Mirror Descent Methods in Reinforcement Learning

Master Thesis Mathematics

Reza Zolnouri

4 December 2024

Supervised by Prof. Dr. Semih Cayci

Submitted to Chair of Mathematics of Information Processing,
RWTH Aachen University

First Examiner: Prof. Dr. Semih Cayci

Second Examiner: Prof. Dr. Benjamin Berkels

Contents

Introduction

Reinforcement Learning (RL) has emerged as a powerful framework for modeling and solving complex sequential decision-making problems across various domains, including robotics, game-playing, and autonomous systems (see, for example, Silver et al. [2016], Levine et al. [2016], Bojarski et al. [2016]. Reinforcement learning involves developing strategies (policies) that dictate the selection of actions based on various states, with the objective of maximizing cumulative reward or minimizing cumulative cost. Unlike supervised learning, where specific actions are provided, the agent must autonomously identify which actions lead to the lowest cost through experimentation. In complex and demanding scenarios, the consequences of actions extend beyond immediate costs, influencing future states and, consequently, all future costs. These two fundamental aspects—exploratory trial-and-error and the propagation of delayed costs—are key characteristics that set reinforcement learning apart from other learning paradigms (Sutton and Barto [2020])

One of the central problems in RL is the development of optimization methods that can efficiently find optimal or near-optimal policies. Traditional approaches, such as value iteration and policy iteration (Puterman [1994]), often become computationally infeasible in large state and action spaces. Among various algorithms employed in RL nowadays, Policy Gradient Methods (Williams [1992], Sutton et al. [1999], Konda and Tsitsiklis [2000], Kakade [2001]) are a class of algorithms that directly optimize parameterized policies by descending the gradient of some scalar performance measure (for example, the expected cumulative cost) with respect to the policy parameters (Sutton and Barto [2020]). These gradient-based algorithms optimize over a class of parameterized policies and have become a popular choice for RL problems in both theoretical and practical applications (Alfano et al. [2024]). In recent years, Mirror Descent (MD) techniques have gained attention due to their ability to handle constrained optimization problems and exploit the geometry of the parameter space (Tomar et al. [2020], Agarwal et al. [2021]).

Mirror Descent (A. Nemirovski and D. B. Yudin [1983]) is a first-order algorithm in constrained convex optimization that generalizes the traditional gradient descent by incorporating a distance-generating function, known as the mirror map, to adaptively scale updates based on the geometry of the problem (Tomar et al. [2020]). This method exhibits an efficiency estimate that is mildly dependent on the decision variables dimension and thus suitable for solving very large-scale optimization problems (Beck and Teboulle [2003]). In the context of RL, Mirror Descent methods have been proposed to efficiently optimize policies by leveraging Bregman divergences (L. Bregman [1967]) as regularizers. These methods have shown promising theoretical convergence properties and practical performance.

Historically, the concept of mirror descent was introduced by A. Nemirovski and D.

B. Yudin [1983] as an extension of gradient descent for solving high-dimensional convex optimization problems. Mirror descent employs Bregman divergences to adaptively scale gradient updates, allowing for efficient optimization in non-Euclidean geometries. Beck and Teboulle [2003] further developed mirror descent methods and applied them to various optimization problems, establishing convergence properties and practical algorithms. Moreover, the stochastic version of mirror descent was later explored by Nemirovski et al. [2009], providing strong convergence guarantees in stochastic optimization settings.

In the field of reinforcement learning, Numerous traditional algorithms for Discrete Markov Decision Processes (DMDPs) are grounded in the dynamic programming framework introduced by Bellman [1957]. These algorithms involve value iteration, policy iteration, temporal difference learning, and Q-learning (see, for example, Puterman [1994], Bertsekas and Tsitsiklis [1996], Sutton and Barto [2020] Gerald Tesauro and Keith [1995]). The analysis of these methods in the tabular setting primarily relies on the contraction property of the Bellman operator, a characteristic that poses significant challenges when attempting to extend these algorithms to scenarios involving nonlinear function approximation and policy parametrization. In contrast, policy gradient methods focus on identifying local minima of expected value functions. This approach renders them applicable to any differentiable policy parametrization and facilitates seamless integration with function approximation techniques. Notably, policy gradient methods exhibit robust performance when parameterized using modern deep neural networks (Schulman et al. [2015], Schulman et al. [2017]), highlighting their adaptability and effectiveness in complex, high-dimensional environments (Xiao [2022]).

Mirror descent techniques have been employed to address policy optimization challenges. Duchi et al. [2011] applied mirror descent algorithms to online learning and RL problems, demonstrating their effectiveness in handling large-scale decision-making tasks. Geist et al. [2019] proposed a general theory of regularized MDPs policy optimization, and Kakade [2001] introduced the Natural Policy Gradient (NPG) method, which adjusts policy updates using the Fisher information matrix to account for the underlying geometry of the policy space. This method has been influential in developing algorithms that achieve better performance and convergence rates compared to standard policy gradient methods. Agarwal et al. [2021] improved the NPG convergence rate to $O\left(\frac{1}{k}\right)$, and later, by using entropy regularization, Cen et al. [2022] show that the Entropy-regularized NPG method has linear (geometric) convergence, and also for un-regularized NPG framework, Khodadadian et al. [2021] shows that the NPG method can obtain linear convergence with an adaptive step-size rule. Entropy-regularized NPG has also been studied by Cayci et al. [2024a] in the Linear Function Approximation regime, and the linear convergence guarantees have been established up to a function approximation error.

In 2022, Guanghui Lan builds upon these foundational concepts and introduced a seminal algorithm called Policy Mirror Descent in his paper "Policy Mirror Descent for reinforcement learning: linear convergence, new sampling complexity, and generalized problem classes" (Lan [2022]), which can be viewed as an adaptation of the mirror descent algorithm to the realm of policy optimization (Zhan et al. [2023]). This work serves as the primary reference for the current study. By introducing approximation schemes (APMD) and thoroughly analyzing their impact on convergence, Lan provides valuable insights into the practical implementation of mirror descent methods within RL frameworks. Furthermore, this approach has been further developed by Li and Lan in their study on Homotopic Policy Mirror Descent Li et al. [2022], and it has also been adapted to general state spaces by Ju and Lan [2022]. These extensions highlight the versatility and scalability of mirror descent techniques in tackling complex RL environments, reinforcing their significance in advancing policy optimization methodologies. More recently, Zhan et al. [2023] extended the framework of Lan [2022] to accommodate a broader class of convex regularizers, including those that are nonsmooth. In contrast to Lan's work that considered a generic Bregman divergence, Zhan et al. algorithm selects the Bregman divergence adaptively in cognizant of the regularizer, which leads to complementary perspectives and insights.

