 Expected Return Definition

Expected return

In general, reinforcement learning algorithms intend to derive a policy $\pi_{\boldsymbol{\theta}}$ which maximizes the **expected return** $J(\pi_{\boldsymbol{\theta}})$. Two formulations of this return have been given (Sutton et al., 2000), i.e., the average-reward formulation and the start-state formulation. Let us now define both of these cases.



$\mathcal{P}^u_{xx'}$	$x = x_1$		$x = x_2$	
	$u = u_1$	$u = u_2$	$u = u_1$	$u = u_2$
$x' = x_1$	1	0	1	0
$x' = x_2$	0	1	0	1

$\mathcal{R}^u_x$	$u = u_1$	$u = u_2$
$x = x_1$	2	0
$x = x_2$	0	1

Figure 2.2: This figure shows two simple examples. In (a) you can see a simple, one dimensional linear quadratic control problem, i.e., controlling the concentration c of a substance X (e.g., Cloride) in a basin of substance Y (e.g., water). The amount of inflow q_{in} equals the outflow, i.e., $q_{out} = q_{in} = q$. By tuning the inflow concentration c_{in}, we can control the basin concentration c. In (b), the block diagram and equations of the system dynamics are given. In (c) you can see a simple two state, discrete action and state problem, previously presented in (Kakade, 2002; Mahadevan, 1996). In (d) the transition probabilities $\mathcal{P}^u_{xx'}$ and in (e) the rewards $\mathcal{R}^u_x$ are given.

Definition 1 (Expected return) *The **expected return** $J(\pi_\theta)$ of a policy π_θ can be defined in two ways. The **average reward formulation** is given by*

$$J(\pi_\theta) = \lim_{T\to\infty} \frac{1}{T} E\left\{ \sum_{t=0}^{T} r\left(\mathbf{x}_t, \mathbf{u}_t\right) \middle| \pi_\theta \right\}, \tag{2.1}$$

*and the **discounted start-state formulation** is given by*

$$J_\gamma(\pi_\theta \,|\mathbb{X}_0\,) = \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} \gamma^t r\left(\mathbf{x}_t, \mathbf{u}_t\right) \middle| \pi_\theta, \mathbb{X}_0 \right\}, \tag{2.2}$$

where γ denotes the discounting factor for future rewards, and $\mathbb{X}_0$ the set of start states.

For analyzing the expected return in-depth, we will have to focus on a few necessary topics. Before doing so, let us pick up our previous examples, i.e., linear quadratic control, and the two-state problem. We will now give the expected return of both examples.

Example 3 (Linear quadratic control) *We will take up the linear quadratic regulation problem from Example 1, page 9, with the same Gaussian policy π_θ. The return of the* *LQR expected return*

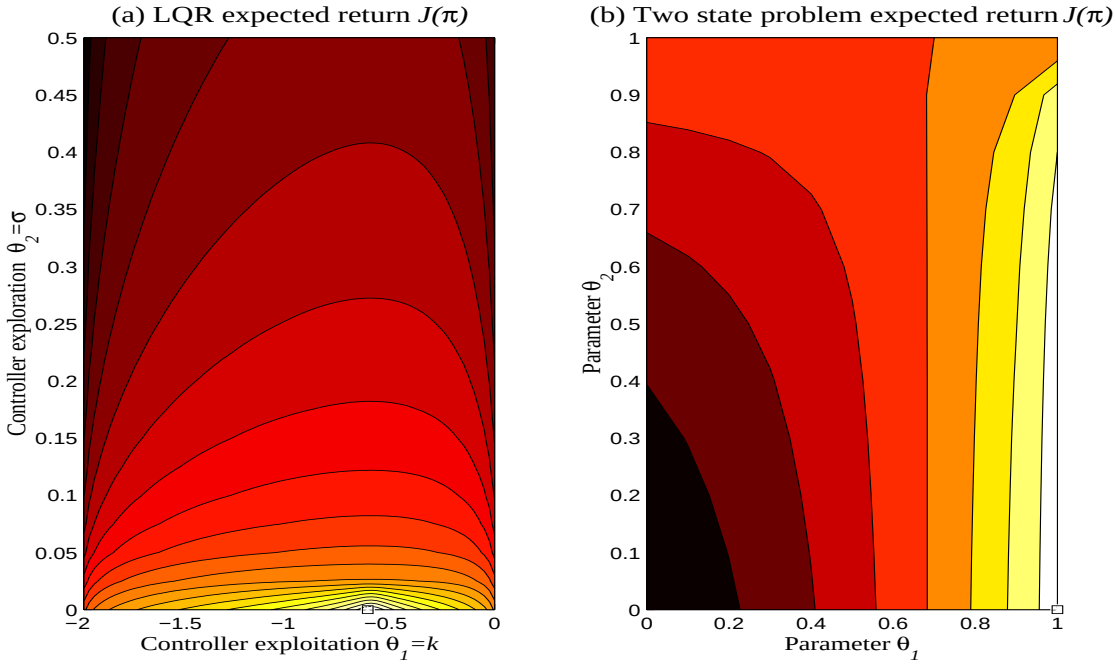


Figure 2.3: This figure shows the average reward in the discounted start-state formulation for both (a) the one-dimensional linear quadratic control (LQR) problem with a Gaussian policy, and (b) the the two-state problem with a decision border policy. In the LQR problem (a) the parameters of the system are $A = b = R = Q = 1$, $\gamma = 0.95$, and the parameters of the Gaussian policy $\pi_{\boldsymbol{\theta}}(u|x)$ are $\boldsymbol{\theta} = [k, \sigma]^T$ are shown on the axes. In the two state problem (b), we define the policy parameters $\theta_{x_1u_1} = \theta_1$, $\theta_{x_1u_2} = 1-\theta_2$, $\theta_{x_2u_1} = 1-\theta_2$, and $\theta_{x_2u_2} = \theta_2$. Furthermore, we also use a discount factor of $\gamma = 0.95$ in this problem. The optimal solutions $\boldsymbol{\theta}^*$ of both problems are $\boldsymbol{\theta}^*_{\text{LQR}} = [-0.6037, 0]^T$ for (a), and $\boldsymbol{\theta}^*_{\text{TwoState}} = [1, 0]^T$ for (b) are indicated by the rectangles.

policy in the average reward formulation is given by

$$\begin{aligned} J(\pi_{\boldsymbol{\theta}}) &= \lim_{T\to\infty} \frac{1}{T} E\left\{ \sum_{t=0}^{T} r\left(\mathbf{x}_t, \mathbf{u}_t\right) \middle| \pi_{\boldsymbol{\theta}} \right\}, \\ &= \lim_{T\to\infty} \frac{1}{T} E\left\{ -\frac{1}{2}\mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 - \frac{1}{2} \sum_{t=0}^{T} \sigma \left(R + \mathbf{b}^T \mathbf{P} \mathbf{b}\right) \sigma \right\}, \\ &= -\frac{1}{2}\left(R + \mathbf{b}^T \mathbf{P} \mathbf{b}\right) \sigma^2, \end{aligned}$$

if the system is stable. The Ricatti matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ is the positive definite solution of the modified Ricatti equation $\mathbf{P} = \mathbf{Q} + \gamma \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{k}\mathbf{b}^T \mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} \mathbf{b} \mathbf{k}^T + \mathbf{k}\mathbf{b}^T \mathbf{P} \mathbf{b} \mathbf{k}^T + \mathbf{k} R \mathbf{k}^T$. Similarly, the return of the policy in the discounted start-state formulation is given by

$$\begin{aligned} J_\gamma(\pi_{\boldsymbol{\theta}} \,|\{\mathbf{x}_0\}) &= \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} \gamma^t r\left(\mathbf{x}_t, \mathbf{u}_t\right) \middle| \pi_{\boldsymbol{\theta}}, \mathbf{x}_0 \right\}, \\ &= -\frac{1}{2}\mathbf{x}_0^T \mathbf{P}_\gamma \mathbf{x}_0 - \frac{1}{2}\frac{1}{1-\gamma}\left(R + \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{b}\right) \sigma^2. \end{aligned}$$

The Ricatti matrix $\mathbf{P}_\gamma \in \mathbb{R}^{n\times n}$ is changed in comparison to the average reward case in order to take the discount factor into account. It is the positive definite solution of the modified

Ricatti equation $\mathbf{P}_\gamma = \mathbf{Q} + \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{A} - \gamma\mathbf{k}\mathbf{b}^T\mathbf{P}_\gamma\mathbf{A} - \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{b}\mathbf{k}^T + \gamma\mathbf{k}\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\mathbf{k}^T + \mathbf{k}R\mathbf{k}^T$. *In here, we assume a special case, i.e., the case where we are always starting from state* $\mathbf{x}_0 \in \mathbb{X}$. *The expected return in average reward formulation for the concentration control LQR problem from Figure 2.2 (a-c) is given in Figure 2.3 (a). Alternatively, we could assume a bounded start-state set with an initial probability distribution. The proof of this example can be found in appendix Section A.1.1, page 47.*

Example 4 (Discrete state and action spaces) *Similarly, we can study finite discrete state- and action-space problems from Example 2, page 10. In this case, we have to generate a state transition matrix* $\mathbf{P} = [P_{ij}] \in \mathbb{R}^{n\times n}$ *where the matrix entries* P_{ij} *are defined by* $P_{ij} = \sum_{k=1}^{m} p(x_i|x_j,u_k)\,\pi(u_k|x_j) = \sum_{k=1}^{m} \mathcal{P}_{x_j x_i}^{u_k}\pi(u_k|x_j)$. *The state-reward vector* $\mathbf{R} = [R_j] \in \mathbb{R}^n$ *so that the average reward for all actions in this state becomes* $R_j = \sum_{k=1}^{m} r(x_j,u_k)\,\pi(u_k|x_j) = \sum_{k=1}^{m} \mathcal{R}_{x_j}^{u_k}\pi(u_k|x_j)$. *Furthermore, if we assume that each state* $x_i \in \mathbb{X}_0$ *has the probability* $p(x_i)$ *of being a start-state, we can combine these in the start-state vector* $\mathbf{S} = [S_i] \in \mathbb{R}^n$ *so that* $S_i = p(x_i)$. *Using these vectors, we are given the expected return of the policy in discounted start-state formulation by* *Discrete problems expected return*

$$\begin{aligned} J(\pi_{\boldsymbol{\theta}}|\mathbb{X}_0) &= \lim_{N\to\infty} E\left\{\sum_{t=0}^{N}\gamma^t r(x_t,u_t)\middle|\pi_{\boldsymbol{\theta}},\mathbb{X}_0\right\}, \\ &= \lim_{N\to\infty}\mathbf{S}^T\sum_{t=0}^{N}\gamma^t\mathbf{P}^t\mathbf{R} = \lim_{N\to\infty}\mathbf{S}^T(\mathbf{I}-\gamma\mathbf{P})^{-1}\left(\mathbf{I}-\gamma^N\mathbf{P}^N\right)\mathbf{R}, \\ &= \mathbf{S}^T(\mathbf{I}-\gamma\mathbf{P})^{-1}\mathbf{R}, \end{aligned}$$

The average return for the undiscounted case cannot easily be determined analytically at this stage. We will later return to it after introducing transition kernels and stationary distributions. An example for such a problem is given in Figure 2.2 (c-e), and the expected return is shown in Figure 2.3 (b). This also is an example for theorem 2 in Section 2.4, page 22.

We will return to the expected return of the policy at later stages of this report. In order to treat the topic of estimating the average reward properly, we have to discuss further topics: value functions, trajectories, and transition kernels.

2.2 Value Functions

Since the dawn of reinforcement learning when Bellman derived the dynamic programming algorithms (Bellman, 1957), value functions have been an important tool for reinforcement learning. In average reward formulation, the **state value function** $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$ is defined as the accumulated difference from the expected return in the steps following the visit $\mathbf{x}$. In discounted formulation, the state value function $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$ is defined as the accumulated discounted rewards in all the steps after visiting $\mathbf{x}$. *State value function* $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$

The state value function $V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$ can be seen as a potential function of the state-space $\mathbb{X}$ similar to the potential in classical electrodynamics, and vector analysis. The **state-action value function** $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u})$ is its counterpart in the state-action space $\mathbb{X}\times\mathbb{U}$. We will now shortly define, and discuss value functions. *State-action value function* $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u})$

Definition 2 (Value functions) *The **state-action value function** $Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ for the **average reward case** is given by*

$$Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} r(\mathbf{x}_t, \mathbf{u}_t) - J(\pi_\theta) \middle| \mathbf{x}_0 = \mathbf{x}, \mathbf{u}_0 = \mathbf{u}, \pi_\theta \right\}, \tag{2.3}$$

*and the **state value function** $V^{\pi_\theta}(\mathbf{x})$ is given by*

$$V^{\pi_\theta}(\mathbf{x}) = \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} r(\mathbf{x}_t, \mathbf{u}_t) - J(\pi_\theta) \middle| \mathbf{x}_0 = \mathbf{x}, \pi_\theta \right\}, \tag{2.4}$$

*according to (Sutton & Barto, 1998). The state-action value function for the **discounted start-state case** is given by*

$$Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} \gamma^t r(\mathbf{x}_t, \mathbf{u}_t) \middle| \mathbf{x}_0 = \mathbf{x}, \mathbf{u}_0 = \mathbf{u}, \pi_\theta \right\}, \tag{2.5}$$

and the state value function is given by

$$V^{\pi_\theta}(\mathbf{x}) = \lim_{T\to\infty} E\left\{ \sum_{t=0}^{T} \gamma^t r(\mathbf{x}_t, \mathbf{u}_t) \middle| \mathbf{x}_0 = \mathbf{x}, \pi_\theta \right\}, \tag{2.6}$$

according to (Sutton & Barto, 1998).

The classical example for a value function is linear quadratic regulation, i.e., Example 1, page 9. Here we easily see that the basis functions of the value functions are quadratic, which allows us to derive the value functions. They are discussed in Example 5, and shown in Figure 2.4 (a-b).

LQR value functions

Example 5 (Linear quadratic control) *In the context of Gaussian policy linear quadratic regulation as in Example 1, page 9, we have the state-action value function of*

$$Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = -\frac{1}{2}\begin{bmatrix} \mathbf{x}^T, & u \end{bmatrix}\begin{bmatrix} \mathbf{Q} + \mathbf{A}^T\mathbf{P}\mathbf{A} & \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \mathbf{b}^T\mathbf{P}\mathbf{A} & \mathbf{R} + \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \frac{1}{2}\left(R + \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2,$$

and the state value function of

$$V^{\pi_\theta}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x},$$

for the average reward case. Similarly, we have the state-action value function of

$$Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = -\frac{1}{2}\begin{bmatrix} \mathbf{x}^T, & u \end{bmatrix}\begin{bmatrix} \mathbf{Q} + \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{A} & \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{b} \\ \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{A} & \mathbf{R} + \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} - \frac{1}{2}\frac{\gamma}{1-\gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2,$$

and the state value function of

$$V^{\pi_\theta}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{P}_\gamma\mathbf{x} - \frac{1}{2}\frac{1}{1-\gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2,$$

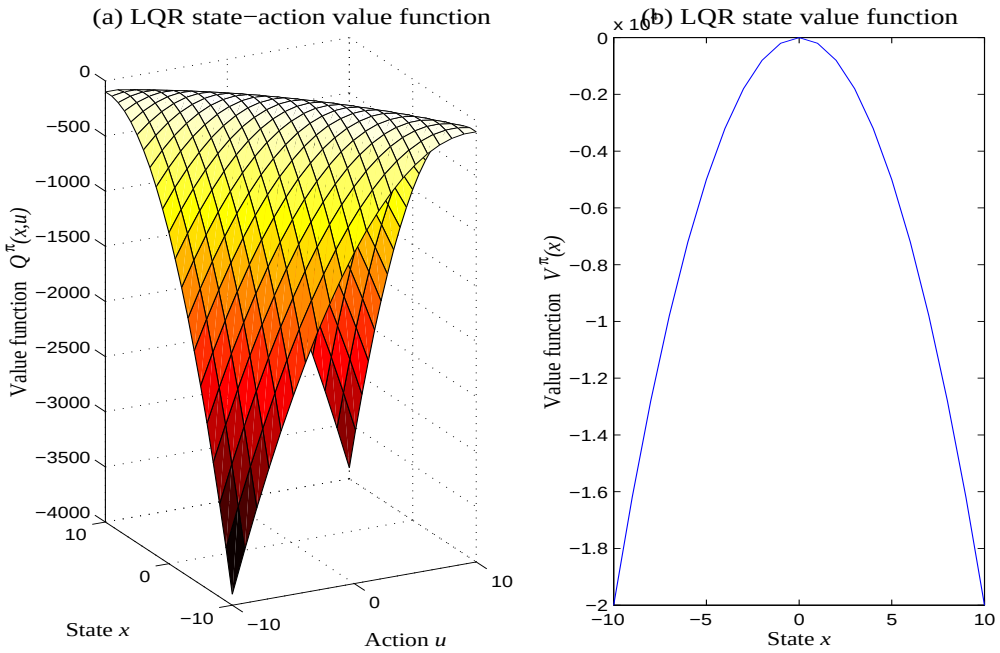


Figure 2.4: This figures shows both (a) the state-action value function $Q^{\pi_\theta}(x,u)$ and (b) the state value function $V^{\pi_\theta}(x)$ of an one-dimensional linear quadratic control problem with a Gaussian policy. The parameters of the system are $A = b = R = Q = 1$, $\gamma = 0.95$, and the parameters of the Gaussian policy $\pi_{\boldsymbol{\theta}}(u|x)$ are $\boldsymbol{\theta} = [k, \sigma]^T$ with $k = -0.5$, and $\sigma = 0.1$.

for the start-state case. $\mathbf{P}$ *and* $\mathbf{P}_\gamma$ *are defined as in Example 3, page 11 in both cases. For a proof of both cases, see in the appendix, Section A.1.2, page 47. The value functions for discounted start-state case LQR are plotted in Figure 2.4 (a-b).*

Bellman and Poisson equations

Due to the assumption, that our process and our policy are both Markovian, the value functions of both cases have a defining relation between the values at different states. This relation can be expressed in form of the **Poisson equation** for the average reward case, and in form of the **Bellman equation** for the discounted start-state case. These equations are given in the definition below.