This thesis aims to conduct a thorough analysis of mirror descent techniques in RL, with a focus on PMD, APMD, and Entropy-regularized NPG methods. By breaking down the theoretical underpinnings of these algorithms and examining their convergence properties, we seek to provide insights into their relative strengths and limitations. Additionally, we investigate the impact of different regularizers on the convergence behavior of these methods through numerical experiments.

Objectives of the Thesis

1. **Theoretical Analysis**: Providing fundamental theoretical backgrounds To analyze PMD, Stochastic PMD (SPMD) and APMD as proposed by Lan [2022], and to investigate Entropy-regularized Natural Policy Gradient method by Cen et al. [2022].

2. **Comparative Study**: To establish the convergence guarantees and characterize the distinct behaviors of PMD, SPMD, APMD, and Entropy-Regularized Natural Policy Gradient (NPG) methods, as well as to explore their potential interrelationships

3. **Regularization Effects**: Investigating the Impact of Convex Regularizers on Optimization Performance and Exploration Dynamics

4. **Numerical Experiments**: To conduct experiments that validate the performance of the algorithms and provide empirical evidence of the impact of different regularizers on convergence behaviors.

Thesis Structure

The thesis is organized as follows:

- **Chapter 1: Theoretical Background**
  This chapter introduces the fundamental concepts necessary for understanding the optimization algorithms discussed in the thesis. It covers projected subgradient descent, Bregman divergence, strong convexity, the mirror descent algorithm, and variational inequalities.

- **Chapter 2: Reinforcement Learning Setup**
  We provide a formal definition of Markov Decision Processes (MDPs) and discuss key concepts in RL, such as value functions, policies, and Bellman equations. The chapter also introduces the regularized RL framework and presents essential lemmas that underpin the analysis of policy optimization methods.

- **Chapter 3: Policy Mirror Descent and Stochastic Policy Mirror Descent**
  This chapter introduces the deterministic and stochastic PMD algorithm and establishes its convergence analysis, addressing the challenges of stochasticity in estimates and further assumptions on maintaining linear convergence. The concept of distribution mismatch and its impact on the exploration dynamics is also introduced.

- **Chapter 4: Approximate Policy Mirror Descent**
  We explore the APMD algorithm as a variant of PMD by incorporating a perturbation term, detailing its setup and convergence analysis.

- **Chapter 5: Entropy-Regularized Natural Policy Gradient**
  This chapter focuses on the NPG method with entropy regularization introduced by Cen et al. [2022]. We discuss policy parameterization techniques to set up the Entropy-regularized NPG framework and point out the convergence guarantees. The connections between PMD and Entropy-regularized NPG are analyzed, revealing how Entropy-regularized NPG can be considered as an instance of PMD.

- **Chapter 6: Numerical Experiment**
  This chapter presents an experimental investigation into the impact of various regularizers and the selection of different hyperparameters on the convergence of value functions and policies. Additionally, it examines the exploratory behavior of these models in both deterministic and stochastic settings. The chapter details the experimental setup, implementation specifics, and discussion of the simulation results obtained. at the end, we summarize the key findings of the thesis, reflect on the implications of the results.

Contributions

This thesis contributes to the field of reinforcement learning by providing:

- A concise overview of foundational theoretical tools and key concepts.
- A comprehensive literature review and theoretical comparison of PMD, SPMD, APMD, and Entropy-regularized NPG methods, including enhancements to existing proofs and arguments while uncovering broader connections within the field. Additionally, an analysis of the performance evaluation distribution over states and its impact on convergence guarantees, will be discussed in Section (3.4), which has not been detailed in Lan [2022].
- Establishing a theoretical connection between Approximate Policy Mirror Descent and Entropy-Regularized Natural Policy Gradient methods and provide a comprehensive analysis that, to the best of my knowledge, has not been previously available in the literature."
- Insights into the role of regularization in policy optimization algorithms both in theory and numerical experiments.
- Comparative performance analysis with different regularizers through numerical experiments. We compared the performance of APMD using various ℓ^p regularizers (for different values of p) and the log-barrier regularizer. This analysis reveals how different regularizations behave and the efficiency of the APMD algorithm. These findings provide practical guidance for selecting appropriate regularizers in PMD applications.
- A foundation for future research on improving policy optimization techniques using mirror descent frameworks.

By advancing the understanding of mirror descent techniques in reinforcement learning, this work aims to inform the development of more efficient and robust algorithms capable of tackling complex decision-making problems.

Chapter 1

Theoretical background

1.1 Projected subgradient descent

Gradient descent is an optimization algorithm used to minimize a differentiable function by iteratively moving toward the steepest descent direction of that function, i.e., starting at some point x_0 and successively generating vectors $x_1, x_2, \ldots$, that the target function is decreasing at every iteration (D. P. Bertsekas [1999]). The fundamental algorithms for minimizing smooth functions, the gradient, as well as Newton's algorithms, are based on linear or quadratic approximation of the function given by the first terms of a Taylor series. However, this method is unfeasible for nondifferentiable functions, for such a function cannot be well approximated either by a linear or by a quadratic function. A variant of the gradient descent scheme is the projected subgradient descent, which is an optimization strategy used to solve constrained convex optimization problems. Unlike traditional gradient descent, which operates in an unconstrained space, projected subgradient descent ensures that the solution stays within a feasible set by incorporating a projection step. After each subgradient update using an arbitrary subgradient of the function, the algorithm projects the iterated point back onto the feasible set if it falls outside (Boris T. Polyak [1987]). This method is particularly useful for problems where the feasible region is non-smooth, or the gradient may not exist everywhere, as it relies on subgradients, which generalize the concept of gradients to non-differentiable functions. Denoting the Euclidean norm by $\|\cdot\|_2$, and the convex compact set of constraints by $\mathcal{X} \subset \mathbb{R}^n$, We define the projection operator $\Pi_{\mathcal{X}}$ on $\mathcal{X}$ (Boyd and Vandenberghe [2014]) by

$$\Pi_{\mathcal{X}}(x) = \arg\min_{y \in \mathcal{X}} \|x - y\|_2, \tag{1.1.1}$$

To minimize a convex function f on $\mathbb{R}^n$, subgradient descent algorithm involves the following steps:

1. Begin at an initial point $x_1 \in \mathbb{R}^n$.

2. At each iteration t, choose a subgradient $g_t \in \partial f(x_t)$ for the current point $x_t \in \mathcal{X}$.

3. Select an appropriate stepsize η_t, which controls the magnitude of the update at each iteration.

4. Proceeds by iterating the following update equation:

$$x_{t+1} = x_t - \eta_t g_t$$

This process continues iteratively until a stopping criterion is met, such as reaching a sufficiently small subgradient norm or a predefined number of iterations.

To make sure that the updated point lies in $\mathcal{X}$, a projection step could be performed. This gives the projected subgradient descent algorithm, which iterates the following equations for $t \geq 1$:

$$\begin{aligned} y_{t+1} &= x_t - \eta_t g_t \\ x_{t+1} &= \Pi_{\mathcal{X}}(y_{t+1}). \end{aligned} \tag{1.1.2}$$

An equivalent form of making this update rule is to take the following proximal steps (Beck and Teboulle [2003]):

$$\begin{aligned} y_{t+1} &= \arg\min_y \left(f(x_t) + \langle g_t, y - x_t \rangle + \frac{1}{2\eta_t} \|y - x_t\|_2^2 \right) \\ x_{t+1} &= \Pi_{\mathcal{X}}(y_{t+1}), \end{aligned} \tag{1.1.3}$$

Here, an interpretation of projected subgradient descent is to take a small step in the direction that minimizes the local first-order Taylor approximation of f (Boris T. Polyak [1987]) while ensuring the updated point remains within the feasible region. At each iteration, the method seeks to balance moving towards optimality and staying close to the current iterate. This balance is governed by the stepsize η_t, which is adjusted at each iteration to control the size of the update.

1.2 Bregman divergence and strong convexity

A generalization of the concept of distance between two points is the Bregman divergence, also referred to as Bregman distance (L. Bregman [1967]). Unlike a metric, a Bregman divergence does not necessarily satisfy properties such as symmetry or triangle inequality. Formally, a Bregman divergence is defined using a differentiable function given over a

convex set $\Phi : \Omega \to \mathbb{R}$, referred to as the distance-generating function, which is considered to be strongly convex, i.e.,

$$\Phi(x) > \Phi(y) + \langle \nabla\Phi(y), x - y\rangle \quad \forall x, y \in \Omega. \tag{1.2.1}$$

This function serves as the foundation for constructing the divergence and provides a means of measuring the discrepancy between points in a way that generalizes the notion of distance in non-Euclidean spaces.

Specifically, for two points $x, y \in \Omega$, the Bregman divergence $D_\Phi(x, y)$ is given by (L. Bregman [1967]):

$$D_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y\rangle, \tag{1.2.2}$$

Here, $\nabla\Phi(y)$ represents the gradient of Φ evaluated at y. The strong convexity of Φ ensures that the divergence is non-negative, i.e., $D_\Phi(x, y) \geq 0$, with equality if and only if $x = y$. However, Bregman distance is actually not necessarily a distance since in general it is not symmetric and does not satisfy the triangle inequality (Beck and Teboulle [2003]).

The Bregman divergence measures the difference between the value of Φ at x and the first-order Taylor expansion of Φ around y evaluated at point x. From a geometrical point of view, the Bregman divergence quantifies the gap between the convex function Φ and its tangent hyperplane at y. This gap grows as the points x and y move further apart in a manner that reflects the curvature of the function Φ. If the function Φ has higher curvature, the Bregman divergence grows more rapidly as x moves away from y. In flatter regions of the function, the divergence will be smaller. Therefore, the Bregman divergence provides a geometrically meaningful way to capture distances that are influenced by the shape of the distance-generating function.

Simplex setup (Bubeck [2015]). Over the domain of n dimensional simplex $\Delta_n = \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$, the negative entropy function $\Phi(x) = \sum_{i=1}^n x_i \log x_i$, gives a choice of distance-generating function on $\mathbb{R}^n_+$. In this case, the induced Bregman divergence is obtained as follows:
By (1.2.2), and having that $\nabla\Phi(y) = (1 + \log y_1, 1 + \log y_2, \ldots, 1 + \log y_n)$, we have:

$$D_\Phi(x, y) = \sum_{i=1}^n [x_i \log x_i - y_i \log y_i] - \sum_{i=1}^n [(1 + \log y_i)(x_i - y_i)]$$

$$= \sum_{i=1}^n [x_i \log x_i - y_i \log y_i - (1 + \log y_i)(x_i - y_i)]$$

$$= \sum_{i=1}^{n} [x_i \log x_i - x_i \log y_i] + \sum_{i=1}^{n} (x_i - y_i) = \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i}.$$

This is exactly the Kullback-Leibler divergence:

$$D_\Phi(x, y) = \mathrm{KL}(x \parallel y) = \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i}. \tag{1.2.3}$$

Hence, the Bregman divergence induced by the negative entropy function is the Kullback-Leibler divergence.

Generalized strong convexity (Lan [2022]). Using the notion of Bregman divergence, the generalized strong convexity of a function is defined as follows. A function $f : \Omega \to \mathbb{R}$ is said to be strongly convex with modulus $\mu > 0$ with respect to the Bregman divergence D_Φ if, for all $x, y \in \Omega$, the following inequality holds:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \mu D_\Phi(x, y). \tag{1.2.4}$$

Example 1.1(1-Strongly convex case). Considering Φ the negative entropy function as distance-generating function, and $h(p) : \Delta_n \to \mathbb{R}$ as follows:

$$h(p) = D_\Phi(p, q) = KL(p \parallel q)$$

In which $q \in \Delta_n$, and the support of p is contained in the support set of q, $\mathrm{supp}(p) \subseteq \mathrm{supp}(q)$, meaning that for non-zero components of p, the corresponding component of q is non-zero. For any $p, p' \in \Delta_n$ s.t. $\mathrm{supp}(p) \subseteq \mathrm{supp}(p')$, we have:

$$h(p) - h(p') - \langle \nabla h(p'), p - p' \rangle$$

$$= \sum_{i=1}^{n} p_i \log(\frac{p_i}{q_i}) - \sum_{i=1}^{n} p'_i \log(\frac{p'_i}{q_i}) - \sum_{i=1}^{n} (\log \frac{p'_i}{q_i} + 1)(p_i - p'_i)$$

$$= \sum_{i=1}^{n} p_i \log(\frac{p_i}{q_i}) - \sum_{i=1}^{n} p_i \log(\frac{p'_i}{q_i})$$

$$= \sum_{i=1}^{n} p_i \log(p_i) - p_i \log(p'_i) = KL(p \parallel p') \tag{1.2.5}$$

This implies that if the mentioned conditions are satisfied, the 1-Strong convexity property holds for h.