Definition 3 (Poisson- & Bellman equations) *The* ***Poisson equation*** *states, that*

$$V^{\pi_\theta}(\mathbf{x}) = \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \left(r(\mathbf{x},\mathbf{u}) - J(\pi_{\boldsymbol{\theta}}) + \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u},\mathbf{x}) V(\mathbf{x}') d\mathbf{x}' \right) d\mathbf{u}, \tag{2.7}$$

$$Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) = r(\mathbf{x},\mathbf{u}) - J(\pi_{\boldsymbol{\theta}}) + \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u},\mathbf{x}) V(\mathbf{x}') d\mathbf{x}', \tag{2.8}$$

for the average reward formulation, and the ***Bellman equation*** *states that*

$$V^{\pi_\theta}(\mathbf{x}) = \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \left(r(\mathbf{x},\mathbf{u}) + \gamma \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u},\mathbf{x}) V(\mathbf{x}') d\mathbf{x}' \right) d\mathbf{u}, \tag{2.9}$$

$$Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) = r(\mathbf{x},\mathbf{u}) + \gamma \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u},\mathbf{x}) V(\mathbf{x}') d\mathbf{x}', \tag{2.10}$$

for the discounted formulation.

Please note that this definition is extended to the continuous case. This is not common in the literature, but futile once we introduce function approximation or handle continuous state-action tasks such as robot control or linear quadratic regulation.

Again, we can visualize these equations using a simple example which is limited to discrete states, and discrete actions. The general case with continuous actions is not straightforward as only few techniques exist for solving integral equations such as the Bellman or Poisson equation. In linear quadratic regulation (LQR) problems, the Bellman equations can only be solved due to the assumption that $V^{\pi_\theta}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$, and subsequently showing that the integral equation has a solution given these value functions.

Discrete Bellman equation

Example 6 (Discrete state and action spaces) *Similarly, we can easily determine value function for discrete problems using the Bellman equations*

$$V_i = \sum_{k=1}^{n} \pi_{\boldsymbol{\theta}}(u_k | x_i) \left(\mathcal{R}_i^k + \gamma \sum_{j=1}^{n} \mathcal{P}_{ij}^k V_j \right) = R_i + \gamma \sum_{j=1}^{n} P_{ij} V_j,$$

using the definitions from Example 4, page 13, and denoting $\mathbf{V} = [V_i] \in \mathbb{R}^n$ *with* $V_i = V^{\pi_\theta}(x_i)$*. As we have a vector-matrix equation* $\mathbf{V} = \mathbf{R} + \gamma\mathbf{P}\mathbf{V}$*, we can solve it directly and obtain* $\mathbf{V} = (\mathbf{I} - \gamma\mathbf{P})^{-1}\mathbf{R}$*, see (Russel & Norvig, 1995). Alternatively, we can use the policy evaluation algorithm in order to obtain the state values of discrete states (Sutton & Barto, 1998). The same can be done for the average reward case when knowing the expected reward.*

Advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$

Apart from the previously described value functions, we still have one further value function, i.e., the **advantage function** $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ as presented in (Baird, 1993). Other researchers call this function the Bellman error function (Jaakkola et al., 1994), and we could think of it also as the expected TD(0) error of one particular action $\mathbf{u}$ taken in state $\mathbf{x}$.

Definition 4 (Advantage function) *The* ***advantage function*** $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ *of a policy* $\pi_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$ *is given by*

$$A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) - V^{\pi_\theta}(\mathbf{x}). \tag{2.11}$$

Its Bellman equation for the discounted case is given by

$$V^{\pi_\theta}(\mathbf{x}) + A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = r(\mathbf{x}, \mathbf{u}) + \gamma \int_{\mathbb{X}} p(\mathbf{x}' | \mathbf{u}, \mathbf{x}) V(\mathbf{x}') d\mathbf{x}', \tag{2.12}$$

and its Poisson equation for the average reward case by

$$V^{\pi_\theta}(\mathbf{x}) + A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = r(\mathbf{x}, \mathbf{u}) - J(\pi_{\boldsymbol{\theta}}) + \int_{\mathbb{X}} p(\mathbf{x}' | \mathbf{u}, \mathbf{x}) V(\mathbf{x}') d\mathbf{x}'. \tag{2.13}$$

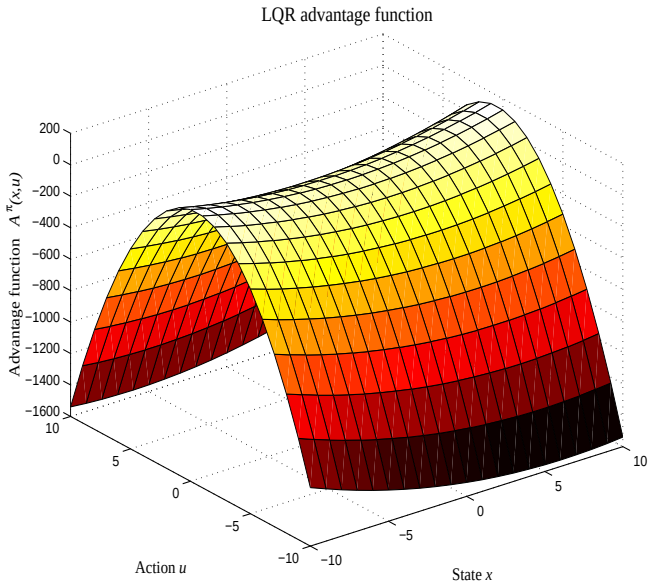


Figure 2.5: This figures shows the advantage function $A^{\pi_\theta}(x,u)$ of an one-dimensional linear quadratic control problem with a Gaussian policy. Note that this function is no longer convex like the previous value functions. The parameters of the system are again $A = B = R = Q = 1$, $\gamma = 0.95$, and the parameters of the Gaussian policy $\pi_{\boldsymbol{\theta}}(u|x)$ are $\boldsymbol{\theta} = [k, \sigma]^T$ with $k = -0.5$, and $\sigma = 0.1$.

The advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ contains all information which is necessary to decide which is the optimal action since $\mathbf{u} = \text{argmax}_{\mathbf{u}^* \in \mathbb{U}} Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}^*) = \text{argmax}_{\mathbf{u}^* \in \mathbb{U}} A^{\pi_\theta}(\mathbf{x}, \mathbf{u}^*)$. Unlike the state value function, and the state-action value function, the advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ has no state dependent offset as it is mean-zero $\int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) d\mathbf{u} = 0$. Therefore the advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ is not a potential function unlike the state value function $V^{\pi_\theta}(\mathbf{x})$, and the state-action value $Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ function. In order to get a closer look, let us discuss the advantage function of our linear quadratic regulation example with a Gaussian policy.

Example 7 (Linear quadratic control) *The advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$ for LQR with a Gaussian policy $\pi_{\boldsymbol{\theta}}$ as in Example 1, page 9, is given by* *LQR advantage function*

$$A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = -\frac{1}{2} \begin{bmatrix} \mathbf{x}^T, & u \end{bmatrix} \mathbf{H} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \frac{1}{2} \left(R + \mathbf{b}^T \mathbf{P} \mathbf{b} \right) \sigma^2,$$

with

$$\mathbf{H} = \begin{bmatrix} -2\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \mathbf{k}\left(\mathbf{b}^T\mathbf{P}\mathbf{b} + R\right)\mathbf{k}^T & \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix},$$

for the average reward case. For the discounted start-state it is given by

$$A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = -\frac{1}{2} \begin{bmatrix} \mathbf{x}^T, & u \end{bmatrix} \mathbf{H} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} + \frac{1}{2} \left(R + \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{b} \right) \sigma^2,$$

with

$$\mathbf{H} = \begin{bmatrix} -2\gamma\mathbf{k}\mathbf{b}^T\mathbf{P}_\gamma\mathbf{A} - \gamma\mathbf{k}\left(\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b} + R\right)\mathbf{k}^T & \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{b} \\ \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{A} & R + \gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b} \end{bmatrix}.$$

Note, that both definition do not differ except for the constant γ, and become the same for $\gamma \to 1$ since $\lim_{\gamma \to 1} \mathbf{P}_\gamma = \mathbf{P}$. Therefore the advantage function is a more general value function than the previously used ones. The discounted start-state value function for our example problem is plotted in Figure 2.5. For a derivation see appendix, Section A.1.3, page 51.

2.3 Trajectories, Transition Kernels and Stationary Distributions

As we are discussing Markov Chains, we can use a variety of statistical tools which long have been neglected in the area of reinforcement learning: transition kernels, and stationary distributions. Let us now assume that we are interested in sequences of states and actions of length n starting at an initial state-action pair $\mathbf{x}_0$, $\mathbf{u}_0$, and ending at state $\mathbf{x}_n$. The space of all such sequences of length n, we will refer to them as trajectories, is defined by $\mathbb{T}_n = (\mathbb{X}, \mathbb{U})^n \times \mathbb{X}$. A single trajectory in $\mathcal{T}^n \in \mathbb{T}_n$ can be defined as $\mathcal{T}^n = [\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \ldots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \mathbf{x}_n]$. The probability $p(\mathcal{T}^n | \mathbf{x}_0)$ of such a sequence $\mathcal{T}^n \in \mathbb{T}_n$ given the start state $\mathbf{x}_0$ is obvious from our problem statement

Probability of a trajectory

$$p(\mathcal{T}^n | \mathbf{x}_0) = \prod_{t=0}^{n-1} p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) \pi(\mathbf{u}_t | \mathbf{x}_t). \tag{2.14}$$

Furthermore, the probability $p(\mathcal{T}^n)$ of the whole trajectory $\mathcal{T}^n$ as a whole is defined by $p(\mathcal{T}^n) = p(\mathcal{T}^n | \mathbf{x}_1) p(\mathbf{x}_1)$. Let us now consider the case that we are just interested in the probability $p(\mathbf{x}_n | \mathbf{x}_1, n) = K^n_{\pi_\theta}(\mathbf{x}_1, \mathbf{x}_n)$ of reaching state $\mathbf{x}_n$ given that we started in state $\mathbf{x}_1$ in exactly n steps. This probability can be calculated by integrating out all actions, and intermediary states $\tilde{\mathcal{T}}^n = [\mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \ldots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}] \in \mathbb{U} \times (\mathbb{X}, \mathbb{U})^{n-1}$. We then can calculate the probability of going from $\mathbf{x}_0$ to $\mathbf{x}_n$ in exactly n steps, i.e.,

Transition kernel interpretation

$$\begin{aligned} K^n_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_n) &= \int_{\mathbb{T}_n} p(\mathcal{T}^n | \mathbf{x}_0) \, d\tilde{\mathcal{T}}^n, \\ &= \int_{\mathbb{U}} \int_{\mathbb{X}} \int_{\mathbb{U}} \cdots \int_{\mathbb{X}} \int_{\mathbb{U}} \prod_{t=0}^{n-1} p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) \pi(\mathbf{u}_t | \mathbf{x}_t) \, d\mathbf{u}_0 d\mathbf{x}_1 d\mathbf{u}_1 \ldots d\mathbf{x}_{n-1} d\mathbf{u}_{n-1}. \end{aligned}$$

From the equation above, we see that we can define this term recursively by integrating out a single variable at a time. In order to do this we have to isolate variables and replace terms. By reordering the variables, we get

$$\begin{aligned} K^n_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_n) &= \int_{\mathbb{X}} \cdots \int_{\mathbb{X}} \prod_{t=0}^{n-1} \underbrace{\left(\int_{\mathbb{U}} p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) \pi(\mathbf{u}_t | \mathbf{x}_t) \, d\mathbf{u}_t \right)}_{K_{\pi_\theta}(\mathbf{x}_t, \mathbf{x}_{t+1})} d\mathbf{x}_1 \ldots d\mathbf{x}_{n-1}, \\ &= \int_{\mathbb{X}} \cdots \int_{\mathbb{X}} \prod_{t=0}^{n-1} K_{\pi_\theta}(\mathbf{x}_t, \mathbf{x}_{t+1}) \, d\mathbf{x}_1 \ldots d\mathbf{x}_{n-1}. \end{aligned}$$

We define the term in the parentheses to be the transition kernel $K_{\pi_\theta}(\mathbf{x}_t, \mathbf{x}_{t+1})$ of the Markov process. If we further reorder the equations, we get

$$K^n_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_n) = \int_{\mathbb{X}} \dots \underbrace{\int_{\mathbb{X}} \underbrace{\int_{\mathbb{X}} K_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_1) K_{\pi_\theta}(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_1}_{K^2_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_2)} K_{\pi_\theta}(\mathbf{x}_2, \mathbf{x}_3) d\mathbf{x}_2}_{K^3_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_3)} \dots \underbrace{\qquad\qquad}_{K^n_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}_n)} K_{\pi_\theta}(\mathbf{x}_{n-1}, \mathbf{x}_n) d\mathbf{x}_{n-1}.$$

This gives us a recursive definition of the n-step transition kernels

Transition kernel definition

$$K^{n+1}_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{X}} K^{n-1}_{\pi_\theta}(\mathbf{x}, \mathbf{x}'') K_{\pi_\theta}(\mathbf{x}'', \mathbf{x}) d\mathbf{x}'',$$

with $K^1_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = K_{\pi_\theta}(\mathbf{x}, \mathbf{x}')$.

Definition 5 (Transition kernels) *Given a policy $\pi_\theta(\mathbf{u}|\mathbf{x})$, and the system transition probabilities $p(\mathbf{x}'|\mathbf{x}, u)$, we can define a* ***transition kernel*** *$K_{\pi_\theta}(x, x')$ as*

$$K_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{U}} p(\mathbf{x}'|\mathbf{x}, \mathbf{u}) \pi_\theta(\mathbf{u}|\mathbf{x}) d\mathbf{u}. \tag{2.15}$$

*The n-**step transition kernels** is given by*

$$K^{n+1}_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{X}} K^{n-1}_{\pi_\theta}(\mathbf{x}, \mathbf{x}'') K_{\pi_\theta}(\mathbf{x}'', \mathbf{x}') d\mathbf{x}'', \tag{2.16}$$

with $K^1_{\pi_\theta}(\mathbf{x}, \mathbf{x}'') = K_{\pi_\theta}(\mathbf{x}, \mathbf{x}'')$.

Having these definitions, we have reduced our **controlled Markov decision problem** to an **uncontrolled Markov chain**. For uncontrolled Markov chains, a variety of nice properties are given. We will list the most important ones here without attempting to give a complete list.

Komolgorov-Chapman equation. In order to get from one state $\mathbf{x}$ to another state $\mathbf{x}'$ in $n+m$ steps, the agent has to pass through an intermediary state $\mathbf{x}''$. This gives us the Komolgorov-Chapman equation

Komolgorov-Chapman equation

$$K^{n+m}_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{X}} K^m_{\pi_\theta}(\mathbf{x}, \mathbf{x}'') K^n_{\pi_\theta}(\mathbf{x}'', \mathbf{x}') d\mathbf{x}''.$$

This appears trivial from our previous discussion, and a proof for it is given in (Berger & Casella, 2002).

Stationary distribution or invariant measure. For a stable Markov chain, a stationary distribution or an invariant measure $\nu^{\pi_\theta}(\mathbf{x})$ exists for which the equation

Stationary distribution

$$\nu^{\pi_\theta}(\mathbf{x}) = \int_{\mathbb{X}} K_{\pi_\theta}(\mathbf{x}, \mathbf{x}') \nu^{\pi_\theta}(\mathbf{x}') d\mathbf{x}'$$

holds (Berger & Casella, 2002).

Convergence to stationary distribution

Law of W. Doeblin. If a Markov chain is stable, irreducible, recurrent and aperiodic, we have

$$\nu^{\pi_\theta}(\mathbf{x}) = \lim_{n\to\infty} \int_{\mathbb{X}} K^n_{\pi_\theta}(\mathbf{x}, \mathbf{x}') \, p(\mathbf{x}') \, d\mathbf{x}',$$

independent of the start-state distribution $p(\mathbf{x}')$. See (Grimmett & Stirzaker, 2001) for details.

Resolvent kernel

Resolvent kernel. Associated with this kernel, we have a **resolvent kernel** $K_{\gamma\pi_\theta}(\mathbf{x}, \mathbf{x}')$ given by

$$K_{\gamma\pi_\theta}(\mathbf{x}, \mathbf{x}') = (1-\gamma) \sum_{n=0}^{\infty} \gamma^n K^n_{\pi_\theta}(\mathbf{x}, \mathbf{x}').$$

Obviously, a stationary distribution $\nu^{\pi_\theta}_\gamma(\mathbf{x})$ can also (does always?) exist for this kernel (Robert & Casella, 1999). These two kernels are connected by $\sum_{n=0}^{\infty} K^n_{\pi_\theta}(\mathbf{x}, \mathbf{x}') = \frac{\gamma}{1-\gamma} \sum_{n=0}^{\infty} K^n_{\gamma\pi_\theta}(\mathbf{x}, \mathbf{x}')$, if a stationary distribution $\nu^{\pi_\theta}(\mathbf{x})$ exists (Berger & Casella, 2002).

Stationary distributions are a complicated topic as they do not exist for every Markov Chain (e.g., most importantly, they do not exist for physical control problems if the initial policy is instable). Furthermore they are hard to infer if the state space is continuous.

Average state distribution

Therefore we introduce **average state distribution** $d^{\pi_\theta}(\mathbf{x})$ which becomes the stationary distribution if it exists.

Definition 6 (Average state distribution) *The average state distribution is given by*

$$d^{\pi_\theta}(\mathbf{x}) = \lim_{n\to\infty} \frac{1}{n} \int_{\mathbb{X}} p(\mathbf{x}_0) \sum_{t=0}^{n} K^t_{\pi_\theta}(\mathbf{x}_0, \mathbf{x}) \, d\mathbf{x}_0, \tag{2.17}$$

where $p(\mathbf{x}_0)$ denotes the start-state distribution. If the conditions of the law of W. Doeblin are fulfilled, it becomes the stationary distribution, i.e., $d^{\pi_\theta}(\mathbf{x}) = \nu^{\pi_\theta}(\mathbf{x})$.

The discounted average state distribution is given by

$$d^{\pi_\theta}_\gamma(\mathbf{x}) = \int_{\mathbb{X}} p(\mathbf{x}_0) \frac{K_{\gamma\pi_\theta}(\mathbf{x}_0, \mathbf{x})}{1-\gamma} d\mathbf{x}_0. \tag{2.18}$$

In the policy gradient literature $d^{\pi_\theta}(\mathbf{x})$, and $d^{\pi_\theta}_\gamma(\mathbf{x})$ are often referred to as stationary distributions. For $d^{\pi_\theta}(\mathbf{x})$, and $d^{\pi_\theta}_\gamma(\mathbf{x})$, this is not a necessary but a sufficient condition. This makes an essential difference as a stationary distribution cannot exits for an unstable task. However, many tasks learned by humans, e.g., standing, walking, grasping, etc., are initially unstable. Therefore, we could not apply the theorems based on the stationary distribution in such tasks. Using our average state distribution instead, these tasks generalize.

Let us now close this section after another example which we use to bring light into the complicated story. In this Example 8 we analytically discuss the stationary distribution $\nu^{\pi_\theta}(x_i)$, and the average state distribution for the discounted case $d^{\pi_\theta}_\gamma(x_i)$ of the discrete state and action.

Example 8 (Discrete state and action problems) *(a) Clearly in Example 2 (page 10), and Example 4 (page 13), we are already given the kernel of a discrete state and action problem, i.e.,* $K_{\pi_\theta}(x_i, x_j) = P_{ij}$*. In vector-matrix notation, we can now rewrite the definition of a stationary distribution to* $\boldsymbol{\nu} = \mathbf{P}\boldsymbol{\nu}$ *with* $\boldsymbol{\nu} = [\nu_i] \in \mathbb{R}^n$*. When solving for* $\boldsymbol{\nu}$*, we get the equation* $(\mathbf{I} - \mathbf{P})\boldsymbol{\nu} = \mathbf{0}$*. We realize, that* $\boldsymbol{\nu}$ *is an eigenvector of the transition matrix* $\mathbf{P}$*. Furthermore, we see that if the law of W. Doeblin is fulfilled, we get* $\boldsymbol{\nu} = \lim_{n\to\infty} \mathbf{P}^n \mathbf{S}$*, i.e., a fast numerical approximation of the stationary distribution* $\boldsymbol{\nu}$*.*

(b) In case, that a stationary distribution exists, we obviously have $d^{\pi_\theta}(x_i) = \nu_i$*. If it does not exist, we still get a not stationary average state distribution* $d^{\pi_\theta}(x_i) = d_i$ *by summing up* $\mathbf{d} = \frac{1}{N}\sum_{t=0}^{N} \mathbf{P}^t \mathbf{S}$*, where* $\mathbf{d} = [d_i] \in \mathbb{R}^n$*. We will later see that we can also use this measure for policy gradient methods where a stationary distribution does not exist.*

(c) The second revelation comes from looking at Example 4 again. We see that we used the resolvent of the transition kernel already in there. It is given by $K_{\gamma\pi_\theta}(x_i, x_j) = K_{ij}$ *with* $\mathbf{K} = [K_{ij}] \in \mathbb{R}^{n\times n}$*, and* $\mathbf{K} = (1-\gamma)\sum_{t=0}^{N} \gamma^t \mathbf{P}^t$*. It is clear* $\mathbf{d}^\gamma = \mathbf{K}\mathbf{S}/(1-\gamma) = [d_i^\gamma] \in \mathbb{R}^n$*, and* $d_\gamma^{\pi_\theta}(x_i) = d_i^\gamma$*.*

What is the meaning of average state distributions for our applications? It has major importance. Let us assume that we have a temporal sequence $\mathcal{T}^n \in \mathbb{T}_n$ of length n given by $\mathcal{T}^n = [\mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \mathbf{x}_3, \mathbf{u}_3, \ldots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \mathbf{x}_n]$, and a function

$$f(\mathcal{T}^n) = \sum_{t=0}^{n-1} f(\mathbf{x}_t, \mathbf{u}_t).$$

This trajectory has the same probability $p(\mathcal{T}^n | \mathbf{x}_0)$ as stated in equation 2.14. We intend to evaluate the expectation $E_{\mathcal{T}^n}\{f(\mathcal{T}^n)\} = \int_{\mathbb{T}_n} p(\mathcal{T}^n | \mathbf{x}_0) f(\mathcal{T}^n) d\mathcal{T}^n$ of this function for such a sequence of length n. We get

$$\begin{aligned}
E_{\mathcal{T}^n}\{f(\mathcal{T}^n | \mathbf{x}_0)\} &= \int_{\mathbb{T}_n} p(\mathcal{T}^n | \mathbf{x}_0) f(\mathcal{T}^n) d\mathcal{T}^n, \\
&= \int_{\mathbb{T}_n} \sum_{t=0}^{n-1} p(\mathcal{T}^n | \mathbf{x}_0) f(\mathbf{x}_t, \mathbf{u}_t) d\mathcal{T}^n, \\
&= \int_{\mathbb{X}} \sum_{t=0}^{n-1} K_{\pi_\theta}^t(\mathbf{x}_0, \mathbf{x}) \int_{\mathbb{U}} \pi(\mathbf{u} | \mathbf{x}) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}.
\end{aligned}$$

When obtaining the average for all trajectories from all starting states with infinite length, we can derive for the average reward case $f(\mathcal{T}^n) = \frac{1}{n}\sum_{t=0}^{n-1} f(\mathbf{x}_t, \mathbf{u}_t)$, that

Averages in trajectory and sample view

$$\begin{aligned}
\lim_{n\to\infty} E_{\mathcal{T}}\left\{\frac{1}{n}\sum_{t=0}^{n-1} f(\mathbf{x}_t, \mathbf{u}_t)\right\} &= \lim_{n\to\infty} \frac{1}{n}\int_{\mathbb{X}} p(\mathbf{x}_0) E_{\mathcal{T}^n}\{f(\mathcal{T}^n | \mathbf{x}_0)\} d\mathbf{x}_0, \\
&= \lim_{n\to\infty} \frac{1}{n}\int_{\mathbb{X}} p(\mathbf{x}_0) \int_{\mathbb{X}} \sum_{t=0}^{n} K_{\pi_\theta}^t(\mathbf{x}_0, \mathbf{x}) \int_{\mathbb{U}} \pi(\mathbf{u} | \mathbf{x}) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x} d\mathbf{x}_0, \\
&= \lim_{n\to\infty} \frac{1}{n}\int_{\mathbb{X}} p(\mathbf{x}_0) \int_{\mathbb{X}} \sum_{t=0}^{n} K_{\pi_\theta}^t(\mathbf{x}_0, \mathbf{x}) d\mathbf{x}_0 \int_{\mathbb{U}} \pi(\mathbf{u} | \mathbf{x}) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}, \\
&= \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi(\mathbf{u} | \mathbf{x}) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}, \\
&= E_{d^{\pi_\theta}, \pi_\theta}\{f(\mathbf{x}, \mathbf{u})\}.
\end{aligned}$$

For the discounted case, i.e., $f\left(\mathcal{T}^n\right) = \sum_{t=0}^{n-1} \gamma^t f(\mathbf{x}_t, \mathbf{u}_t)$, we get

$$\begin{aligned}
E_{\mathcal{T}}\left\{\sum_{t=0}^{\infty} \gamma^t f(\mathbf{x}_t, \mathbf{u}_t)\right\} &= \lim_{n\to\infty} \int_{\mathbb{X}} p(\mathbf{x}_0) E_{\mathcal{T}^n}\left\{f\left(\mathcal{T}^n \left|\mathbf{x}_1\right.\right)\right\} d\mathbf{x}_1, \\
&= \lim_{n\to\infty} \int_{\mathbb{X}} p(\mathbf{x}_0) \int_{\mathbb{X}} \sum_{t=0}^{n} \gamma^t K_{\pi_\theta}^t\left(\mathbf{x}_0, \mathbf{x}\right) \int_{\mathbb{U}} \pi\left(\mathbf{u}\left|\mathbf{x}\right.\right) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x} d\mathbf{x}_0, \\
&= \int_{\mathbb{X}} p(\mathbf{x}_0) \int_{\mathbb{X}} \frac{K_{\gamma\pi_\theta}\left(\mathbf{x}_0, \mathbf{x}\right)}{1-\gamma} d\mathbf{x}_0 \int_{\mathbb{U}} \pi\left(\mathbf{u}\left|\mathbf{x}\right.\right) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}, \\
&= \int_{\mathbb{X}} d_\gamma^{\pi_\theta}\left(\mathbf{x}\right) \int_{\mathbb{U}} \pi\left(\mathbf{u}\left|\mathbf{x}\right.\right) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}, \\
&= E_{d^{\pi_\theta}, \pi_\theta, \gamma}\left\{f(\mathbf{x}, \mathbf{u})\right\}.
\end{aligned}$$

We can conclude this in the sequence summation theorem.

Theorem 1 (Sequence Summation Theorem) *For the average reward case, the relation between trajectories, and samples for a function* $f\left(\mathcal{T}^n\right) = \frac{1}{n}\sum_{t=0}^{n-1} f(\mathbf{x}_t, \mathbf{u}_t)$ *is given by*

$$\lim_{n\to\infty} E_{\mathcal{T}}\left\{\frac{1}{n}\sum_{t=0}^{n-1} f(\mathbf{x}_t, \mathbf{u}_t)\right\} = \int_{\mathbb{X}} d^{\pi_\theta}\left(\mathbf{x}\right) \int_{\mathbb{U}} \pi\left(\mathbf{u}\left|\mathbf{x}\right.\right) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}. \tag{2.19}$$

For the start-state case, with a function $f\left(\mathcal{T}^n\right) = \sum_{t=0}^{n-1} \gamma^t f(\mathbf{x}_t, \mathbf{u}_t)$*, we have*

$$E_{\mathcal{T}}\left\{\sum_{t=0}^{\infty} \gamma^t f(\mathbf{x}_t, \mathbf{u}_t)\right\} = \int_{\mathbb{X}} d_\gamma^{\pi_\theta}\left(\mathbf{x}\right) \int_{\mathbb{U}} \pi\left(\mathbf{u}\left|\mathbf{x}\right.\right) f(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x}. \tag{2.20}$$

This theorem becomes already quite practical in order to calculate the expected return as well as to derive the Fisher information matrix for natural gradient learning. The transitions kernel view in general allows us to bring reinforcement learning to a higher degree of theoretical foundation. We will see both in the next section.

2.4 Kernel-based Expected Return

Let us now return to the expected return, and apply our newly derived knowledge here. The expected return is then given by the theorem below.

Theorem 2 (Expected Return) *The expected return is given by*

$$J(\pi_\theta) = \int_{\mathbb{X}} d^{\pi_\theta}\left(\mathbf{x}\right) \int_{\mathbb{U}} \pi_\theta\left(\mathbf{u}\left|\mathbf{x}\right.\right) r\left(\mathbf{x}, \mathbf{u}\right) d\mathbf{u} d\mathbf{x}, \tag{2.21}$$

both in average reward, and start-state formulation.

Proof. The proof is rather simple, we just set $f(\mathbf{x}, \mathbf{u}) = r(\mathbf{x}, \mathbf{u})$, and apply the sequence summation theorem. We directly get this result, and therefore have obtained the expected return. ■

Clearly, we now have a sample-based representation of the average reward. If a stationary distribution $\nu^{\pi_\theta}(\mathbf{x})$ exists, and the start-state distribution $p(\mathbf{x}_0)$ equals the stationary distribution, i.e., $p(\mathbf{x}_0) = \nu^{\pi_\theta}(\mathbf{x})$, we can show the equivalance of both the average reward and the discounted reward case:

Lemma 1 (Connection between both formulations) *The expected return of the average reward case is connected to the discounted start-state case*

$$J(\pi_\theta) = (1-\gamma) J_\gamma(\pi_\theta | \mathbb{X}, \nu^{\pi_\theta}(\mathbf{x})), \tag{2.22}$$

if a stationary distribution $\nu^{\pi_\theta}(\mathbf{x})$ exists, and the start-state distribution $p(\mathbf{x}_0)$ equals the stationary distribution, i.e., $p(\mathbf{x}_0) = \nu^{\pi_\theta}(\mathbf{x})$.

Proof. We have by definition

$$\begin{aligned} J_\gamma(\pi_\theta | \mathbb{X}, \nu^{\pi_\theta}(\mathbf{x})) &= \int_{\mathbb{X}} \nu^{\pi_\theta}(\mathbf{x}) V^{\pi_\theta}(\mathbf{x}) d\mathbf{x}, \\ &= \int_{\mathbb{X}} \nu^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_\theta(\mathbf{u}|\mathbf{x}) \left(r(\mathbf{x}, \mathbf{u}) + \gamma \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u}, \mathbf{x}) V^{\pi_\theta}(\mathbf{x}') d\mathbf{x}' \right) d\mathbf{u}\, d\mathbf{x}. \end{aligned}$$

Since $d^{\pi_\theta}(\mathbf{x}) = \nu^{\pi_\theta}(\mathbf{x})$, we see that

$$\begin{aligned} J_\gamma(\pi_\theta | \mathbb{X}, \nu^{\pi_\theta}(\mathbf{x})) &= \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_\theta(\mathbf{u}|\mathbf{x}) r(\mathbf{x}, \mathbf{u}) d\mathbf{u}\, d\mathbf{x} \\ &\quad + \gamma \int_{\mathbb{X}} \nu^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_\theta(\mathbf{u}|\mathbf{x}) \int_{\mathbb{X}} p(\mathbf{x}'|\mathbf{u}, \mathbf{x}) V^{\pi_\theta}(\mathbf{x}') d\mathbf{x}' d\mathbf{u}\, d\mathbf{x}, \\ &= J(\pi_\theta) + \gamma \int_{\mathbb{X}} \nu^{\pi_\theta}(\mathbf{x}') V^{\pi_\theta}(\mathbf{x}') d\mathbf{x}', \\ &= J(\pi_\theta) + \gamma J_\gamma(\pi_\theta | \mathbb{X}, \nu^{\pi_\theta}(\mathbf{x})). \end{aligned}$$

We can solve this and get

$$J(\pi_\theta) = (1-\gamma) J_\gamma(\pi_\theta | \mathbb{X}, \nu^{\pi_\theta}(\mathbf{x})),$$

which clearly proves our theorem. ■

This theorem was first presented by Jaakola, Singh & Jordan (1994) in a slighly different context. Baxter & Bartlett (2000) draw the slighly incorrect solution from it that this would be the case for arbitrary start-state distributions. We can see from Example 3, and Example 9, that this extension is not true.

Connection for $\gamma \to 1$

Lemma 1 allows us to draw an interesting conclusion: since the start-state distribution does not matter for $\gamma \to 1$, we immediately see from Lemma 1, that

$$J(\pi_\theta) = \lim_{\gamma \to 1} (1-\gamma) J_\gamma(\pi_\theta | \mathbb{X}_0), \tag{2.23}$$

for arbitrary start-state spaces and distributions if a stationary distribution $\nu^{\pi_\theta}(\mathbf{x})$ exists. The following example shows how that works for LQR.

LQR expected return connection

Example 9 (Linear quadratic control) *We study the expected return of linear quadratic regulation problems as in Example 3, page 11, with the same Gaussian policy π_θ. In this*

case, a stationary distribution $\nu^{\pi_\theta}(\mathbf{x})$ exists if and only if the system is stable. For a stable LQR system, we have $\lim_{\gamma\to 1}\mathbf{P}_\gamma = \mathbf{P} < \infty$. We can insert the expected return from Example 3 into equation(2.23), and get

$$\begin{aligned}
&\lim_{\gamma\to 1}(1-\gamma)\,J_\gamma\left(\pi_\theta\,|\{\mathbf{x}_0\}\right),\\
&=\lim_{\gamma\to 1}(1-\gamma)\left(-\frac{1}{2}\mathbf{x}_0^T\mathbf{P}_\gamma\mathbf{x}_0-\frac{1}{2}\frac{1}{1-\gamma}\left(R+\gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2\right),\\
&=\lim_{\gamma\to 1}\left(-(1-\gamma)\,\frac{1}{2}\mathbf{x}_0^T\mathbf{P}_\gamma\mathbf{x}_0-\frac{1}{2}\left(R+\gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2\right),\\
&=-\frac{1}{2}\left(R+\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2,\\
&=J(\pi_\theta).
\end{aligned}$$

This is exactly the solution we obtained in Example 3.

It is clear from Example 9 that the relation does not hold not for all $\gamma\in[0,1)$.

2.5 Analytical solution for the value function

Similarly interesting, we can also get a transition kernel-based view on the value function for the discounted case. We can reformulate the Bellman equation by

$$\begin{aligned}
V^{\pi_\theta}(\mathbf{x}) &= \int_{\mathbb{U}}\pi_\theta(\mathbf{u}\,|\mathbf{x})\left(r(\mathbf{x},\mathbf{u})+\gamma\int_{\mathbb{X}}p\left(\mathbf{x}'\,|\mathbf{u},\mathbf{x}\right)V^{\pi_\theta}(\mathbf{x}')d\mathbf{x}'\right)d\mathbf{u},\\
&= \bar{r}^{\pi_\theta}(\mathbf{x})+\gamma\int_{\mathbb{X}}K_{\pi_\theta}\left(\mathbf{x},\mathbf{x}'\right)V^{\pi_\theta}(\mathbf{x}')d\mathbf{x}',
\end{aligned}$$

by defining the state average reward $\bar{r}^{\pi_\theta}(\mathbf{x})=\int_{\mathbb{U}}\pi_\theta(\mathbf{u}\,|\mathbf{x})\,r(\mathbf{x},\mathbf{u})d\mathbf{u}$. This is an Fredholm integral equation of the second kind. In many cases, these equations cannot be solved. However, for the case

$$|\gamma|<\frac{1}{\sqrt{\int_{\mathbb{X}}\int_{\mathbb{X}}K_{\pi_\theta}(\mathbf{x},\mathbf{x}')\,d\mathbf{x}'d\mathbf{x}}},$$

the solution is given by a Neumann series (Bronstein, Semendjajew, Musiol, & Mühlig, 1995). Since $K_{\pi_\theta}(\mathbf{x},\mathbf{x}')$ is a probability, we have $\int_{\mathbb{X}}\int_{\mathbb{X}}K_{\pi_\theta}(\mathbf{x},\mathbf{x}')\,d\mathbf{x}'d\mathbf{x}=1$, and know that this method can be applied if $\gamma<1$. The solution is

Analytical solution of the Bellman equation

$$\begin{aligned}
V^{\pi_\theta}(\mathbf{x}) &= \int_{\mathbb{X}}\sum_{t=0}^{\infty}\gamma^t K_{\pi_\theta}^t\left(\mathbf{x},\mathbf{x}'\right)\bar{r}^{\pi_\theta}(\mathbf{x})d\mathbf{x}',\\
&= \int_{\mathbb{X}}\sum_{t=0}^{\infty}\gamma^t K_{\pi_\theta}^t\left(\mathbf{x},\mathbf{x}'\right)\int_{\mathbb{U}}\pi_\theta\left(\mathbf{u}\,|\mathbf{x}'\right)r(\mathbf{x}',\mathbf{u})d\mathbf{u}d\mathbf{x}'.\\
&= \int_{\mathbb{X}}\frac{K_{\gamma\pi_\theta}(\mathbf{x},\mathbf{x}')}{1-\gamma}\int_{\mathbb{U}}\pi_\theta\left(\mathbf{u}\,|\mathbf{x}'\right)r(\mathbf{x}',\mathbf{u})d\mathbf{u}d\mathbf{x}'.
\end{aligned}$$

We therefore know that $V^{\pi_\theta}(\mathbf{x})$ does always exist, and can always be approximated in the discounted case. This solution fits well into the sequence summation theorem. Furthermore, we can derive the same for the average reward case from the Poisson equation if $\bar{r}^{\pi_\theta}(\mathbf{x})-J(\pi_\theta)$ is bounded by a real number.