Lemma 1.1 (Three-Point lemma; Chen and Teboulle [1993], Lemma 3.1.) Let ϕ be a distance generation function and a, b, c in the relative interior of $\mathcal{D}$ and $c \in \mathcal{D}$, we have

$$D_\phi(c, a) + D_\phi(a, b) - D_\phi(c, b) = \langle \nabla\phi(b) - \nabla\phi(a), c - a \rangle \tag{1.2.6}$$

Proof. Using the definition of the Bregman divergence D_ϕ, we have

$$\langle \nabla\phi(a), c - a \rangle = \phi(c) - \phi(a) - D_\phi(c, a),$$

$$\langle \nabla\phi(b), a - b \rangle = \phi(a) - \phi(b) - D_\phi(a, b),$$

$$\langle \nabla\phi(b), c - b \rangle = \phi(c) - \phi(b) - D_\phi(c, b).$$

Subtracting the first and second equations from the last one yields the result.

1.3 The mirror descent algorithm

In 1983, A. Nemirovski and D. B. Yudin [1983] introduced the *mirror descent* algorithm as a generalized optimization method for convex problems. This approach extends the classical gradient descent by leveraging a non-Euclidean geometry, defined via a Bregman divergence, to guide the optimization trajectory. Unlike traditional methods, which rely on Euclidean distance, mirror descent employs a distance-generating function (called mirror map in this context) to adaptively project iterates onto a feasible set. The core idea is to perform gradient updates in the dual space defined by the distance-generating function and then map the result back to the primal space, ensuring that the iterates remain within the constrained domain.

Let $\mathcal{D} \subset \mathbb{R}^n$ be a convex open set such that $\mathcal{X}$ is contained within its closure, i.e., $\mathcal{X} \subseteq \overline{\mathcal{D}}$, and $\mathcal{X} \cap \mathcal{D} \neq \emptyset$. We refer to $\Phi : \mathcal{D} \to \mathbb{R}$ as a mirror map if it satisfies the following properties (Bubeck [2015]):

1. Φ is strictly convex and differentiable.
2. The gradient of Φ covers all possible values, that is, $\nabla\Phi(\mathcal{D}) = \mathbb{R}^n$.
3. The gradient of Φ diverges on the boundary of $\mathcal{D}$, meaning:

$$\lim_{x \to \partial\mathcal{D}} \|\nabla\Phi(x)\| = +\infty.$$

Formally, mirror descent iterates according to the rule: given a convex objective function f and a Bregman divergence D_Φ induced by a strongly convex function Φ, called mirror map in this context, the update step is defined as

$$x_{k+1} = \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \left\{ \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{\eta_k} D_\Phi(x, x_k) \right\}, \tag{1.3.1}$$

where η_k is the step size. The method is trying to minimize the local linearization of the function while not moving too far away from the previous point, with distances measured via the Bregman divergence of the mirror map. This formulation allows mirror descent to efficiently handle high-dimensional problems and achieve faster convergence in settings where the geometry of the feasible set or the objective function favors non-Euclidean updates(Beck and Teboulle [2003]).

By selecting an appropriate mirror map, the mirror descent algorithm can be alternatively performed in the following way(Bubeck [2015]):

Algorithm 1 Mirror Descent

Require: Objective function f, initial point $x_1 \in \mathcal{X} \cap \mathcal{D}$, stepsize η_t

1: **for** $t = 1, 2, \ldots$ **do**
2: acquire $g_t \in \partial f(x_t)$
3: $\hat{x}_t \leftarrow \nabla\Phi(x_t)$ ▷ Map primal point to dual space
4: $\hat{x}_{t+1} \leftarrow \hat{x}_t - \eta g_t$ ▷ Perform gradient step in the dual space
5: $y_{t+1} \leftarrow (\nabla\Phi)^{-1}(\hat{x}_{t+1})$ ▷ Map updated dual point back to the primal space
6: $x_{t+1} \leftarrow \Pi_{\mathcal{X} \cap \mathcal{D}}(y_{t+1})$ ▷ Project new point onto feasible region
7: **end for**

The updated point x_{t+1} in the last step of the algorithm is obtained via the projection into the feasible set using the Bregman divergence, which can be written as:

$$\begin{aligned}
x_{t+1} &= \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} D_\Phi(x, y_{t+1}) \\
&= \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \{\Phi(x) - \Phi(y_{t+1}) - \langle \nabla\Phi(y_{t+1}), x - y_{t+1} \rangle\} \\
&= \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \{\Phi(x) - \langle \nabla\Phi(y_{t+1}), x \rangle\} \\
&= \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \{\eta \langle g_t, x \rangle + \Phi(x) - \Phi(x_t) - \langle \nabla\Phi(x_t), x - x_t \rangle\} \\
&= \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \{D_\Phi(x, x_t) + \eta \langle g_t, x \rangle\},
\end{aligned} \tag{1.3.2}$$

This demonstrates that both formulations are equivalent. As we can observe, in Mirror Descent, the gradient of the mirror map Φ is employed to map points from the "primal" space to the "dual" space. This transformation enables updates that respect the structure of the problem, ensuring that the optimization steps are tailored to the problem's constraints and geometry rather than relying on simple Euclidean projections (Bubeck [2015]).

Example 1.2. (The entropic descent algorithm, Beck and Teboulle [2003]). Consider the negative entropy setting where the mirror map is $\Phi(x) = \sum_{i=1}^{n} x_i \log x_i$, the initial point is the maximum entropy distribution over a set of size n, i.e., $x_1 = (1/n, \ldots, 1/n)$. Following the mirror descent algorithm, we have

$$\nabla\Phi(x_t)_j = 1 + \log(x_t)_j, \quad \nabla\Phi^{-1}(\hat{x}_{t+1})_j = \exp((\hat{x}_{t+1})_j - 1)$$

thus:

$$(y_{t+1})_j = \exp((\hat{x}_{t+1})_j - 1) = \exp(\log((x_t)_j) - \eta(g_t)_j - 1) = (x_t)_j \exp(-\eta(g_t)_j). \quad (1.3.3)$$

Now, to perform the projection with respect to Bregman divergence, in this case KL-divergence, for any y we have:

$$\arg\min_{x \in \mathcal{X} \cap \mathcal{D}} D_\Phi(x, y) = \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i}$$

Knowing that x is a probability distribution, Jensen's inequality for concave functions yields