We also see that if we learn models for $p(\mathbf{x}'\,|\mathbf{u},\mathbf{x})$, and $r(\mathbf{x}',\mathbf{u})$, we can analytically determine the value function $V^{\pi_\theta}(\mathbf{x})$, and do not need to learn it. However, determining

all integrals is a non-trivial task for non-discrete problems. Let us assume, we have the analytical models of $\tilde{p}\left(\mathbf{x}'\left|\mathbf{u},\mathbf{x}\right.\right) \approx p\left(\mathbf{x}'\left|\mathbf{u},\mathbf{x}\right.\right)$, and $\tilde{r}(\mathbf{x}',\mathbf{u}) \approx r(\mathbf{x}',\mathbf{u})$. In this case, we can test whether our approximator of $\tilde{V}^{\pi_\theta}\left(\mathbf{x}\right) \approx V^{\pi_\theta}\left(\mathbf{x}\right)$ can ever stably approximate $V^{\pi_\theta}\left(\mathbf{x}\right)$. For this we have to plug all these functions into the Bellman equation (or alternatively the Poisson equation), and show that

$$\tilde{V}^{\pi_\theta}\left(\mathbf{x}\right) = \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}\left(\mathbf{u}\left|\mathbf{x}\right.\right)\left(\tilde{r}(\mathbf{x},\mathbf{u}) + \gamma \int_{\mathbb{X}} \tilde{p}\left(\mathbf{x}'\left|\mathbf{u},\mathbf{x}\right.\right)\tilde{V}^{\pi_\theta}\left(\mathbf{x}'\right)d\mathbf{x}'\right)d\mathbf{u}$$

holds. If we do model-based reinforcement learning, we can usually directly calculate the value function from the analytical models. It is fairly obvious, that for many general function approximation methods used for $\tilde{V}^{\pi_\theta}\left(\mathbf{x}\right)$, e.g., tile-coding or multi-layer perceptrons, this equations will not hold. However, for nearest neighbor, or locally weighted regression architectures, it might very well hold. This agrees with Sutton & Bartos (1998) observation that these methods are usually more applicable in reinforcement learning. However, we also see that

$$V^*\left(\mathbf{x}\right) = \max_{\mathbf{u}\in\mathbb{U}}\left(r(\mathbf{x},\mathbf{u}) + \gamma \int_{\mathbb{X}} p\left(\mathbf{x}'\left|\mathbf{u},\mathbf{x}\right.\right)V^*(\mathbf{x}')d\mathbf{x}'\right),$$

is a very difficult integral equation. In fact, this integral equation usually has only few solutions, and becomes unsolvable for most approximators of $V^*\left(\mathbf{x}\right)$. This can also be seen as a reason why greedy methods fail with function approximation.

Chapter 3

Policy Gradient Theory

The essential idea of policy gradient methods in general is to follow the gradient of the average reward in parameter space to a locally optimal solution. This can be expressed as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \left. \frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \theta} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}, \tag{3.1}$$

where α_t denotes a learning rate. Obviously, the main problem for policy gradient methods is to obtain the policy gradient $\partial J(\pi_{\boldsymbol{\theta}})/\,\partial \boldsymbol{\theta}$ in parameter space. Furthermore, we will see that the standard policy gradient is often not efficient as it gets stuck in local minima. This chapter deals with these problems. We will first present the policy gradient theorem with continuous spaces and actions, and then show how the policy gradient estimate can be improved using a baseline. Furthermore, we will present the compatible function approximation introduced by Sutton et al. (2000), and Konda & Tsitsiklis (2001, 2000, 2002). We will show it cannot represent the state-action value function but only the advantage function of the policy, and illustrate this with several examples. We will derive the optimal baseline using the compatible function approximation. Subsequently, we will show how an All-Action algorithm makes baselines obsolete. Finally, we will prove that Kakade's "average natural gradient" is in fact the true natural gradient, and the whole reinforcement learning problem boils down to learning the compatible function approximation.

3.1 Policy Gradient Theorem

Policy gradient theorem

The **policy gradient theorem** was first derived for the average-reward formulation by Marbach and Tsitsiklis (1998). Subsequently, Sutton et al. (2000) showed the same for the discounted start-state case. Both authors have shown the theorem for discrete state and action spaces as well as discrete time. We will now present the theorem for continuous state and action spaces.

Theorem 3 (Policy gradient theorem) *For any continuous Markov decision problem in average reward and discounted start-state formulation, the* ***policy gradient*** *can be expressed as*

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}} Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})\, d\mathbf{u} d\mathbf{x}. \tag{3.2}$$

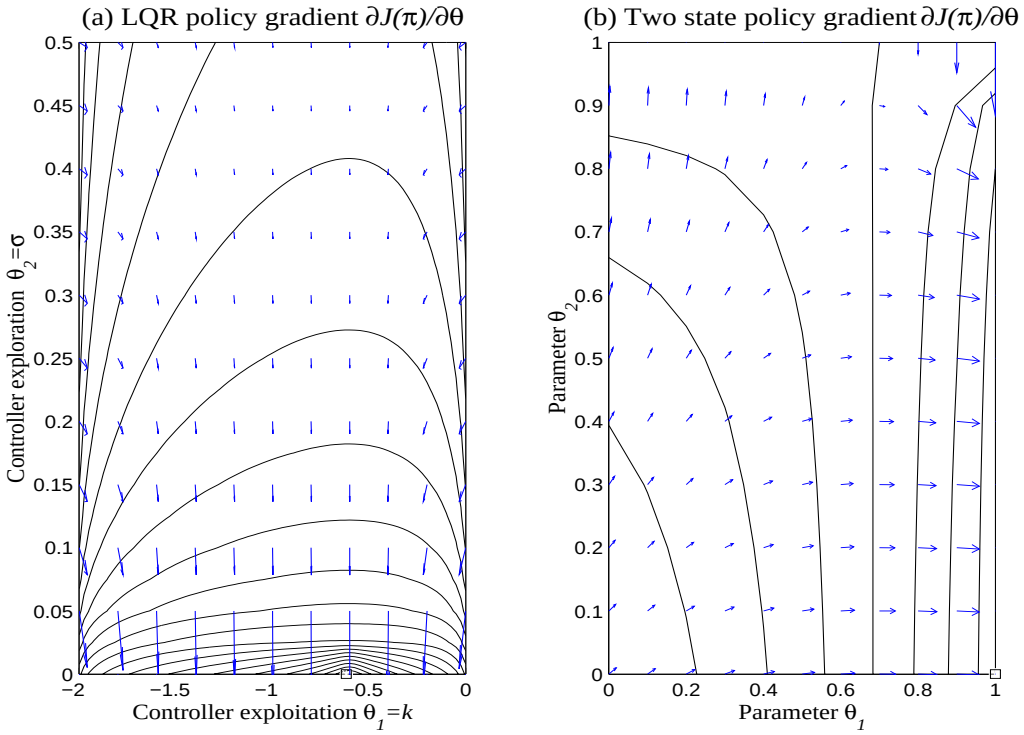


Figure 3.1: This figure shows the policy gradients for both (a) the one dimensional LQR problems, and (b) the two-state problem. It is obvious in both figures that the large plateaus in the expected return landscape cause major problems for the policy gradient approach.

Proof. The proof can be found in the Appendix. In Section **??**, we have a proof based on the second summation theorem which is more general. In Section **??**, we modify for the continuous case (Sutton et al., 2000). ■

In Figure 3.1, the policy gradient of our two standard examples from the last chapter is shown. Clearly, in both cases the policy gradient suffers significantly from the large plateaus around the optimal solutions, and therefore might be get stuck in suboptimal solutions. Furthermore, in Figure 3.1 (b), we realize that paths along the gradient lead out of the admissible parameters area.

3.2 Baselines

Baseline

Obviously, any method which increases the speed of convergence to the locally optimal solution is desirable. Already Williams (1991, 1992) noticed that a **baseline** $b^{\pi_\theta}(\mathbf{x})$ enhances the convergence speed of policy gradient methods when we have to estimate the gradient. Here, we will now show that we can add arbitrary baselines without affecting the expectation of the policy gradient.

Theorem 4 (Baselines) *Even when an arbitrary baseline $b^{\pi_\theta}(\mathbf{x})$ is subtracted from the state-action value function $Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})$, the policy gradient remains the same in ex-*

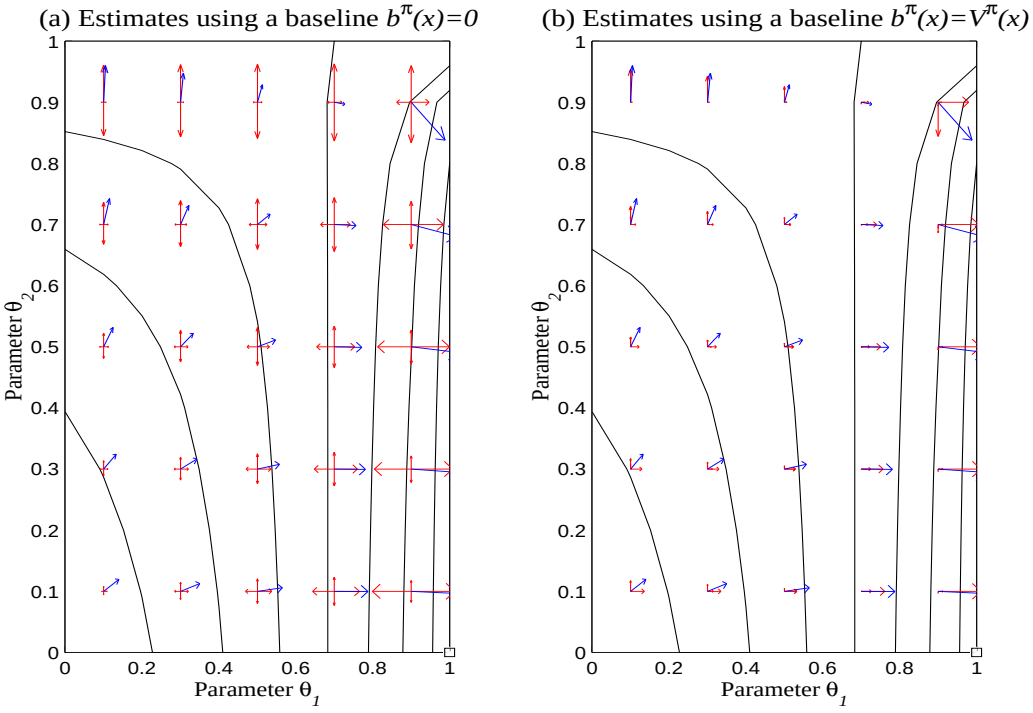


Figure 3.2: This figure shows the effects of baselines on the policy gradient estimates for the two state problem. The blue arrows show the resulting gradient while the red arrows show the gradient components contributed by single actions. The red arrows in (a) are scaled down by 1/10, i.e., the resulting gradient length is roughly 1/10th of the single components. Furthermore, all different actions lead to gradients in other directions which cancel out only in expectation. In practice, the estimate will always be noisy. In (b), both the resulting gradient as well as its component are shown on the same scale. As all components point into the right directions, the resulting gradient estimate will have a reduced variance, and increased accuracy. The size of gradient components is always smaller than the resulting gradient.

pectation, i.e.,

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) - b^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})\right) d\mathbf{u} d\mathbf{x}, \tag{3.3}$$

$$= \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u})\, d\mathbf{u} d\mathbf{x}. \tag{3.4}$$

Proof. Since $\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})$ is a probability distribution, we have $\int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})\, d\mathbf{u} = 1$, this implies that

$$\frac{\partial}{\partial \boldsymbol{\theta}} \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})\, d\mathbf{u} = \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} d\mathbf{u} = \frac{\partial}{\partial \boldsymbol{\theta}} 1 = 0.$$

Subsequently, we use this in order to modify the previous theorem

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) - b^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})\right) d\mathbf{u} d\mathbf{x},$$

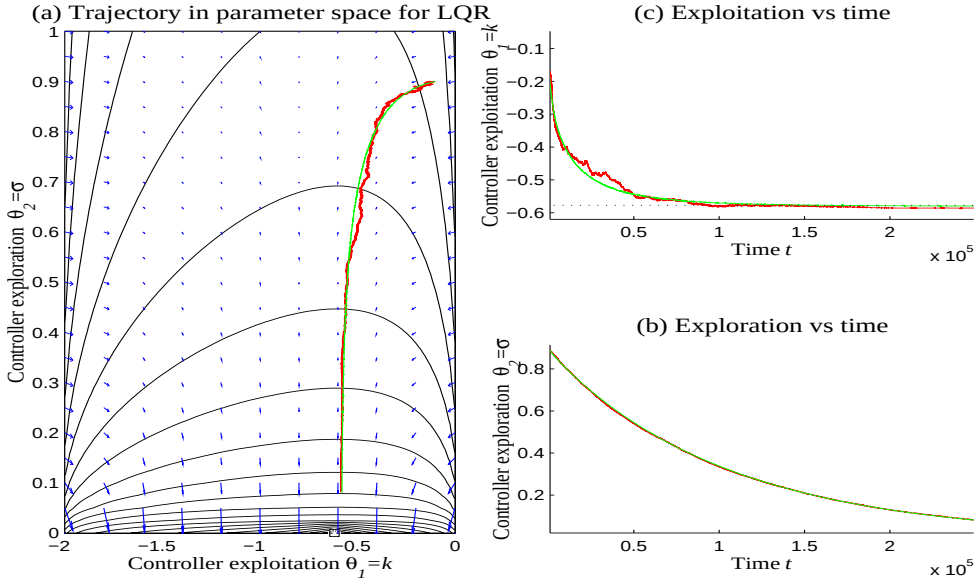


Figure 3.3: In this figure, we compare the performance on the same learning problem with and without a baseline. The red, jagged curves represent the performance with no baseline, i.e., $b^{\pi_\theta}(x) = 0$. The green, smooth curves represent the perfromance with a baseline of $b^{\pi_\theta}(x) = V^{\pi_\theta}(x)$. An approximate of the policy gradient is obtained using a single roll-out, i.e., a trajectory of length $T = 100$ steps. After one roll-out, we do one offline update of the parameters using a learning rate $\alpha_t = \sigma_t^2 \cdot 10^{-6}$. Please note that despite this successful trial, many trials from other starting positions lead to suboptimal solutions. This can be seen as part of the exploration- exploitation dilemma. In (a), we see the trajectory in parameter space; in (b), and (c), the parameters are plotted versus time. We use the analytical value functions $Q^{\pi_\theta}(x, u)$, and $V^{\pi_\theta}(x)$ for the updates.

$$= \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) \, d\mathbf{u} - b^{\pi_\theta}(\mathbf{x}) \underbrace{\int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} d\mathbf{u} d\mathbf{x}}_{=0},$$

$$= \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) \, d\mathbf{u} d\mathbf{x}.$$

Clearly this proves our theorem. ■

However, the important question is what is the optimal baseline? We will address this question in a following section. In Figure 3.2 (a), we can see how a simple baseline $b^{\pi_\theta}(\mathbf{x}) = V^{\pi_\theta}(\mathbf{x})$ affects the performance of the learning system. Furthermore, in Figure 3.3 (a-b), we can see the reason for this: the gradient contributions for each state and action are calculated. The gradients components for different actions point in various directions if no baseline (i.e., $b^{\pi_\theta}(\mathbf{x}) = 0$) is used as can be seen in Figure 3.3 (a). However, using a baseline $b^{\pi_\theta}(\mathbf{x}) = V^{\pi_\theta}(\mathbf{x})$, the components of all actions are aligned in a similar direction.

From the baseline theorem, we can directly see that the policy gradient $\partial J(\pi_\theta)/\partial \boldsymbol{\theta}$ does not use the whole information included in the state-action value function $Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})$. Let us split it into the advantage function $A^{\pi_\theta}(\mathbf{x}, \mathbf{u})$, and the state value function $V^{\pi_\theta}(\mathbf{x})$, i.e., $Q^{\pi_\theta}(\mathbf{x}, \mathbf{u}) = A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) + V^{\pi_\theta}(\mathbf{x})$. In this case, it is clear that

$$\frac{\partial J(\pi_\theta)}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(A^{\pi_\theta}(\mathbf{x}, \mathbf{u}) + V^{\pi_\theta}(\mathbf{x}) \right) d\mathbf{u} d\mathbf{x}, \tag{3.5}$$

$$= \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} A^{\pi_\theta}(\mathbf{x},\mathbf{u})\, d\mathbf{u} d\mathbf{x}, \tag{3.6}$$

since $V^{\pi_\theta}(\mathbf{x})$ is not a function of $\mathbf{u}$. This makes clear that all information in $V^{\pi_\theta}(\mathbf{x})$ is not necessary for determining the policy gradient, i.e., it is redundant.

3.3 Compatible Function Approximation

Clearly, we have to represent at least the advantage function $A^{\pi_\theta}(\mathbf{x},\mathbf{u})$ or the whole state-action value function $Q^{\pi_\theta}(\mathbf{x},\mathbf{u})$, in order to calculate the policy gradient. In (Sutton et al., 2000), the following derivation of a **compatible function approximation** is given. For this we have to find a function approximator $f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})$ which represents it so that the squared error between these two is minimal, i.e.,

Compatible function approximation

$$E^2 = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \left(Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) - f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})\right)^2 d\mathbf{u} d\mathbf{x} \longrightarrow \min.$$

The following theorem provides us with the necessary conditions for this equivalence.

Theorem 5 (Compatible basis functions) *If a function approximator* $f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})$ *satisfies the compatibility*

$$\frac{\partial f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})}{\partial \boldsymbol{w}_\theta} = \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}, \tag{3.7}$$

and, after convergence, (2) the error minimization

$$E^2 = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \left(Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) - f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})\right)^2 d\mathbf{u} d\mathbf{x} \longrightarrow \min, \tag{3.8}$$

we can replace $Q^{\pi_\theta}(\mathbf{x},\mathbf{u})$ *by* $f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})$ *so that we get*

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u}) d\mathbf{u} d\mathbf{x}. \tag{3.9}$$

Proof. Assume that a function approximator fulfills the conditions (3.7), and (3.8). By differentiating the error by the function approximation parameters $\boldsymbol{w}_{\boldsymbol{\theta}}$, we learn that this is only the case for

$$\frac{\partial E^2}{\partial \boldsymbol{w}_\theta} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \frac{\partial f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}}{\partial \boldsymbol{\theta}} \left(Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) - f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})\right) d\mathbf{u} d\mathbf{x} = 0.$$

We can substitute for $\partial f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}} / \partial \boldsymbol{\theta}$ using equation (3.7), and obtain

$$\int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(Q^{\pi_\theta}(\mathbf{x},\mathbf{u}) - f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u})\right) d\mathbf{u} d\mathbf{x} = 0.$$

By splitting the integral, we obtain

$$\int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} Q^{\pi_\theta}(\mathbf{x},\mathbf{u})\, d\mathbf{u} d\mathbf{x} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} f_{\boldsymbol{w}_\theta, Q^{\pi_\theta}}(\mathbf{x},\mathbf{u}) d\mathbf{u} d\mathbf{x}.$$

This implies

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} f_{\boldsymbol{w}_{\boldsymbol{\theta}}, Q^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}) d\mathbf{u} d\mathbf{x},$$

which proves the theorem. ■

From equation (3.7) it becomes clear that we do can derive the class of all compatible function approximators easily. All we need to do is to integrate these basis functions up in the function approximators parameter space.

Theorem 6 (Compatible function approximation) *A policy compatible function approximator of the advantage function $A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u})$ is given by*

$$f_{\boldsymbol{w}_{\boldsymbol{\theta}}, A^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}) = \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{w}_{\boldsymbol{\theta}}, \tag{3.10}$$

and of the state-action value function $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u})$ by

$$f_{\boldsymbol{w}_{\boldsymbol{\theta}}, Q^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}) = \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{w}_{\boldsymbol{\theta}} + C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}), \tag{3.11}$$

where $C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$ represents unknown basis function. We will refer to $f_{\boldsymbol{w}_{\boldsymbol{\theta}}, A^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u})$ as $f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u})$ from now on.

Proof. We integrate the basis functions over $\boldsymbol{w}_{\boldsymbol{\theta}}$, and get

$$\begin{aligned} f_{\boldsymbol{w}_{\boldsymbol{\theta}}, Q^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}) &= \int \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} d\boldsymbol{w}_{\boldsymbol{\theta}}, \\ &= \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{w}_{\boldsymbol{\theta}} + C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}). \end{aligned}$$

This is a compatible function approximator. Using the previously derived result for the integral of the derivative of the policy $\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})$, i.e., $\int_{\mathbb{U}} \partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) / \partial \boldsymbol{\theta} \, d\mathbf{u} = \mathbf{0}$,we see that the state value function equals

$$V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) = \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) d\mathbf{u} = \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}^T d\mathbf{u} \, \boldsymbol{w}_{\boldsymbol{\theta}} + C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) = C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}).$$

This implies that for $f_{\boldsymbol{w}_{\boldsymbol{\theta}}, Q^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}) = Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u})$,with $Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) = A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) + V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$,we also have

$$\begin{aligned} A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) &= \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})} \left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{w}_{\boldsymbol{\theta}} = f_{\boldsymbol{w}_{\boldsymbol{\theta}}, A^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u}), \\ V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) &= C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}). \end{aligned}$$

This concludes the theorem. ■

We clearly see the weakness of the state-action value function approximator $f_{\boldsymbol{w}_{\boldsymbol{\theta}}, Q^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u})$, i.e., we have to learn an additional function $C^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) = V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$ in order to learn $\boldsymbol{w}_{\boldsymbol{\theta}}$, although this function disappears in expectation of the gradient. However, it is not clear how we could learn $f_{\boldsymbol{w}_{\boldsymbol{\theta}}, A^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u})$ directly from the data. We will study this in-depth in the Chapter **??**. From now on, we will focus on the advantage function approximator $f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) = f_{\boldsymbol{w}_{\boldsymbol{\theta}}, A^{\pi_{\boldsymbol{\theta}}}}(\mathbf{x}, \mathbf{u})$, and show for our previous examples how it fits in.

Example 10 (Gaussian policy) *In order to derive the policy compatible function approximation for our Gaussian policy (Example 1, page 9), we have to calculate the gradients of the policy with respect to its parameters:*

Gaussian policy compatible function approximation

$$\frac{\partial \pi_{\boldsymbol{\theta}}(u|\mathbf{x})}{\partial \mathbf{k}} = \frac{1}{\sigma^2}\pi_{\boldsymbol{\theta}}(\mathbf{x},u)\left(u-\mathbf{k}^T\mathbf{x}\right)\mathbf{x},$$
$$\frac{\partial \pi_{\boldsymbol{\theta}}(u|\mathbf{x})}{\partial \sigma} = \pi_{\boldsymbol{\theta}}(\mathbf{x},u)\left(-\frac{1}{\sigma}+\frac{1}{\sigma^3}\left(u-\mathbf{k}^T\mathbf{x}\right)^2\right).$$

which gives us the full gradient

$$\frac{\partial \pi_{\boldsymbol{\theta}}(u|\mathbf{x})}{\partial \boldsymbol{\theta}} = \pi_{\boldsymbol{\theta}}(\mathbf{x},u)\left[\begin{array}{cc} \frac{1}{\sigma^2}\left(u-\mathbf{k}^T\mathbf{x}\right)\mathbf{x}^T, & -\frac{1}{\sigma}+\frac{1}{\sigma^3}\left(u-\mathbf{k}^T\mathbf{x}\right)^2 \end{array}\right]^T.$$

By integrating in function approximation parameter space, we can obtain

$$\begin{aligned} f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(\mathbf{x},u) &= \int \frac{1}{\pi_{\boldsymbol{\theta}}(u|\mathbf{x})}\frac{\partial \pi_{\boldsymbol{\theta}}(u|\mathbf{x})}{\partial \boldsymbol{\theta}} d\boldsymbol{w}_{\boldsymbol{\theta}}, \\ &= \boldsymbol{w}_{\boldsymbol{\theta}}^T\left[\begin{array}{cc} \frac{1}{\sigma^2}\left(u-\mathbf{k}^T\mathbf{x}\right)\mathbf{x}^T, & -\frac{1}{\sigma}+\frac{1}{\sigma^3}\left(u-\mathbf{k}^T\mathbf{x}\right)^2 \end{array}\right]^T, \\ &= \frac{1}{\sigma^2}\boldsymbol{w}_1^T u\mathbf{x} - \frac{1}{\sigma^2}\mathbf{k}^T\mathbf{x}\boldsymbol{w}_1^T\mathbf{x} - w_2\frac{1}{\sigma} + w_2\frac{1}{\sigma^3}\left(u^2 - 2u\mathbf{k}^T\mathbf{x} + \left(\mathbf{k}^T\mathbf{x}\right)^2\right), \\ &= \left[\begin{array}{cc} \mathbf{x}^T & u^T \end{array}\right]\left[\begin{array}{cc} -\mathbf{k}\boldsymbol{w}_1^T\frac{1}{\sigma^2} + w_2\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T & \frac{1}{2}\boldsymbol{w}_1\frac{1}{\sigma^2} - \mathbf{k}w_2^T\frac{1}{\sigma^3} \\ \frac{1}{2}\boldsymbol{w}_1^T\frac{1}{\sigma^2} - w_2\frac{1}{\sigma^3}\mathbf{k}^T & \frac{1}{\sigma^3}w_2 \end{array}\right]\left[\begin{array}{c} \mathbf{x} \\ u \end{array}\right] - w_2\frac{1}{\sigma}. \end{aligned} \tag{3.12}$$

In here, we use $\boldsymbol{w}_{\boldsymbol{\theta}} = \left[\boldsymbol{w}_1^T, w_2\right]^T$. *By this, we have obtained the compatible function approximator.*

Example 11 (Gibbs policy) *Similarly, we can do the same for the Gibbs policy (Example 2, page 10). The derivative of the log-policy with respect to* θ_{xu} *would be given by*

Gibbs policy compatible function approximation

$$\begin{aligned} \frac{1}{\pi(u_i|x_j)}\frac{\partial \pi(u_i|x_j)}{\partial \theta} &= \frac{\partial \log \pi(u_i|x_j)}{\partial \theta} \\ &= \frac{\partial}{\partial \theta}\left(\theta^T\phi(u|x) - \log\left(\sum_{u'\in\mathbb{U}}\exp\left(\theta^T\phi(u'|x)\right)\right)\right), \\ &= \phi(u|x) - \sum_{u'\in\mathbb{U}}\pi(u'|x)\,\phi(u'|x). \end{aligned}$$

This implies

$$\frac{1}{\pi(u_i|x_j)}\frac{\partial \pi(u_i|x_j)}{\partial \theta_{kl}} = \left\{\begin{array}{ccc} 1-\pi(u_k|x_l) & \text{if} & k=i\wedge l=j \\ -\pi(u_k|x_l) & \text{if} & k=i\wedge l\neq j \\ 0 & \text{if} & k\neq i \end{array}\right. .$$

The compatible function approximation is given by $f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(x,u) = \partial \log \pi(u_i|x_j)/\partial \theta^T \boldsymbol{w}_{\boldsymbol{\theta}}$.

Example 12 (Decision border policy) *Similarly, we can do the same for the decision border policy (Example 2, page 10). Here we have*

Decision border policy compatible function approximation

$$\frac{1}{\pi(u_i|x_j)}\frac{\partial \pi(u_i|x_j)}{\partial \theta_{kl}} = \left\{\begin{array}{ccc} \frac{1}{\theta_{kl}} & \text{if} & k=i\wedge l=j\neq m \\ -\frac{1}{1-\sum_{h=1}^{m-1}\theta_{kh}} & \text{if} & k=i\wedge j=m \\ 0 & \text{if} & k\neq i \end{array}\right. .$$

The compatible function approximation is given by $f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(x,u) = \partial \log \pi(u_i|x_j)/\partial \theta^T \boldsymbol{w}_{\boldsymbol{\theta}}$.

These example show very clearly how such a compatible function approximator can be obtained. We will now show that for both cases it can indeed represent the advantage function.

LQR analytical function approximation parameters

Example 13 (Linear quadratic control with Gaussian policies) *Let us pick up Example 1, page 9. We will now show that the advantage function $A^{\pi_\theta}(\mathbf{x},\mathbf{u})$ of linear quadratic control (Example 7, page 17) can be represented by the compatible function approximator $f_{\boldsymbol{w}_\theta}(x,u)$ from Example 10, page 33. From Example 10, we know that the advantage for LQR under policy π_θ is given by*

$$A^{\pi_\theta}(\mathbf{x},\mathbf{u}) = -\frac{1}{2}\begin{bmatrix}\mathbf{x}^T, & u\end{bmatrix}\mathbf{H}\begin{bmatrix}\mathbf{x}\\ u\end{bmatrix} + \frac{1}{2}\left(R+\gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2, \tag{3.13}$$

in the start-state formulation. We intend to show that the advantage function $A^{\pi_\theta}(\mathbf{x},\mathbf{u})$ from Example 7 can be represented by the compatible function approximation $f_{\boldsymbol{w}_\theta}(x,u)$ from Example 10, i.e., $A^{\pi_\theta}(\mathbf{x},\mathbf{u}) = f_{\boldsymbol{w}_\theta}(x,u)$. For this we compare all terms of $A^{\pi_\theta}(\mathbf{x},\mathbf{u})$, equation (3.12), and $f_{\boldsymbol{w}_\theta}(x,u)$, equation (3.13), respectively. This gives us

$$\begin{bmatrix} -\mathbf{k}\boldsymbol{w}_1^T + w_2\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T & \frac{1}{2}\boldsymbol{w}_1 - \mathbf{k}w_2\frac{1}{\sigma^3} \\ \frac{1}{2}\boldsymbol{w}_1^T - w_2\frac{1}{\sigma^3}\mathbf{k}^T & \frac{1}{\sigma^3}w_2 \end{bmatrix} = -\frac{1}{2}\mathbf{H},$$
$$-w_2\frac{1}{\sigma} = +\frac{1}{2}\left(R+\gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2,$$

using $\boldsymbol{w}_\theta = \left[\boldsymbol{w}_1^T, w_2\right]^T$. This means that we have five equations which have to be satisfied by $\boldsymbol{w}_\theta$. If we solve these, we obtain a unique solution

$$\boldsymbol{w}_1 = -\mathbf{k}\left(R+\gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^2 - \gamma\mathbf{A}^T\mathbf{P}_\gamma\mathbf{b}\sigma^2,$$
$$w_2 = -\frac{1}{2}\left(R+\gamma\mathbf{b}^T\mathbf{P}_\gamma\mathbf{b}\right)\sigma^3,$$

which satisfies all five equations. Clearly, this shows that the advantage can be represented in the compatible function approximator. It is clear from this and the previous example that the same is true for the average reward case as it only differs by having $\gamma = 1$ as we have seen in Example 7.

Discrete Gibbs-policy analytical function approximation parameters

Example 14 (Discrete actions & state spaces with Gibbs policies) *Let us pick up Example 2, page 10. Similarly, we can directly obtain the same result for deterministic environments, i.e., $\mathcal{P}^u_{xx'} \in \{0,1\}$. In this case, we can show that*

$$\begin{aligned}
A^{\pi_\theta}(x_i,u_j) &= Q^{\pi_\theta}(x_i,u_j) - V^{\pi_\theta}(x_i)\\
&= r(x_i,u_j) + \sum_{k=1}^{n}\gamma\mathcal{P}^{u_j}_{x_ix_k}V^{\pi_\theta}(x_k) - V^{\pi_\theta}(x_i)\\
&= \left(1-\pi(u_j|x_i)\right)\left(r(x_i,u_j) + \sum_{k=1}^{n}\gamma\mathcal{P}^{u_j}_{x_ix_k}V^{\pi_\theta}(x_k)\right)\\
&- \sum_{h\neq j}\pi(u_h|x_i)\left(r(x_i,u_h) + \sum_{k=1}^{n}\gamma\mathcal{P}^{u_h}_{x_ix_k}V^{\pi_\theta}(x_k)\right)\\
&= \left(1-\pi(u_j|x_i)\right)Q^{\pi_\theta}(x_i,u_j) - \sum_{h\neq j}\pi(u_h|x_i)\,Q^{\pi_\theta}(x_i,u_h)\\
&= \frac{\partial\log\pi(u_j|x_i)}{\partial\theta}^T\begin{bmatrix}\boldsymbol{w}_{1\cdots(i-1)m}, & Q^{\pi_\theta}(x_i,u_1), & \cdots, & Q^{\pi_\theta}(x_i,u_n), & \boldsymbol{w}_{(i+1)m\cdots nm}\end{bmatrix}^T\\
&= \frac{\partial\log\pi(u_j|x_i)}{\partial\theta}^T\boldsymbol{w}_\theta.
\end{aligned}$$

Therefore it can also be represented by a policy compatible function approximator, and we know that $\boldsymbol{w}_\theta = [Q^{\pi_\theta}(x_1, u_1), \ldots, Q^{\pi_\theta}(x_m, u_n)]$.

Furthermore, we can show that the state-action value function cannot be represented without further basis function. For linear quadratic control with Gaussian basis functions, we can even obtain the basis functions.

State-action value functions need additional basis functions

Example 15 (Compatible Q-function approximation) *To undermine this claim, we reexamine our Gaussian policy LQR control example (Example 1, page 9). If we could represent* $f_{\boldsymbol{w}_\theta}(\mathbf{x}, u) = Q^{\pi_\theta}(\mathbf{x}, \mathbf{u})$*, we would have*

$$-w_2^T \frac{1}{\sigma} = -\frac{1}{2}\frac{\gamma}{1-\gamma}\left(R + \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{b}\right)\sigma^2,$$

$$\begin{bmatrix} -\mathbf{k}\boldsymbol{w}_1^T + w_2^T \frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T & \frac{1}{2}\boldsymbol{w}_1 - \mathbf{k} w_2^T \frac{1}{\sigma^3} \\ \frac{1}{2}\boldsymbol{w}_1^T - w_2^T \frac{1}{\sigma^3}\mathbf{k}^T & \frac{1}{\sigma^3} w_2^T \end{bmatrix} = -\frac{1}{2}\begin{bmatrix} \mathbf{Q} + \gamma \mathbf{A}^T \mathbf{P}_\gamma \mathbf{A} & \gamma \mathbf{A}^T \mathbf{P}_\gamma \mathbf{b} \\ \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{A} & \mathbf{R} + \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{b} \end{bmatrix},$$

in the discounted start-state formulation. Without further derivation, we see that this is not the case as there are more open parameters in this formulation than in the representation and there is no relationship among them. Furthermore, the relation between the rightmost lower elements of the matrix and the relation of the constant addition are clearly a contradiction. However, using ***additional basis functions***

$$C^{\pi_\theta}(\mathbf{x}) = V^{\pi_\theta}(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T \mathbf{P}_\gamma \mathbf{x} - \frac{1}{2}\frac{1}{1-\gamma}\left(R + \gamma \mathbf{b}^T \mathbf{P}_\gamma \mathbf{b}\right)\sigma^2,$$

we could represent it. The same holds true for the average reward formulation where exactly the same problem occurs.

The discussion in the previous section has shown fairly well that we are only interested in advantage functions. In this section, we have seen, that we can only represent the advantage functions without further basis functions. Furthermore, we have a compatible approximator with linear constants as parameters. We will use this now to obtain three major results, i.e., the optimal baseline, the all-action algorithm, and the natural gradient[1].

3.4 The Optimal Baseline

Optimal baseline

Already for REINFORCE, researchers (Williams, 1992; Gullapalli et al., 1994; Dayan, 1990) asked themselves the question what might be the **optimal baseline** $b^{\pi_\theta}(\mathbf{x})$. Similarly, this question has arisen again due to refocusing on policy gradient methods, and researchers have started investigating this topic again (Berny, 2000; Greensmith, Bartlett, & Baxter, 2001; Weaver & Tao, 2001a, 2001b). For REINFORCE mostly the average return of the policy $b^{\pi_\theta}(\mathbf{x}) = J(\pi)$, or the state value-function $b^{\pi_\theta}(\mathbf{x}) = V^{\pi_\theta}(\mathbf{x})$ has been used. Nevertheless, already Dayan (1990) was able to show that these baselines are suboptimal for a simple two-state MDP. Berny (2000) was the first to show that the optimal baseline for REINFORCE and related algorithms must **minimize the variance of the gradient**. Such an optimal baseline can be denoted by

Minimum variance baselines

$$b^{\pi_\theta}(\mathbf{x}) = \min_{b(\mathbf{x})} \operatorname{Var}\left(\frac{1}{\pi_\theta(\mathbf{u}|\mathbf{x})}\frac{\partial \pi_\theta(\mathbf{u}|\mathbf{x})}{\partial \theta}\left(f_{\boldsymbol{w}_\theta}(\mathbf{x}, \mathbf{u}) - b(\mathbf{x})\right)\right).$$

[1] It would be interesting to show that we can derive from the linear parameterization of the policy that the advantage function always has the form of the compatible approximator.