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} x_i \frac{y_i}{x_i}$$

$$\Rightarrow \log(\sum_{i=1}^{n} y_i) = \log(\sum_{i=1}^{n} x_i \frac{y_i}{x_i}) \geq \sum_{i=1}^{n} x_i log(\frac{y_i}{x_i})$$

$$\Rightarrow \log(\frac{1}{\|y_i\|_1}) \leq \sum_{i=1}^{n} x_i log(\frac{x_i}{y_i})$$

and equality holds if and only if all x_i/y_i are equal, i.e., $x_i = y_i/\|y\|_1$. This implies that

$$\Pi_{\mathcal{X} \cap \mathcal{D}}(y) = \arg\min_{x \in \mathcal{X} \cap \mathcal{D}} D_\Phi(x, y) = y/\|y\|_1$$

Applying the projection to (1.3.3), we obtain

$$(x_{t+1})_j = \frac{(x_t)_j \exp(-\eta(g_t)_j)}{\sum_{j=1}^{n} (x_t)_j \exp(-\eta(g_t)_j)} \quad (1.3.4)$$

Following the obtained update rule in (1.3.4) for each coordinates of x_{t+1}, one can iteratively performe mirror gradient descent steps with respect to negative entropy mirror map.

1.4 Variational inequalities

In this section, we provide a brief overview of variational inequalities, which are extensively discussed in Lan [2021], section 3.8. Variational inequalities are fundamental in the study of optimization and equilibrium problems and will be employed to articulate our optimization problem in (2.3.9). By formulating our problem within the variational inequality framework, we can capture the necessary and sufficient conditions for optimality, enabling a deeper analytical understanding of the solution space.

Given a non-empty closed convex subset $\mathcal{D}$ of the Euclidean $n - dimensional$ space $\mathbb{R}$ and a continuous mapping $f : \mathcal{D} \to \mathbb{R}^n$, the variational inequality, denoted VI $(\mathcal{D}, f)$, is to find a vector $x^\star \in \mathcal{D}$ such that

$$\langle f(x^\star), x - x^\star \rangle \geq 0, \quad \forall x \in \mathcal{D} \tag{1.4.1}$$

Such a point $x^\star$ is called a strong solution of VI$(\mathcal{D}, f)$. In particular, if f is given by the gradient of F, then (1.4.1) is exactly the first-order necessary optimality condition of $\min_{x\in\mathcal{D}} F(x)$. One important class of VI problems is called monotone VI, for which the operator $f(\cdot)$ is monotone, i.e.,

$$\langle f(x) - f(y), x - y \rangle \geq 0, \quad \forall x, y \in \mathcal{D}$$

A related notion is a weak solution of VI $(\mathcal{D}, f)$, i.e., a point $x^\star \in X$ such that

$$\langle f(x), x - x^\star \rangle \geq \forall x \in \mathcal{D}$$

Note that if $f(\cdot)$ is monotone and continuous, a weak solution of $(\mathcal{D}, f)$ must be a strong solution and vice versa (Lan [2021]).

A class of generalized monotone VI(GMVI) problems, which satisfies for any $x^\star$ is the following (Lan [2021], (3.8.3))

$$\langle f(x), x - x^\star \rangle \geq 0, \quad \forall y \in \mathcal{D} \tag{1.4.2}$$

Clearly this condition is satisfied if f is monotone, moreover, this assumption holds if f is pseudo-monotone as well, i.e.,

$$\langle f(y), x - y \rangle \geq 0 \Rightarrow \langle f(x), x - y \rangle \geq 0 \tag{1.4.3}$$

Lemma 1.2 (Lan [2021], Lemma 3.10.). Given a strict convex function f over $\mathcal{D}$, point $x \in \mathcal{X}$ is a strong solution of $\mathrm{VI}(\mathcal{D}, f)$, for any $\eta > 0$, if and only if

$$x = \arg\min_{y \in \mathcal{D}} \{\langle \eta f(x), y \rangle + D_\phi(y, x)\} \tag{1.4.4}$$

Where ϕ is a distance generating function for Bregman divergence $D_\phi(\cdot, \cdot)$.

Proof. Lan [2021], Lemma 3.10. □

Chapter 2

Reinforcement learning setup

2.1 Markov decision processes

Markov decision processes (MDPs) are defined as controlled stochastic processes that satisfy the Markov property, where cost values are assigned to state transitions. Alternatively, instead of costs, rewards can be assigned to these transitions, depending on the nature of the problem. MDPs are a classical formalization of sequential decision making, where actions influence not just immediate cost, but also subsequent situations, or states, and through those future costs. Thus MDPs involve delayed cost and the need to trade off immediate and delayed cost (Sutton and Barto [2020]). In an MDP, the process specifies the probability of transitioning to a state s' incurring a cost c, when action a is taken in state s. The resulting sequence of state transitions can be evaluated based on the observed costs. Solving an MDP involves directing the agent's behavior to achieve optimal performance. This typically entails minimizing the total cost over time or, when working with a reward function, maximizing the cumulative reward. Due to the stochastic nature of action outcomes, the optimal control strategy cannot generally be represented as a simple sequence of actions. Instead, MDP solutions are expressed as policies or universal plans, which determine the appropriate action to take in each state during the process. Given the uncertainty in action outcomes, following a given policy may lead to different paths of states and actions (see Puterman [1994], Chapter 2; Sutton and Barto [2020], Chapter 3).

The definition of a Markov Decision Process can vary slightly across different frameworks. Here, following Lan [2021] and Xiao [2022] we describe and MDP by the 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, c, \gamma)$, where :

- $\mathcal{S}$ is the set of states that represents all possible situations the process can be in.

- $\mathcal{A}$ is the set of actions available to the decision-maker, which influence the transitions between states.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel, where $\mathcal{P}(s' \mid s, a)$ gives the probability of moving from state s to state s' after taking action a.
- $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the cost(reward) function, where $c(s, a)$ gives the expected cost(reward) received after taking action a in state s.
- γ is the discount factor, a scalar in $[0, 1]$ used to weight future cumulative cost

Unless otherwise specified, we assume the sets $\mathcal{S}$ and $\mathcal{A}$ are finite and $\gamma \in (0, 1)$.

In an MDP, the **Markov property** means that the transition probabilities from one state to another depend only on the current state and the action chosen and not on any of the prior states or actions that led to the current state. Mathematically, the Markov property can be expressed as:

$$\mathcal{P}(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = \mathcal{P}(s_{t+1}|s_t, a_t)$$

therefore, the probability distribution of the next state s_{t+1} is independent of the history of previous states and actions (Sutton and Barto [2020]).