Using the compatible function approximation, we can try to search the optimal baseline $b^{\pi_\theta}(\mathbf{x})$ for approximation the policy gradient with a minimal variance. Instead of deriving this result here, we will present it straight ahead and refer the interested reader to the Appendix, Section **??**, page **??**.

Theorem 7 (Optimal baseline) *The optimal baseline $b^{\pi_\theta}(\mathbf{x})$ which minimizes the variance*

$$\mathrm{Var}\left\{\frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}\left(f_{\boldsymbol{w}_\theta}(\mathbf{x},\mathbf{u})-b^{\pi_\theta}(\mathbf{x})\right)\right\}\rightarrow \min, \tag{3.14}$$

is given by

$$b^{\pi_\theta}(\mathbf{x}) = \mathbf{F}_{\boldsymbol{\theta}}^{-1}(\mathbf{x})\int_{\mathbb{U}} \boldsymbol{w}_{\boldsymbol{\theta}}^{T}\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}^{T}\frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})^2}d\mathbf{u}, \tag{3.15}$$

where

$$\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbb{U}}\frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}\left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})}{\partial \boldsymbol{\theta}}^{T}\right)d\mathbf{u} \tag{3.16}$$

denotes a matrix of point $\mathbf{x}$ for policy $\pi_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$.

Obviously, all integrals can be evaluated analytically, and therefore we have an algebraic function with no unknown components except for the policy compatible function approximation parameters $\boldsymbol{w}_{\boldsymbol{\theta}}$ in here.

The matrix $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$ is of greater interest: Kakade (2001) calls $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$ a point Fisher information matrix, and we will later discuss the background of this. However, already for simple examples, $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$ can be non-invertible (see Example 16).

The reader might wonder why we do not discuss this theorem in a similar depth as previous ones. The reason for this is two-fold. First, we will see in the next section that we can make baselines obsolete using a simple trick, the All-Action Algorithm. Second, there might be infinitely many optimal baselines since $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$ is not generally invertible.

3.5 All-Action Algorithm

We have seen in the last section, that we can obtain optimal baselines. However, these appeared to be rather impractical as there might be infinitely many solutions for them. This poses the question whether we might be able to make them obsolete. Luckily, this is the case as Sutton already noticed in (Sutton et al., 2001) where he indicates that there is an **all-action algorithm**.

All-action algorithm

Theorem 8 (All-action algorithm) *The all-action form of the policy gradient does not require any baselines. It is given in the form*

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_\theta}(\mathbf{x})\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})\boldsymbol{w}_{\boldsymbol{\theta}}d\mathbf{x}, \tag{3.17}$$

where

$$\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbb{U}} \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})} \left(\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}}^T \right) d\mathbf{u} \tag{3.18}$$

can be derived analytically.

Proof. We can obtain this directly from the baseline theorem

$$\begin{aligned}
\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} &= \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(f_{\boldsymbol{w}_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) - b^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \right) d\mathbf{u} d\mathbf{x}, \\
&= \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}} \left(\frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}}^T \boldsymbol{w}_{\boldsymbol{\theta}} - b^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \right) d\mathbf{u} d\mathbf{x}, \\
&= \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \int_{\mathbb{U}} \frac{1}{\pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}} \frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{u}\,|\mathbf{x})}{\partial \boldsymbol{\theta}}^T d\mathbf{u}\; \boldsymbol{w}_{\boldsymbol{\theta}} d\mathbf{x} \\
&= \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})\, \boldsymbol{w}_{\boldsymbol{\theta}} d\mathbf{x}.
\end{aligned}$$

This clearly proves the theorem. ■

Furthermore, if we define the **all-action matrix** $\mathbf{F}(\boldsymbol{\theta}) = \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})\, d\mathbf{x}$, we can express the policy gradient by *All-action matrix*

$$\frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \int_{\mathbb{X}} d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) \mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})\, \boldsymbol{w}_{\boldsymbol{\theta}} d\mathbf{x} \;= \mathbf{F}(\boldsymbol{\theta})\, \boldsymbol{w}_{\boldsymbol{\theta}}.$$

Therefore, we can get a minimum variance estimate already if we have just estimated the average state density $d^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$, and the parameters $\boldsymbol{w}_{\boldsymbol{\theta}}$ with sufficient exactness.

Conclusion 3 (Baselines) *In order to obtain a minimum variance estimate of the policy gradient $\partial J(\pi_{\boldsymbol{\theta}})/\partial \boldsymbol{\theta}$, we do not need any baselines $b^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$. Only the All-Action-Matrix $\mathbf{F}(\boldsymbol{\theta})$, and the policy compatible function approximation parameters $\boldsymbol{w}_{\boldsymbol{\theta}}$ have to be estimated.*

Nevertheless, $\mathbf{F}(\boldsymbol{\theta})$ is difficult to estimate from the data even given $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$. In practice it usually requires a large amount of trials. Furthermore, the problems with plateaus remain. In Section 3.6, we will see that both problems can be avoided.

Example 16 (Gaussian all-action matrix) *A simple example where the matrix $\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$ is easy to determine is our Gaussian policy (Example 1, page 3). Here, we have* *Gaussian all-action matrix*

$$\begin{aligned}
\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x}) &= \int_{-\infty}^{\infty} \pi_{\boldsymbol{\theta}}(u\,|\mathbf{x}) \frac{\partial \log \pi_{\boldsymbol{\theta}}(u\,|\mathbf{x})}{\partial \boldsymbol{\theta}} \frac{\partial \log \pi_{\boldsymbol{\theta}}(u\,|\mathbf{x})}{\partial \boldsymbol{\theta}}^T du, \\
&= \int_{-\infty}^{\infty} \pi_{\boldsymbol{\theta}}(u\,|\mathbf{x}) \begin{bmatrix} \frac{\partial \log \pi}{\partial \mathbf{k}} \frac{\partial \log \pi}{\partial \mathbf{k}}^T & \frac{\partial \log \pi}{\partial \mathbf{k}} \frac{\partial \log \pi}{\partial \sigma} \\ \frac{\partial \log \pi}{\partial \sigma} \frac{\partial \log \pi}{\partial \mathbf{k}}^T & \frac{\partial \log \pi}{\partial \sigma} \frac{\partial \log \pi}{\partial \sigma} \end{bmatrix} du, \\
&= \begin{bmatrix} \frac{1}{\sigma^2} \mathbf{x}\mathbf{x}^T & 0 \\ 0 & \frac{2}{\sigma^2} \end{bmatrix}.
\end{aligned}$$

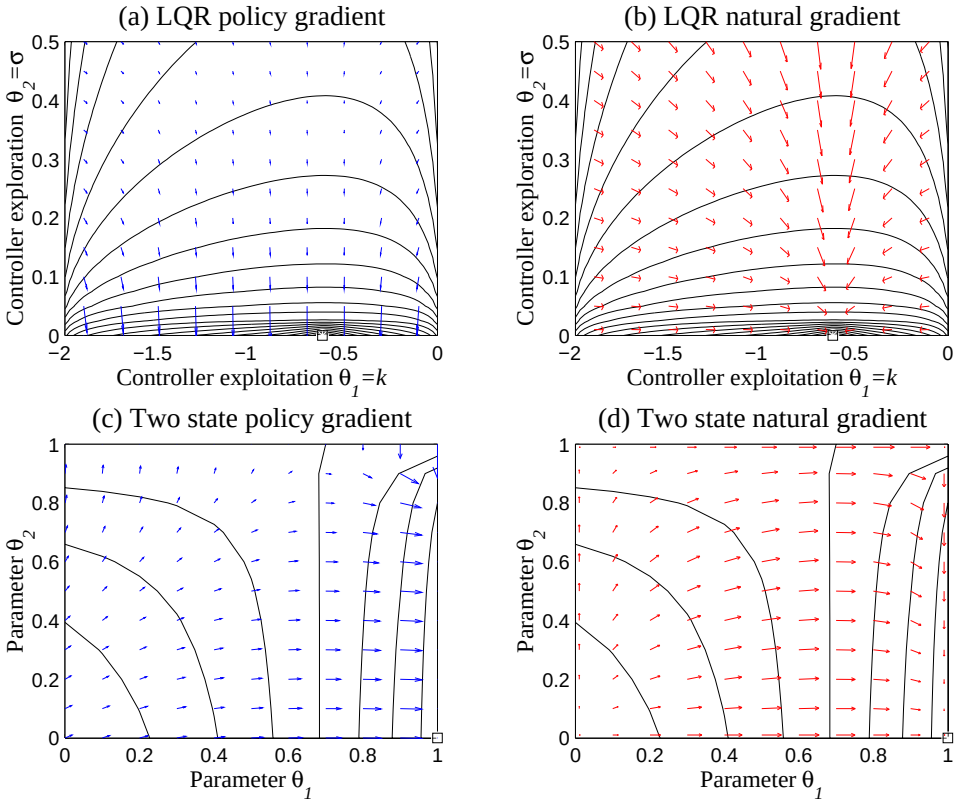


Figure 3.4: This figure compares the natural gradient to the policy gradient. In (a), the policy gradient, and in (b) the natural gradient of the LQR problem with a Gaussian policy is shown. The LQR again had the parameters $A = b = R = Q = 1$, and $\gamma = 0.95$. The natural gradient had to be normalized to be nicely visible. In (c), the policy gradient, and in (d) the natural gradient of the two state problem with a decision border policy. The natural gradient of the two-state problem has not been normalized. The discount factor of the two-state problem is $\gamma = 0.95$.

Clearly, this matrix is not generally invertable since $\mathbf{x}\mathbf{x}^T$ is of rank 1. For a proof of this, please refer to appendix, Section A.1.5, page 53.

3.6 Natural Gradient

Natural gradients

Natural gradients have received a lot of attention lately (Amari, 2000) as they improve the performance of stochastic gradient estimators significantly: they are unbiased, have a minimum variance, and do not get stuck in plateaus. In natural gradient methods, the gradient multiplied with the inverse of the **Fisher information matrix** $\mathbf{M}(\boldsymbol{\theta})$ is used as a new, more efficient gradient, i.e.,

Fisher information matrix

$$\tilde{\nabla} J(\pi_{\boldsymbol{\theta}}) = \mathbf{M}^{-1}(\boldsymbol{\theta}) \frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}.$$

A particular advantage of natural gradients is that plateaus are significantly less severe. Since we have large plateaus already for the most simple problems (see Figure 3.1) this is particularly helpful. However, local minima cannot be avoided by these methods.

Kakade (2001) shows that on the average, the Fisher information matrix $\mathbf{M}(\boldsymbol{\theta})$ can become our previously introduced all-action matrix $\mathbf{F}(\boldsymbol{\theta})$, i.e., $\mathbf{F}(\boldsymbol{\theta}) \rightarrow \mathbf{M}(\boldsymbol{\theta})$ for many

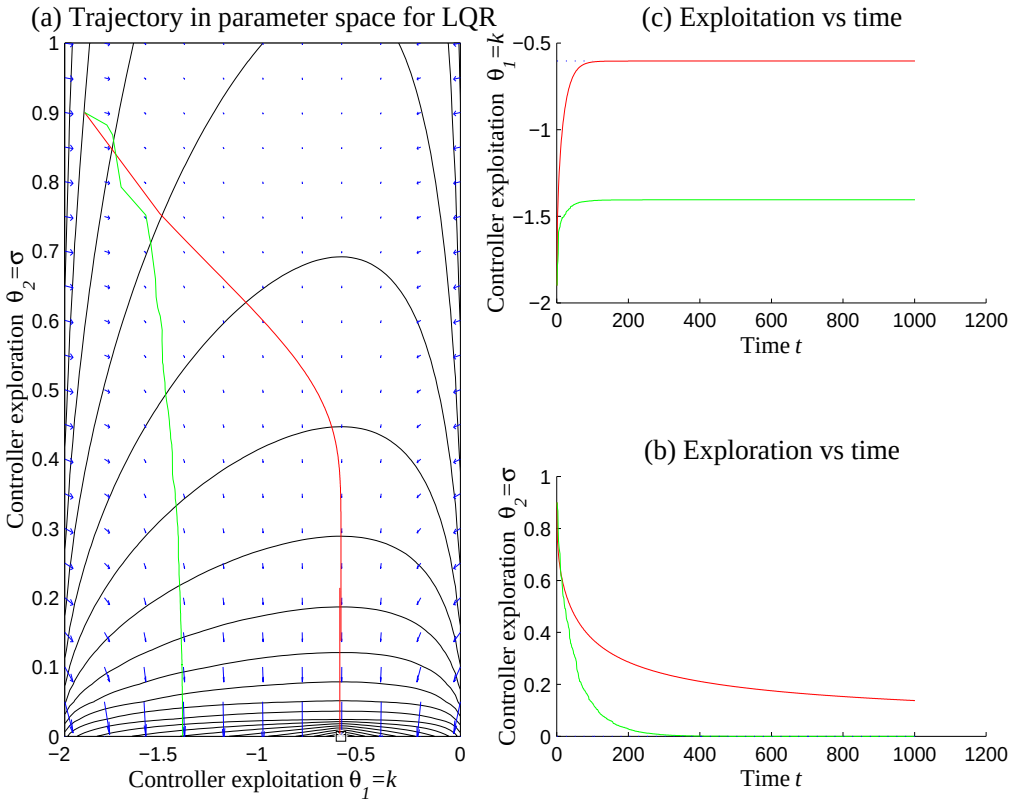


Figure 3.5: This figure shows the performance of natural gradients in comparison to policy gradients. In (a), you can see the performance in terms of a trajectory in the parameters space. In (a), and (b), you can see the parameters over time. To make the performance differences obvious, a large learning rate of $\alpha = \sigma_t \cdot 10^{-3}$ was taken. For such learning rates, there is practically no case where the optimal solution is being found by the policy gradient algorithm in this problem unless the exploration rate is being fixed. The natural gradient, however, converges to the optimal solution with no problem.

trials. This opens the question whether this can be the case in general. From (Berger & Casella, 2002), we know that for an estimation problem of $J(\pi_{\boldsymbol{\theta}}) = \int_{\mathbb{T}} R(\mathcal{T})\, p(\mathcal{T})\, d\mathcal{T}$, the Fisher information matrix is given by

$$\mathbf{M}(\theta) = E_{\mathcal{T}} \left\{ \frac{\partial \log p(\mathcal{T})}{\partial \boldsymbol{\theta}} \frac{\partial \log p(\mathcal{T})}{\partial \boldsymbol{\theta}}^T \right\},$$

if $\mathrm{Var}_{\mathcal{T}} \{J(\pi_{\boldsymbol{\theta}})\} < \infty$, and $\partial J(\pi_{\boldsymbol{\theta}}) / \partial \boldsymbol{\theta} = \int_{\mathbb{T}} R(\mathcal{T})\, \partial p(\mathcal{T}) / \partial \boldsymbol{\theta}\, d\mathcal{T}$ by the Cramér-Rao inequality. We can derive

$$\begin{aligned}
\frac{\partial p(\mathcal{T})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(p(\mathbf{x}_0) \prod_{t=1}^{\infty} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \, \pi(\mathbf{u}_{t-1} | \mathbf{x}_{t-1}) \right), \\
&= \left(p(\mathbf{x}_0) \prod_{t=1}^{\infty} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \right) \frac{\partial}{\partial \boldsymbol{\theta}} \left(\prod_{t=1}^{\infty} \pi(\mathbf{u}_{t-1} | \mathbf{x}_{t-1}) \right), \\
&= \left(p(\mathbf{x}_0) \prod_{t=1}^{\infty} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \right) \left(\prod_{t=1}^{\infty} \pi(\mathbf{u}_{t-1} | \mathbf{x}_{t-1}) \right)
\end{aligned}$$

$$\cdot \left(\sum_{t=1}^{\infty} \frac{\partial \log \pi \left(\mathbf{u}_{t-1} | \mathbf{x}_{t-1} \right)}{\partial \boldsymbol{\theta}} \right),$$

$$\frac{\partial \log p\left(\mathcal{T}\right)}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{\infty} \frac{\partial \log \pi \left(\mathbf{u}_{t-1} | \mathbf{x}_{t-1} \right)}{\partial \boldsymbol{\theta}}.$$

We can differentiate $\partial \log p\left(\mathcal{T}\right) / \partial \boldsymbol{\theta}$ again, and we get

$$\frac{\partial^2 \log p\left(\mathcal{T}\right)}{\partial \boldsymbol{\theta}^2} = \sum_{t=1}^{\infty} \frac{\partial^2 \log \pi_{\boldsymbol{\theta}} \left(\mathbf{u}_{t-1} | \mathbf{x}_{t-1} \right)}{\partial \boldsymbol{\theta}^2}.$$

From (Casella & Berger, 1999), we know that for any probability density function $p\left(y\right)$ the rule $\int_{\mathbb{Y}} p\left(y\right) \partial \log p\left(y\right) / \partial \boldsymbol{\theta} \cdot \partial \log p\left(y\right) / \partial \boldsymbol{\theta}^T \, dy = - \int_{\mathbb{Y}} p\left(y\right) \partial^2 \log p\left(y\right) / \partial \boldsymbol{\theta}^2 \, dy$ holds. Using this rule and the sequence summation theorem, we can easily derive

$$\begin{aligned}
\mathbf{M}(\theta) &= E_{\mathcal{T}} \left\{ \frac{\partial \log p\left(\mathcal{T}\right)}{\partial \boldsymbol{\theta}} \frac{\partial \log p\left(\mathcal{T}\right)}{\partial \boldsymbol{\theta}}^T \right\}, \\
&= -E_{\mathcal{T}} \left\{ \frac{\partial^2 \log p\left(\mathcal{T}\right)}{\partial \boldsymbol{\theta}^2} \right\}, \\
&= -E_{\mathcal{T}} \left\{ \sum_{t=1}^{\infty} \frac{\partial^2 \log \pi_{\boldsymbol{\theta}} \left(\mathbf{u}_{t-1} | \mathbf{x}_{t-1} \right)}{\partial \boldsymbol{\theta}^2} \right\}, \\
&= -E_{d^{\pi_{\boldsymbol{\theta}}}, \pi_{\boldsymbol{\theta}}} \left\{ \frac{\partial^2 \log \pi_{\boldsymbol{\theta}} \left(\mathbf{u} | \mathbf{x} \right)}{\partial \boldsymbol{\theta}^2} \right\}, \\
&= E_{d^{\pi_{\boldsymbol{\theta}}}, \pi_{\boldsymbol{\theta}}} \left\{ \frac{\partial \log \pi_{\boldsymbol{\theta}} \left(\mathbf{u} | \mathbf{x} \right)}{\partial \boldsymbol{\theta}} \frac{\partial \log \pi_{\boldsymbol{\theta}} \left(\mathbf{u} | \mathbf{x} \right)}{\partial \boldsymbol{\theta}}^T \right\} = \mathbf{F}\left(\boldsymbol{\theta}\right).
\end{aligned}$$

The all-action matrix is the Fisher information matrix

Therefore, we know that $\mathbf{M}\left(\boldsymbol{\theta}\right) = \mathbf{F}\left(\boldsymbol{\theta}\right)$ is the Fisher information matrix in general (for the average reward formulation). This proof extends Kakade's (2001) statement, who showed that $\mathbf{F}\left(\boldsymbol{\theta}\right)$ is the average of the Fisher information matrix. Using the relation $\sum_{n=0}^{\infty} K_{\pi_{\boldsymbol{\theta}}}^n \left(\mathbf{x}, \mathbf{x}'\right) = \frac{\gamma}{1-\gamma} \sum_{n=0}^{\infty} K_{\gamma \pi_{\boldsymbol{\theta}}}^n \left(\mathbf{x}, \mathbf{x}'\right)$, we see that the same is true for the start-state case, if a stationary distribution $\nu^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}\right)$ exists[2].

If we now compute the natural gradient using the all-action gradient estimate, we get

$$\tilde{\nabla} J(\pi_{\boldsymbol{\theta}}) = \mathbf{M}^{-1}\left(\boldsymbol{\theta}\right) \frac{\partial J(\pi_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \mathbf{M}^{-1}\left(\boldsymbol{\theta}\right) \mathbf{F}\left(\boldsymbol{\theta}\right) \boldsymbol{w}_{\boldsymbol{\theta}} = \boldsymbol{w}_{\boldsymbol{\theta}},$$

as shown by Kakade (2001) when assuming that it is just the average Fisher information matrix. This allows us to draw the conclusion for Chapter 3.

Conclusion 4 (Policy gradient methods) *In order to do policy gradient learning, we do not need to estimate the stationary distribution $d^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}\right)$ nor the Fisher information matrix $\mathbf{F}\left(\boldsymbol{\theta}\right)$. The only parameter we need to estimate from the trials is the vector $\boldsymbol{w}_{\boldsymbol{\theta}}$.*

Due to conclusion 4, we know that the optimal way of policy gradient learning is to first approximate $\boldsymbol{w}_{\boldsymbol{\theta}}$, and then use it as a parameter update. We will refer to this algorithm

[2]Can we make the start-state Fisher information matrix case more clear? I am not confident with it.

as natural gradient learning (NGL). As we have not yet derived a method how to learn $f_{\boldsymbol{w}_\theta}(\mathbf{x}_t, \mathbf{u}_t)$, we will for now refer to it as a black box `LEARN`.

Algorithm 1 (Natural gradient learning NGL)

Initialize $\theta_0 \in \mathbb{R}^n$ *arbitrarily.*
Initialize update counter $T = 0$.
Initialize episode counter $h = 0$.
Repeat (Updates)
  Start at state $\mathbf{x}_0 \in \mathbb{X}_0$ *from distribution* $p(\mathbf{x}_0)$.
    Repeat (Sample gathering)
      Initialize time $t = 0$.
      Repeat (Episode)
        Draw $\mathbf{u}_t \in \mathbb{U}$ *from distribution* $\pi_{\theta_T}(\mathbf{u}_t | \mathbf{x}_t)$.
        Observe reward $r(\mathbf{x}_t, \mathbf{u}_t)$, *and next state* $\mathbf{x}_{t+1}$.
        Update $f_{\boldsymbol{w}_{\theta_T}}(\mathbf{x}_t, \mathbf{u}_t)$ *using LEARN*$(\mathbf{x}_t, \mathbf{u}_t, r(\mathbf{x}_t, \mathbf{u}_t), \mathbf{x}_{t+1}, h, T, t)$.
        Increment $t = t + 1$.
      Until episode ends.
      Increment episode counter $h = h + 1$.
  Until $\boldsymbol{w}_{\theta_T}$ *converges.*
  Update $\theta_{T+1} = \theta_T + \alpha_t \boldsymbol{w}_{\theta_T}$.
  Increment $T = T + 1$.
Until θ_T *converges.*

We will analyze the estimation of $\boldsymbol{w}_\theta$ (i.e., different approaches for `LEARN`) in Chapter **??**, page **??**. Before doing so, we will review a few of the previously introduced reinforcement learning methods, and relate them to previous policy gradient methods in Chapter **??**. Let us now conclude this section by showing that if the learning system converges, it converges to the optimal solution of the time-discrete LQR problem.

LQR natural gradient

Example 17 (Natural gradients for LQR) *From Example 13 (page 34), we know* $\boldsymbol{w}_\theta$ *of a Gaussian policy in linear quadratic regulation problems from Example 1 (page 9). Therefore, we have analytically determined the natural gradients of the Gaussian policy for LQR, given by*

$$\tilde{\nabla} J(\pi_\theta) = \begin{bmatrix} \boldsymbol{w}_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{k}\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right) \sigma^2 - \gamma \mathbf{A}^T \mathbf{P} \mathbf{b} \sigma^2 \\ -\frac{1}{2}\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right) \sigma^3 \end{bmatrix},$$

for simplicity, we write $\mathbf{P}$ *instead of* $\mathbf{P}_\gamma$. *Asking the question of convergence, we can determine the fixpoints in parameter space of the difference equation* $\theta_{t+1} = \theta_t + \alpha_t \tilde{\nabla} J(\pi_\theta)$, *i.e., the points where* $\tilde{\nabla} J(\pi_\theta) = 0$. *Clearly, we have*

$$\begin{aligned} 0 &= \mathbf{k}\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right) + \gamma \mathbf{A}^T \mathbf{P} \mathbf{b}, \\ &\Longrightarrow \mathbf{k} = -\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right)^{-1} \gamma \mathbf{A}^T \mathbf{P} \mathbf{b}, \\ 0 &= \frac{1}{2}\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right) \sigma^3, \\ &\Longrightarrow \sigma = 0. \end{aligned}$$

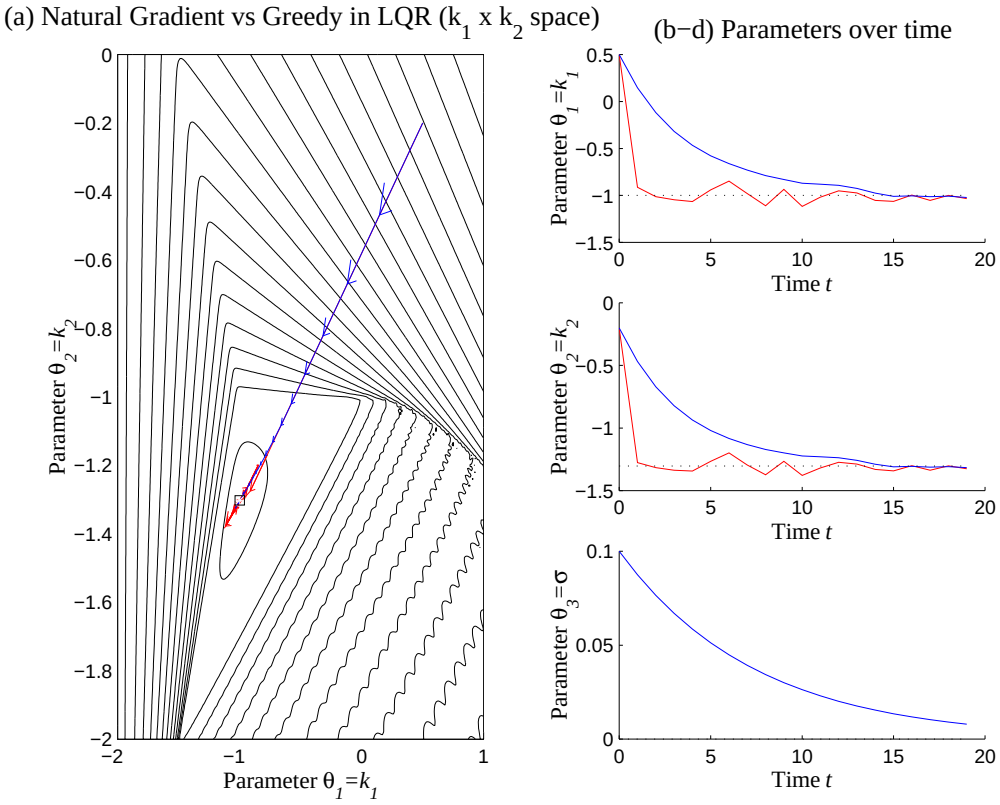


Figure 3.6: This figure shows the performance of natural gradients in comparison to the greedy approach. The value function $V^{\pi_\theta}(\mathbf{x})$ has mean-zero Gaussian distributed errors. In (a), you can see that the greedy learner goes straight close to the optimal solution, and then jumps around close to the optimum due to the noise in the value function. The natural gradient learner follows the straight path of the first greedy step, and subsequently turns towards the optimal solution. Probably better recognizeable in (b-d), the natural gradient is hardly affected by the noise of the system while the greedy approach is highly affected. In here, we use a learning rate of $\alpha_t = 0.1 \cdot J^{-1}(\pi_\theta)$ for the natural gradient. The region of stable solution, i.e., the flat plateau in the middle, can be recognized well. The instable solutions are behind all three edges.

This happens to be the optimal solution for time-discrete LQR problems. However, convergence can be achieved by $\sigma = 0$ alone. Furthermore, from the literature (Bertsekas, 2000), we know the dynamic programming update algorithm for deterministic LQR, i.e., $\mathbf{k}_{t+1} = -\left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1} \gamma \mathbf{A}^T \mathbf{P}_t \mathbf{b}$. *Let us compare this rule to the natural gradient update* $\theta_{t+1} = \theta_t + \alpha_t \tilde{\nabla} J(\pi_\theta) = \theta_t + \alpha_t \boldsymbol{w}_{\theta_t}$ *with* $\theta_t = [\mathbf{k}_t, \sigma_t]$. *For this we heuristically assume a learning rate of* $\alpha_t = -\sigma^{-2}\left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1}$. *This learning rate also equals* $\alpha_t = 2\sigma_t w_2^{-1} = 2J^{-1}(\pi_\theta)$. *It turns out that the two update rules for* $\mathbf{k}_t$ *are identical for this learning rate* α_t, *i.e.,*

$$\begin{aligned}
\mathbf{k}_{t+1} &= \mathbf{k}_t + \alpha_t \boldsymbol{w}_1 \\
&= \mathbf{k}_t - \alpha_t \left(\mathbf{k}\left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)\sigma_t^2 + \gamma \mathbf{A}^T \mathbf{P} \mathbf{b} \sigma_t^2\right), \\
&= \mathbf{k}_t - \sigma^{-2}\left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1} \left(\mathbf{k}_t \left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)\sigma_t^2 + \gamma \mathbf{A}^T \mathbf{P}_t \mathbf{b} \sigma^2\right), \\
&= \mathbf{k}_t - \mathbf{k}_t - \left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1} \gamma \mathbf{A}^T \mathbf{P}_t \mathbf{b},
\end{aligned}$$

$$
\begin{aligned}
&= -\left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1} \gamma \mathbf{A}^T \mathbf{P}_t \mathbf{b}, \\
\sigma_{t+1} &= \sigma_t + w_2 \\
&= \sigma_t - \alpha_t \frac{1}{2} \left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right) \sigma_t^3, \\
&= \sigma_t - \sigma_t^{-2} \left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right)^{-1} \frac{1}{2} \left(R + \gamma \mathbf{b}^T \mathbf{P}_t \mathbf{b}\right) \sigma_t^3, \\
&= \sigma_t - \frac{1}{2} \sigma_t, \\
&= \frac{1}{2^t} \sigma_0.
\end{aligned}
$$

Clearly, the natural gradient update is in general collinear to a dynamic programming update, and for this specific learning rate they even become equal. This shows that the most famous example for dynamic programming in optimal control is in fact also an example for natural gradient learning[3]*. Its performance can be seen in figure 3.6.*

LQR dynamic programming and natural gradient learning are equivalent

[3]The part with the learning rate has to be explored more deeply...should we generally use $\alpha_t = J^{-1}(\pi_{\boldsymbol{\theta}})$?

Bibliography

Baird, L. (1993). *Advantage updating.* (Technical Report)

Baird, L. (1998). *Gradient descent for general reinforcement learning.*

Bartlett, P. L., & Baxter, J. (2000). Estimation and approximation bounds for gradient-based reinforcement learning. *Connection Science*, *3*(28).

Baxter, J., & Bartlett, P. (1999). *Direct gradient-based reinforcement learning.*

Bellman, R. (1957). *Dynamic programming.* Princeton, NJ: Princeton University Press.

Berger, R., & Casella. (2002). *Statistical inference.* TO BE ENTERED.

Berny, A. (2000). Statistical machine learning and combinatorial optimization. *In L. Kallel, B. Naudts, and A. Rogers, editors, Theoretical Aspects of Evolutionary Computing, Lecture Notes in Natural Computing*, *0*(33).

Bertsekas, D. (2000). *Dynamic programming and optimal control.* Athena Scientific.

Bronstein, I. N., Semendjajew, K. A., Musiol, G., & Mühlig, H. (1995). *Taschenbuch der mathematik.* Verlag Harri Deutsch.

Dayan, P. (1990). To be entered. *TO BE ENTERED.*

Dorato, P., Abdallah, C., & Cerone, V. (1995). *Linear quadradic control: an introduction.* Prentice Hall.

Greensmith, E., Bartlett, P., & Baxter, J. (2001). Variance reduction techniques for gradient estimates in reinforcement learning. *Advances in Neural Information Processing Systems*, *14*(34).

Grimmett, G., & Stirzaker, D. (2001). *Probability and random processes.* Oxford University Press.

Gullapalli, V. (1991). Associative reinforcement learning of real-value functions. *SMC*, -(-).

Gullapalli, V., Franklin, J., & Benbrahim, H. (1994). Aquiring robot skills via reinforcement learning. *IEEE Control Systems*, -(39).

Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). Convergence of stochastic iterative dynamic programming algorithms. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems* (Vol. 6, pp. 703–710). Morgan Kaufmann Publishers, Inc.

Kakade, S., & Langford, J. (2002). Approximately optimal approximate reinforcement learning. *International Conference on Machine Learning.*

Konda, V. (2002). Actor-critic algorithms. *Ph.D. Thesis (MIT)*, *3*(36).

Konda, V., & Tsitsiklis, J. (2000). Actor-critic algorithms. *Advances in Neural Information Processing Systems*, *12*(35).

Konda, V., & Tsitsiklis, J. (2001). Actor-critic algorithms. *Submitted to SIAM Journal on Control and Optimisation*(38).

Mahadevan, S. (1996). To be entered. *TO BE ENTERED.*

Marbach, P., & Tsitsiklis, J. (1998). *Simulation-based optimization of markov reward processes.*

Morton, D. (2001). *Monte carlo methods in stochastic programming.* (Graduate Program in Operations Research, University of Texas in Austin)

Russel, & Norvig. (1995). *To be entered.* TO BE ENTERED.

Sutton, R. (2000). Policy gradient methods for reinforcement learning with function approximation. *Presentation at NIPS*, *12*(22).

Sutton, R., & Barto, A. (1998). *Reinforcement learning: an introduction.* MITPRESS.

Sutton, R., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, *12*(22).

Sutton, R., McAllester, D., Singh, S., & Mansour, Y. (2001). *Comparing policy gradient methods.* (Unfinished paper)

Weaver, L., & Tao, N. (2001a). The optimal reward baseline for gradient-based reinforcement learning. *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference*, *17*(29).

Weaver, L., & Tao, N. (2001b). The variance minimizing constant reward baseline for gradient-based reinforcement learning. *Technical Report ANU*, -(30).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(23).

Williams, R. J., & Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, *3*(24).

Appendix A

Proofs for the Examples

A.1 Linear Quadratic Regulation Examples

A.1.1 Expected Return Derivation

Average Reward Case

Start-State Case

A.1.2 Value Function Derivation

Average Reward Case

Start-State Case

In this appendix section, we derive the result that the state value function of the Gaussian policy

$$\pi_{\boldsymbol{\theta}}\left(u\,|\mathbf{x}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(u-\mathbf{k}^T\mathbf{x}\right)^2\right) \tag{A.1}$$

for a given discrete-time LQR problem with matrices $\mathbf{A}, \mathbf{b}, R$, and $\mathbf{Q}$ can be defined as

$$\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}\right) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \frac{1}{2}\frac{1}{1-\gamma}\left(R+\gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2$$

for the discrete-time, undiscounted start-state case where $\mathbf{P}$ is the solution of

$$\mathbf{P} = \left[\mathbf{Q}+\gamma\mathbf{A}^T\mathbf{P}\mathbf{A}+\gamma\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}+\gamma\mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T+\gamma\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T+\mathbf{k}R\mathbf{k}^T\right].$$

Similarly, we have the state-action value function

$$\begin{aligned}\hat{Q}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t,\mathbf{u}_t\right) = \frac{1}{2}\begin{bmatrix}\mathbf{x}^T, & u\end{bmatrix}\begin{bmatrix}\mathbf{Q}+\gamma\mathbf{A}^T\mathbf{P}\mathbf{A} & \gamma\mathbf{A}^T\mathbf{P}\mathbf{b}\\ \gamma\mathbf{b}^T\mathbf{P}\mathbf{A} & \mathbf{R}+\gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ u\end{bmatrix}\\ + \frac{1}{2}\frac{\gamma}{1-\gamma}\left(R+\gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.\end{aligned}$$

Proof. In order to obtain the value function, we have solve the Bellman equation for the *undiscounted case* of this specific problem, i.e.,

$$\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t\right) = \int_{\mathbb{U}}\pi_{\boldsymbol{\theta}}(\mathbf{x}_t,u)\hat{Q}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t,u\right)du.$$

As this becomes a nonlinear, vectorized Fredholm integral equation of second kind with a scalar value function, this is difficult to solve. However, we can rewrite our system equation

to

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{b}u_t, \\ &= \mathbf{A}\mathbf{x}_t + \mathbf{b}\left(\bar{u}_t + w_t\right), \\ &= \mathbf{A}\mathbf{x}_t + \mathbf{b}\bar{u}_t + \mathbf{b}w_t, \\ &= \mathbf{A}\mathbf{x}_t + \mathbf{b}\mathbf{k}^T\mathbf{x} + \mathbf{b}w_t,\end{aligned}$$

where w_t are 'errors' which have a zero mean and are drawn from a mean-zero Gaussian distribution

$$p\left(w = w' \,|\pi_{\boldsymbol{\theta}}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{w^2}{2\sigma^2}\right).$$

The expectations of the 'errors' w_t are given by

$$\begin{aligned}\bar{w} &= E\left\{w_t\right\} = 0, \\ W &= E\left\{w_t w_t\right\} = \sigma^2.\end{aligned}$$

This slight change in problem notation causes a large change in the difficulty of derivation since the structure of the value function for this problem is well-known (?). It is given by Stengel (?) as

$$\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t\right) = \frac{1}{2}\mathbf{x}_t^T\mathbf{P}\mathbf{x}_t + v_t,$$

with

$$v_t = \Delta v_t + \gamma v_{t+1},$$

for the discrete-time case. For the continuous time case, a similar formulation with integrals is given in (?).