A **policy** π is a function $\pi(\cdot|\cdot) : \mathcal{A} \times \mathcal{S} \to [0, 1]$, that determines the probability of selecting a particular action at a given state. Given a state s, action a is chosen by sampling from the set of actions $\mathcal{A}$ according to the distribution $\pi(\cdot|s)$.
We are also interested in the sequence of states and actions that arise when following a given policy, starting from an initial state s_0, which is called trajectory.

Trajectory. Adopting the notation from Lan [2022], $\zeta^\pi(s_0)$ a trajectory is defined as the following sequence:

$$\zeta^\pi(s_0) = \{(s_0, a_0, c_1), (s_1, a_1, c_2), (s_2, a_2, c_3), \ldots\} \tag{2.1.1}$$

where:

- $s_t \in \mathcal{S}$ is the state of the system at time step t,
- $a_t \in \mathcal{A}$ is the action taken at time step t,
- $c_{t+1} = c(s_t, a_t)$ is the cost after taking action a_t in state s_t,

$\zeta^\pi(s_0)$ starts from an initial state s_0 and evolves based on the dynamics of the MDP, where action a_t at state s_t is sampled according to a policy π, and new state s_{t+1} is sampled according to the transition function $\mathcal{P}(\cdot|s_t, a_t)$. As will be demonstrated in subsequent

sections, these trajectories enable us to estimate and develop effective policies even when the transition kernel and dynamics of the Markov Decision Process (MDP) are unknown.

2.2 Fundamental concepts

In this section, we provide a concise overview of essential concepts and elements in reinforcement learning (RL), primarily drawing from Sutton and Barto [2020] and Lan [2022]. However, for consistency and clarity throughout this work, we adopt the notation and perspectives established in our primary reference Lan [2022]. This approach ensures a unified framework as we analyze and develop further theorems and concepts. By synthesizing these foundational ideas, we aim to present a coherent and structured exposition that aligns with the objectives of this thesis.

Nearly all reinforcement learning algorithms focus on estimating value functions, which are mathematical functions that evaluate how advantageous it is for an agent to be in a particular state. More precisely, starting from a state s, the value function calculates the expected discounted return over time, assuming the agent follows a specific policy. The value function at state s following policy π is defined as

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\right] \tag{2.2.1}$$

Building on this concept, if we assume that the action a is taken at state s and following the policy π, we can define the action-value function (Q-function) as follows:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\right] \tag{2.2.2}$$

It's straightforward to verify that:

$$Q^{\pi}(s, a) = c(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s, a) V^{\pi}(s') \tag{2.2.3}$$

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^{\pi}(s, a) = \langle Q^{\pi}(s, \cdot), \pi(\cdot|s) \rangle \tag{2.2.4}$$

The value functions V_π and Q_π can be estimated based on experience. For instance, if an agent follows policy π in a MDP, and maintains an average of actual returns that have followed that state, over time, as the state is encountered more frequently, the average will converge to the true value of the state, $V_\pi(s)$, as the number of visits approaches

infinity. Similarly, if separate averages are maintained for each action taken in a state, those averages will converge to the action values, $Q^\pi(s,a)$.

This class of estimation techniques is referred to as *Monte Carlo methods* (Kalos and Whitlock [2007]), as they rely on averaging over many random samples of actual returns. For example, when the cost function $c(\cdot,\cdot)$ is known, and given a policy π_k, the Q-function for a state-action pair (s,a) can be estimated in a multi-trajectory scenario. Specifically, if M_k independent trajectories of length T_k are available, this allows for estimating the Q-function based on the observed trajectories. For $i = 1, \ldots, M_k$,

$$\zeta_k^i \equiv \zeta_k^i(s,a,c) := \{(s_0^i = s, a_0^i = a, c_1^i = c(s_0^i, a_0^i)); (s_1^i, a_1^i, c_2^i), \ldots, (s_{T_k-1}^i, a_{T_k-1}^i, c_{T_k}^i)\} \tag{2.2.5}$$

Let $\xi_k := \{\zeta_k^i(s,a,c), i = 1, \ldots, M_k, s \in S, a \in A\}$ denote all these random variables. We can estimate Q^{π_k} by

$$Q^{\pi_k,\xi_k}(s,a) = \frac{1}{M_k} \sum_{i=1}^{M_k} \sum_{t=0}^{T_k-1} \gamma^t \left[c(s_t^i, a_t^i)\right]. \tag{2.2.6}$$

(Lan [2022], section 5.1.)

Bellman equation for $V^\pi(s)$. A fundamental property of value functions, widely used in reinforcement learning and dynamic programming, is their ability to satisfy certain recursive relationships. For any policy π and any state s, the following consistency condition holds between the value of state s and the value of its possible successor states(Sutton and Barto [2020], Chapter 4):

$$\begin{aligned}
V^\pi(s) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_t \sim \pi(\cdot \mid s_t), s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)\right] \\
&= \mathbb{E}\left[c(s_0, a_0) + \gamma \sum_{t=1}^{\infty} \gamma^{t-1} c(s_t, a_t) \mid s_0 = s, a_t \sim \pi(\cdot \mid s_t), s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)\right] \\
&= \sum_a \pi(a|s) \sum_{s'} \mathcal{P}(s' \mid s, a) \left[c(s,a) + \gamma \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} \mathcal{P}(s' \mid s, a) \left[c(s,a) + \gamma V^\pi(s')\right]
\end{aligned} \tag{2.2.7}$$

Optimal value function $V^\star$ and optimal policy $\pi^\star$. In reinforcement learning, the goal is often to find a policy that minimizes the cumulative cost over the long run. For finite Markov Decision Processes (MDPs), we can define an optimal policy in the following way. Value functions establish a partial ordering over policies; a policy π is said to be better than or equal to another policy π' if the expected cumulative cost under

π is less than or equal to that under π' for all states. In other words, $V^{\pi}(s) \leq V^{\pi'}(s)$ for all $s \in \mathcal{S}$ (Sutton and Barto [2020]). In reinforcement learning, particularly within the framework of Markov Decision Processes, one fundamental assurance is the existence of a policy that minimizes expected cumulative cost; such a policy is referred to as the optimal policy (Puterman [1994], Section 4.4). Although there may be multiple optimal policies, we denote all such policies by $\pi^{\star}$. These optimal policies share the same optimal state-value function, denoted by $V^{\star}(s)$, and defined as

$$V^{\star}(s) = \min_{\pi} V^{\pi}(s) \qquad \forall s \in \mathcal{S} \tag{2.2.8}$$

$$\pi^{\star}(\cdot|s) \in \arg\min_{\pi} V^{\pi}(s), \quad \forall s \in \mathcal{S}$$

Here, $V^{\star}(s)$ depends on the distribution over states, which is determined by the policy π and the environment's transition probabilities, and unlike the mirror descent optimization problems, the objective function here is not convex in π (Tomar et al. [2020]). The distribution affects how frequently different states are visited, which, in turn, influences the expected cumulative cost. Policies that visit costly states more frequently will generally result in higher value function estimates. Optimal policies also share the same optimal action-value function , denoted $Q^{\star}(s,a)$, which is defined as

$$Q^{\star}(s,a) = \min_{\pi} Q^{\pi}(s,a), \quad \forall s \in \mathcal{S}, \text{ and } a \in \mathcal{A}$$

For a state-action pair (s,a), this function represents the expected cumulative cost for taking action a in state s and then following an optimal policy (Sutton and Barto [2020]). The expected cost depends on how the state-action pair affects the likelihood of entering future states and on the distribution of costs in those states. Thus, we can express $Q^{\star}(s,a)$ in terms of $V^{\star}(s)$ as follows:

$$Q^{\star}(s,a) = \mathbb{E}\left[c(s,a) + \gamma V^{\star}(s') \mid s_0 = s, a_0 = a, s' \sim \mathcal{P}(\cdot \mid s,a)\right].$$

The optimal value function obtained from (2.2.8) also needs to satisfy the Bellman equation (2.2.7). The consistency condition (2.2.7) for $V^{\star}$ can be expressed in a specific form that does not depend on any particular policy. This is known as the Bellman optimality equation , and it is given as follows (Sutton and Barto [2020], Chapter 3):

$$V^{\star}(s) = \min_{a \in \mathcal{A}} \left[c(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^{\star}(s') \right] \tag{2.2.9}$$

Intuitively, the Bellman optimality equation captures the idea that the value of a state,

under an optimal policy, must be equal to the expected return from taking the best possible action in that state. This concept is formalized as follows

$$
\begin{aligned}
V^\star(s) = \min_{a\in\mathcal{A}} Q^\star(s,a) &= \min_{a\in\mathcal{A}} \mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t) \mid s_0=s, a_0=a\right] \\
&= \min_{a\in\mathcal{A}} \mathbb{E}_{\pi^\star}\left[c(s,a) + \gamma\sum_{t=0}^{\infty}\gamma^t c(s_{t+1},a_{t+1}) \mid s_0=s, a_0=a\right] \\
&= \min_{a\in\mathcal{A}} \mathbb{E}_{\pi^\star}\left[c(s,a) + \gamma V^\star(s') \mid s_0=s, a_0=a, s'\sim\mathcal{P}(\cdot\mid s,a)\right] \\
&= \min_{a\in\mathcal{A}} \sum_{s'}\mathcal{P}(s'\mid s,a)\left[c(s,a)+\gamma V^\star(s')\right]. \qquad (2.2.10)
\end{aligned}
$$

Tabular and approximate frameworks. In reinforcement learning, the tabular framework applies to finite Markov Decision Processes (MDPs) where both state and action spaces are sufficiently small to allow for explicit enumeration. In this setting, value functions $V^\pi(s)$ and $Q^\pi(s,a)$, as well as policies $\pi(a \mid s)$, could be represented as tables with entries corresponding to each state or state-action pair. This explicit representation facilitates exact computation and updates. The tabular approach is particularly advantageous in scenarios with manageable state-action spaces, enabling straightforward implementation and theoretical analysis. However, its scalability is limited, making it unsuitable for environments with large or continuous state and action spaces due to the exponential growth in memory and computational requirements (Sutton and Barto [2020] Part I; Wiering and van Otterlo [2012], Section 4.2). Unless otherwise specified, the analysis presented in this work is conducted within a tabular framework.

Learning a model within a continuous domain necessitates making predictions about both the continuous next state and the associated costs based on continuous-state inputs. Unlike discrete environments, where a tabular model can be feasibly constructed for a finite set of states, continuous domains require alternative approaches due to the potentially infinite number of states. Consequently, function approximation becomes essential, as many real-valued states may never be encountered during training. In this case, instead of maintaining explicit tables, value functions and policies are approximated using parameterized function approximators, such as linear models $V(s;\theta_v) = \phi(s)^\top\theta_v$ or neural networks $Q(s,a;\theta_q) = f(s,a;\theta_q)$, which facilitate the development of continuous models by generalizing from observed data. Planning in a continuous state space introduces additional complexities, as an agent must determine optimal actions across an uncountably infinite number of states. This challenge can be addressed through function approximation applied to the policy, enabling the agent to generalize optimal actions across similar states (Sutton et al. [1999], Part II). Alternatively, discretizing the state space for plan-

ning purposes is another viable approach, even if the model itself operates in a continuous domain. This discretization allows for the application of traditional planning algorithms by approximating the continuous space with a finite set of representative states. For example, Gordon [1995] introduced fitted value iteration that adapts value iteration to continuous state spaces. It iterates, updating the values of a finite set of states sampled from the infinite state space and then fitting a function approximator to their values. If the function approximator fits some contraction criteria, then fitted value iteration is proven to converge.
In addition, many of the model-free approaches for continuous state spaces, such as policy gradient methods (Sutton et al. [1999]) and Q-learning with function approximation(Mnih et al. [2015], Melo and Ribeiro [2007]), or deep recurrent Q-learning (Zangirolami and Borrotti [2024]) can be effectively employed to develop and optimize policies directly within the model. These approaches leverage the flexibility of function approximators to handle the intricacies of continuous domains, facilitating the discovery of optimal policies without the need for an explicit model of the environment (Wiering and van Otterlo [2012]).