In order to do this, we have to define a state function $\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}\right)$ so that we can solve the upper equation. Instead of defining $\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}\right)$ directly, we define $\hat{Q}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t, u_t\right)$ first and use it to derive the policy value function.

$$\begin{aligned}\hat{Q}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t, u\right) &= E\left\{r\left(\mathbf{x}_t, u\right) + \gamma\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_{t+1}\right) + v_{t+1}\right\}, \\ &= \frac{1}{2}\left(\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t + Ru^2\right) + \frac{1}{2}\gamma\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right)^T\mathbf{P}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) + v_{t+1}.\end{aligned}$$

We have to integrate over the policy:

$$\begin{aligned}\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t\right) &= \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\hat{Q}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{x}_t, u\right) du, \\ &= \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\left(r\left(\mathbf{x}_t, u\right) + \gamma\hat{V}^{\pi_{\boldsymbol{\theta}}}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) + v_{t+1}\right) du, \\ &= \int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\left(\frac{1}{2}\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t + \frac{1}{2}Ru^2\right. \\ &\left. + \frac{1}{2}\gamma\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right)^T\mathbf{P}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) + \gamma v_{t+1}\right) du.\end{aligned}$$

Clearly, we can integrate the first term to

$$\int_{\mathbb{U}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\frac{1}{2}\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t du = \frac{1}{2}\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t.$$

The second term can be derived using integrals from Bronstein et al. (1995) if we assume that $\mathbb{U} = \mathbb{R}$, i.e.,

$$
\begin{aligned}
&\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\frac{1}{2}Ru^2 du \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(u - \mathbf{k}^T\mathbf{x}_t\right)^2\right)\frac{1}{2}Ru^2 du, \\
&= \frac{1}{2}\int_{-\infty}^{\infty} \underbrace{\frac{1}{\sqrt{2\sigma^2}}}_{=\frac{dv}{du}}\frac{1}{\sqrt{\pi}} \exp\underbrace{\left(-\frac{1}{2\sigma^2}\left(u - \mathbf{k}^T\mathbf{x}_t\right)^2\right)}_{=-v^2} R \underbrace{u^2}_{=\left(\sqrt{2}\sigma v - \mathbf{k}^T\mathbf{x}_t\right)^2} du, \\
&= \frac{1}{2}\int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} \exp\left(-v^2\right) R\left(\sqrt{2}\sigma v - \mathbf{k}^T\mathbf{x}_t\right)^2 du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} \exp\left(-v^2\right) R\left(2\sigma^2 v^2 - 2\sqrt{2}\sigma v\mathbf{k}^T\mathbf{x}_t + \left(\mathbf{k}^T\mathbf{x}_t\right)^2\right) du, \\
&= \frac{2\sigma^2 R}{\sqrt{\pi}}\frac{1}{2}\underbrace{\int_{-\infty}^{\infty} \exp\left(-v^2\right) v^2 dv}_{=\sqrt{\pi}/2} - \frac{R2\sqrt{2}\sigma\mathbf{k}^T\mathbf{x}_t}{\sqrt{\pi}}\frac{1}{2}\underbrace{\int_{-\infty}^{\infty} \exp\left(-v^2\right) v dv}_{=0} \\
&+ \frac{1}{2}\frac{R\left(\mathbf{k}^T\mathbf{x}_t\right)^2}{\sqrt{\pi}}\underbrace{\int_{-\infty}^{\infty} \exp\left(-v^2\right) dv}_{=\sqrt{\pi}}, \\
&= \frac{1}{2}R\sigma^2 + \frac{1}{2}R\left(\mathbf{k}^T\mathbf{x}_t\right)^2, \\
&= \frac{1}{2}R\sigma^2 + \frac{1}{2}\mathbf{x}_t^T\mathbf{k}R\mathbf{k}^T\mathbf{x}_t.
\end{aligned}
$$

The integral $\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right)^T \mathbf{P}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) du$ can be split into pieces and subsequently solved to

$$
\begin{aligned}
\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\mathbf{x}_t^T\mathbf{A}^T\mathbf{P}\mathbf{A}\mathbf{x}_t du &= \mathbf{x}_t^T\mathbf{A}^T\mathbf{P}\mathbf{A}\mathbf{x}_t, \\
\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)u\mathbf{b}^T\mathbf{P}\mathbf{A}\mathbf{x}_t du &= \mathbf{x}_t^T\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}\mathbf{x}_t, \\
\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\mathbf{x}_t\mathbf{A}^T\mathbf{P}\mathbf{b}u du &= \mathbf{x}^T\mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T\mathbf{x}_t, \\
\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)u\mathbf{b}^T\mathbf{P}\mathbf{b}u du &= \mathbf{b}^T\mathbf{P}\mathbf{b}\sigma^2 + \mathbf{x}_t^T\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T\mathbf{x}_t,
\end{aligned}
$$

which gives us a full integral of

$$
\begin{aligned}
&\frac{1}{2}\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u)\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right)^T \mathbf{P}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) du, \\
&= \frac{1}{2}\left(\mathbf{x}_t^T\mathbf{A}^T\mathbf{P}\mathbf{A}\mathbf{x}_t + \mathbf{x}_t^T\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}\mathbf{x}_t + \mathbf{x}^T\mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T\mathbf{x}_t + \mathbf{b}^T\mathbf{P}\mathbf{b}\sigma^2 + \mathbf{x}_t^T\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T\mathbf{x}_t + \mathbf{x}_t^T\mathbf{k}R\mathbf{k}^T\mathbf{x}_t\right).
\end{aligned}
$$

Similarly, we get

$$\int_{\mathbb{R}} \pi_{\boldsymbol{\theta}}(\mathbf{x}_t, u) v_{t+1} du = v_{t+1}.$$

Let us now determine the unknown parameters $\mathbf{P}$ and v_{t+1}. By coefficient comparison, this gives us the equation

$$\begin{aligned}
\frac{1}{2}\mathbf{x}_t^T \mathbf{P} \mathbf{x}_t &= \frac{1}{2}\mathbf{x}_t^T \left[\mathbf{Q} + \mathbf{A}^T\mathbf{P}\mathbf{A} + \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \mathbf{k}R\mathbf{k}^T\right]\mathbf{x}_t, \\
\mathbf{P} &= \left[\mathbf{Q} + \gamma\mathbf{A}^T\mathbf{P}\mathbf{A} + \gamma\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} + \gamma\mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \gamma\mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \mathbf{k}R\mathbf{k}^T\right]
\end{aligned}$$

which we have to solve for $\mathbf{P}$, and

$$\begin{aligned}
v_t &= \frac{1}{2}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2 + \gamma v_{t+1}, \\
v_t &= \Delta v + \gamma v_{t+1}, \\
\Delta v &= \frac{1}{2}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.
\end{aligned}$$

which gives us Δv. We can clearly derive from

$$\begin{aligned}
v_t &= \Delta v + \gamma v_{t+1} \\
&= \Delta v + \Delta v + v_{t+2} \\
&= \underbrace{\Delta v + \gamma\Delta v + \ldots + \gamma^{t_f - t}\Delta v}_{=t_f - t} + v_{t_f} \\
&= \frac{1 + \gamma^{t_f - t + 1}}{1 - \gamma}\Delta v + v_{t_f} \\
&= \frac{1 + \gamma^{t_f - t + 1}}{1 - \gamma}\left(R + \frac{1}{2}\gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2 + v_{t_f}
\end{aligned}$$

Since the final state is the end of any trajectory we have $v_{t_f} = 0$. This gives us

$$v_t = \frac{1}{2}\frac{1 + \gamma^{t_f - t + 1}}{1 - \gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.$$

We now have all components of the value function for this policy for state $\mathbf{x}_t$ at time t for the undiscounted start-state case, i.e.,

$$\hat{V}^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_t) = \frac{1}{2}\mathbf{x}_t^T\mathbf{P}\mathbf{x}_t + \frac{1}{2}\frac{1 + \gamma^{t_f - t + 1}}{1 - \gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.$$

Clearly, the undiscounted formulation would not always exist as it grows with time. For $t_f \to \infty$, we get

$$\hat{V}^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_t) = \frac{1}{2}\mathbf{x}_t^T\mathbf{P}\mathbf{x}_t + \frac{1}{2}\frac{1}{1 - \gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.$$

Similarly, we have

$$\begin{aligned}
\hat{Q}^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_t, u) &= E\left\{r(\mathbf{x}_t, u) + \gamma\hat{V}^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_{t+1}) + \gamma v_{t+1}\right\}, \\
&= \frac{1}{2}\left(\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t + Ru^2\right) + \frac{1}{2}\gamma\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right)^T\mathbf{P}\left(\mathbf{A}\mathbf{x}_t + \mathbf{b}u\right) + \gamma v_{t+1}, \\
&= \frac{1}{2}\begin{bmatrix} \mathbf{x}_t^T & \mathbf{u}_t^T \end{bmatrix}\begin{bmatrix} \mathbf{Q} + \gamma\mathbf{A}^T\mathbf{P}\mathbf{A} & \gamma\mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma\mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix}\begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix} \\
&\quad + \frac{1}{2}\frac{\gamma}{1 - \gamma}\left(R + \gamma\mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2.
\end{aligned}$$

■

A.1.3 Advantage Function Derivation

Average Reward Case

Start-State Case

In this appendix section, we derive the result that advantage function for linear quadratic regulation (LQR) problems under policy $\pi_{\boldsymbol{\theta}}$ is given by

$$A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\begin{bmatrix} \mathbf{x}^T, & u \end{bmatrix} \mathbf{H} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} - \frac{1}{2}\left(R + \gamma \mathbf{b}^T \mathbf{P} \mathbf{b}\right)\sigma^2,$$

with

$$\mathbf{H} = \begin{bmatrix} -2\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \gamma \mathbf{k}\left(\mathbf{b}^T\mathbf{P}\mathbf{b} + R\right)\mathbf{k}^T & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix},$$

for the start-state case.

Proof. We have by definition

$$\begin{aligned} A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) &= r(\mathbf{x}, u) + V^{\pi_{\boldsymbol{\theta}}}(\mathbf{A}\mathbf{x} + \mathbf{b}u) - V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}), \\ &= Q^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) - V^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}), \end{aligned}$$

This gives us

$$\begin{aligned} A^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u}) = \frac{1}{2}&\begin{bmatrix} \mathbf{x}_t^T & \mathbf{u}_t^T \end{bmatrix} \begin{bmatrix} 2\mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & 2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix} \\ &+ \frac{1}{2}\frac{\gamma}{1-\gamma}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2 - \frac{1}{2}\mathbf{x}_t^T\mathbf{P}\mathbf{x}_t - \frac{1}{2}\frac{1}{1-\gamma}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2. \end{aligned}$$

We can now simplify the terms. Let us start with the outer constant, i.e.,

$$\begin{aligned} &\frac{1}{2}\frac{\gamma}{1-\gamma}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2 - \frac{1}{2}\frac{1}{1-\gamma}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2, \\ &= \frac{1}{2}\frac{\gamma - 1}{1-\gamma}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2, \\ &= -\frac{1}{2}\left(2R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2. \end{aligned}$$

Since $\frac{1}{2}\mathbf{x}_t^T\mathbf{P}\mathbf{x}_t$ does not contain any u's, we can move it into the matrix, i.e.,

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix} - \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix} \end{aligned}$$

Therefore, we have to simplify the upper-left element by

$$\begin{aligned} &\mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P}, \\ &= \mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} - \left[\mathbf{Q} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{A} + \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T + \mathbf{k}R\mathbf{k}^T\right], \\ &= -\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T - \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T - \mathbf{k}R\mathbf{k}^T, \\ &= -2\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{b}\mathbf{k}^T - \mathbf{k}R\mathbf{k}^T. \end{aligned}$$

The last step is only possible due the symmetry of the formulation, i.e.,

$$\gamma \mathbf{A}^T\mathbf{P}\mathbf{b}\mathbf{k}^T = \gamma \boldsymbol{\nu}\mathbf{k}^T = \gamma \mathbf{k}\boldsymbol{\nu}^T = \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}.$$

By reinserting this element into matrix $\mathbf{H}$ we get

$$\mathbf{H} = \begin{bmatrix} -2\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \gamma \mathbf{k}\left(\mathbf{b}^T\mathbf{P}\mathbf{b} + R\right)\mathbf{k}^T & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix}.$$

This has proved our example. ■

A.1.4 Compatible Function Approximation

Let us now show that the expected TD(0) advantage can be represented by the compatible function approximation at least for this problem. We analyze all terms

$$\begin{aligned} -w_2\frac{1}{\sigma} &= -\frac{1}{2}\left(R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2, \\ w_2 &= \frac{1}{2}\left(R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^3. \end{aligned}$$

Let us now look this into our main equation

$$\begin{bmatrix} -\mathbf{k}\boldsymbol{w}_1^T\frac{1}{\sigma^2} + w_2\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T & \frac{1}{2}\boldsymbol{w}_1\frac{1}{\sigma^2} - \mathbf{k}w_2^T\frac{1}{\sigma^3} \\ \frac{1}{2}\boldsymbol{w}_1^T\frac{1}{\sigma^2} - w_2\frac{1}{\sigma^3}\mathbf{k}^T & \frac{1}{\sigma^3}w_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -2\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \gamma \mathbf{k}\left(\mathbf{b}^T\mathbf{P}\mathbf{b} + R\right)\mathbf{k}^T & \gamma \mathbf{A}^T\mathbf{P}\mathbf{b} \\ \gamma \mathbf{b}^T\mathbf{P}\mathbf{A} & R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b} \end{bmatrix}.$$

This gives us three equations due to the symmetry of both matrices:

$$\begin{aligned} -\mathbf{k}\boldsymbol{w}_1^T\frac{1}{\sigma^2} + w_2\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T &= -\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} - \frac{1}{2}\gamma \mathbf{k}\left(\mathbf{b}^T\mathbf{P}\mathbf{b} + R\right)\mathbf{k}^T, \\ \frac{1}{2}\boldsymbol{w}_1\frac{1}{\sigma^2} - \mathbf{k}w_2\frac{1}{\sigma^3} &= \frac{1}{2}\gamma \mathbf{A}^T\mathbf{P}\mathbf{b}, \\ \frac{1}{\sigma^3}w_2 &= \frac{1}{2}R + \frac{1}{2}\gamma \mathbf{b}^T\mathbf{P}\mathbf{b}. \end{aligned}$$

Clearly, the third of these equations is equivalent with our previous one. So we just have to analyze the other two.

$$\begin{aligned} \frac{1}{2}\boldsymbol{w}_1\frac{1}{\sigma^2} - \mathbf{k}w_2\frac{1}{\sigma^3} &= \frac{1}{2}\gamma \mathbf{A}^T\mathbf{P}\mathbf{b}, \\ \boldsymbol{w}_1\frac{1}{\sigma^2} &= 2\mathbf{k}w_2\frac{1}{\sigma^3} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}, \\ &= \mathbf{k}\left(R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^3\frac{1}{\sigma^3} + \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}, \\ \boldsymbol{w}_1 &= \mathbf{k}\left(R + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^2 + \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}\sigma^2. \end{aligned}$$

Let us now simplify the right side of the last missing equation:and reinsert it into the equation (, i.e., due to symmetry). We now want to see whether it does not contradict the previous equations. Therefore we insert the parameters into the left side of the last equation:

$$\begin{aligned} &- \mathbf{k}\boldsymbol{w}_1^T\frac{1}{\sigma^2} + w_2\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T, \\ &= -\mathbf{k}\left(\mathbf{k}\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right) + \gamma \mathbf{A}^T\mathbf{P}\mathbf{b}\right)^T\sigma^2\frac{1}{\sigma^2} + \frac{1}{2}\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\sigma^3\frac{1}{\sigma^3}\mathbf{k}\mathbf{k}^T, \\ &= -\mathbf{k}\mathbf{k}^T\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right) - \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A} + \frac{1}{2}\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\mathbf{k}\mathbf{k}^T, \\ &= -\frac{1}{2}\mathbf{k}\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\mathbf{k}^T - \gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}, \\ &= \frac{1}{2}\left(-\mathbf{k}\left(\mathbf{R} + \gamma \mathbf{b}^T\mathbf{P}\mathbf{b}\right)\mathbf{k}^T - 2\gamma \mathbf{k}\mathbf{b}^T\mathbf{P}\mathbf{A}\right). \end{aligned}$$

Clearly, this shows that the expected TD(0) advantage can be represented in the compatible function approximator.

A.1.5 All-Action Matrix Derivation

53

A.1.6 ...

A.2 Discrete State- and Action Example

A.3 All-Action Matrix

Double k-equation: The double k-equation is solved here

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{1}{\pi(\mathbf{x},u)} \frac{\partial \pi(\mathbf{x},u)}{\partial k_i} \frac{\partial \pi(\mathbf{x},u)}{\partial k_j} du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}v^2\right) v^2 \frac{x_i x_j}{\sigma} \frac{1}{\sigma} du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}v^2\right) v^2 \frac{x_i x_j}{\sigma} dv, \\
&= \frac{x_i x_j}{\sigma^2}. \qquad \text{(A.2)}
\end{aligned}$$

Single k, single σ-equation: The single k, single σ-equation is solved here

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{1}{\pi(\mathbf{x},u)} \frac{\partial \pi(\mathbf{x},u)}{\partial \sigma} \frac{\partial \pi(\mathbf{x},u)}{\partial k_j} du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}v^2\right) \frac{x_j}{\sigma} v\left(v^2-1\right) \frac{1}{\sigma} du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}v^2\right) \frac{x_j}{\sigma} v\left(v^2-1\right) dv, \\
&= 0. \qquad \text{(A.3)}
\end{aligned}$$

Double σ-equation: The double σ-equation is solved here

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{1}{\pi(\mathbf{x},u)} \left(\frac{\partial \pi(\mathbf{x},u)}{\partial \sigma}\right)^2 du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}v^2\right) \frac{1}{\sigma} \left(v^2-1\right)^2 \frac{1}{\sigma} du, \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}v^2\right) \frac{1}{\sigma} \left(v^2-1\right)^2 dv, \\
&= \frac{2}{\sigma^2}. \qquad \text{(A.4)}
\end{aligned}$$