Discounted state-visitation distribution, (Lan [2022]). Consider starting from an initial state $s_0 \in S$, where the discounted state-visitation distribution under policy π is represented by the vector $d_s(\pi) \in \Delta_{|\mathcal{S}|}$. The components of this vector are defined as follows:

$$d^{\pi}_{s_0}(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathrm{Pr}^{\pi}(s_t = s \mid s_0), \quad \forall s' \in S. \tag{2.2.11}$$

where $\mathrm{Pr}^{\pi}(s_t = s' \mid s_0)$ denotes the state visitation probability of $s_t = s'$ after we follow the policy π starting at state s_0 The normrmalization coefficient $1-\gamma$ guarantees that $\sum_{s' \in S} d^{\pi}_{s_0}(s') = 1$. Let $\mathcal{P}^{\pi}$ denote the transition probability matrix associated with policy π, i.e., $\mathcal{P}^{\pi}(i,j) := \sum_{a \in \mathcal{A}} \pi(a|i)\mathcal{P}(j|i,a)$, and e_i being the i-th unit vector, then $Pr^{\pi}(s_t = s|s_0) = e_{s_0}^{\top}(\mathcal{P}^{\pi})^t e_s$, and

$$d^{\pi}_{s_0}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t e_{s_0}^{\top}(\mathcal{P}^{\pi})^t e_s. \tag{2.2.12}$$

Given an initial state distribution $\rho \in \Delta_{|\mathcal{S}|}$, we define $d_{\rho}(\pi) \in \Delta_{|\mathcal{S}|}$ whose components are given by (Xiao [2022]):

$$d^{\pi}_{\rho}(s) = \mathbb{E}_{s_0 \sim \rho} d^{\pi}_{s_0}(s) = \sum_{s_0 \in S} \rho(s_0) d^{\pi}_{s_0}(s) \tag{2.2.13}$$

A few important properties derived from the definitions are

$$d_s^\pi(s) \geq 1 - \gamma \quad \text{and} \quad d_\rho^\pi(s) \geq (1-\gamma)\rho(s), \quad \forall s \in S. \tag{2.2.14}$$

Stationary state distribution, (Lan [2022]). Also denoted by steady-state distribution, refers to the long-term probability distribution over states that a Markov Decision Process (MDP) will settle into when it is run under a particular policy. If the MDP runs for an infinite amount of time, under certain conditions, the probability of being in each state will converge to a fixed distribution that no longer changes over time — this is the stationary distribution.
Formally, the stationary state distribution, or steady-state distribution of the MDP under policy π, denoted by ν^π, is a distribution over states $s \in \mathcal{S}$ such that:

$$\nu^\pi(s') = \sum_{s \in \mathcal{S}} \nu^\pi(s) \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}(s'|s,a) \tag{2.2.15}$$

This equation states that the probability of being in state s' under the stationary distribution is equal to the probability of transitioning into s' from all other states, weighted by the stationary distribution over those other states. This is the special distribution under which, if you select actions according to π, you remain in the same distribution (Sutton and Barto [2020], 13.16).

Exploration-exploitation trade-off One of the fundamental challenges unique to reinforcement learning (RL), is the exploration-exploitation trade-off. Unlike other forms of learning, such as supervised or unsupervised learning, RL agents must balance the need to minimize cumulative costs based on existing knowledge (exploitation) with the necessity to gather new information about the environment (exploration). To effectively minimize costs, the agent must exploit actions that it has learned are cost-effective. However, to discover these actions in the first place, it must explore less-known actions, which may initially lead to higher costs. Exclusively exploiting known actions may prevent the agent from discovering better strategies, while only exploring can lead to unnecessary costs without improving performance (Sutton and Barto [2020]).

Support of a policy $\pi(\cdot \mid s)$ at state s is defined as

$$\text{supp}(\pi(\cdot \mid s)) = \{a \in \mathcal{A} \mid \pi(a \mid s) > 0\},$$

the support indicates which actions have a non-zero probability of being chosen under

policy π in state s. Consider the Bellman optimality condition for cost minimization:

$$Q^\star(s,a) = c(s,a) + \gamma \sum_{s'} P(s'|s,a)V^\star(s'),$$

$$V^\star(s) = \min_a Q^\star(s,a)$$

If $\pi(a|s) = 0$ for some action a that achieves $V^\star(s)$ in (2.2.10), then:

$$\pi \neq \pi^\star$$

and the policy π cannot satisfy the Bellman optimality condition. Then it's crucial to maintain a sufficient exploration to keep all possible optimal actions accessible. There are several approaches to promoting exploration in reinforcement learning, such as ϵ-greedy, softmax, Max-Boltzmann methods, and regularization techniques (Zangirolami and Borrotti [2024]). A popular strategy of this kind is to enforce entropy regularization (Williams and Peng [1991]), formally, incorporating a negative entropy term $H(p) = \sum_{i=0}^{n} p_i \log(p_i)$ into the optimization objective can serve as a penalty. This term penalizes low-entropy distributions, encouraging the algorithm to favor higher entropy distributions, which are closer to the maximum entropy state, the uniform distribution. In the case of the negative entropy function employed in the cost minimization task, the uniform distribution would indeed correspond to the minimum value of the penalty. The maximum entropy formulation provides a substantial improvement in exploration and robustness: as discussed by Ziebart [2010], maximum entropy policies are robust in the face of model and estimation errors, and as demonstrated by (Haarnoja et al. [2017]), they improve exploration by acquiring diverse behaviors (Haarnoja et al. [2018]).

2.3 Reinforcement learning for regularized MDPs

Regularization has been suggested to the RL literature either through the lens of optimization (Dai et al. [2018]; Agarwal et al. [2021]), or through the lens of dynamic programming (Geist et al. [2019]; Vieillard et al. [2020]). In light of importance of regularization in RL, it is of imminent value to develop a unified framework for understanding the capability and limitations of regularized policy optimization. A recent line of work (Agarwal et al. [2021]; Mei et al. [2020]; Cen et al. [2022]) has looked into specific types of regularization techniques such as entropy regularization; however, existing convergence theory remains highly inadequate when it comes to a more general family of regularizers (Zhan et al. [2023]).

Building on these foundational concepts and theoretical gaps, Lan [2022] introduces

a generalization that incorporates a class of regularizers into the definition of value functions and state-value functions, as well as the optimization process itself. In regularized reinforcement learning objectives, incorporating a more general regularizer denoted by h into the policy π allows us not only to unify various reinforcement learning scenarios but also to significantly increase the expressiveness and flexibility of the framework. The inclusion of h extends the applicability of reinforcement learning in several ways:

- By choosing an appropriate form for h, such as an indicator function, quadratic penalty, entropy function, or barrier function, the regularizer can effectively encode constraints that the optimal policy should satisfy. For example, considering the regularizer to be a negative entropy function in case of minimizing cost, as discussed before, encourages the exploration.

- h can be used to capture complex dependencies or correlations between actions across different states, which would otherwise be difficult to model (Lan [2022]).

- Furthermore, the regularizer can represent risk preferences or utility functions associated with the policy π, allowing the optimization process to account for uncertainty or prioritize specific outcomes beyond immediate cost minimization. for example, in a variety of application scenarios such as industrial robot arms and self-driving vehicles, the agents are required to operate safely both to themselves and the surroundings (Amodei et al. [2016]), certain actions might be strictly forbidden in some states. One way to incorporate such prescribed operational constraints is through adding a regularizer (e.g., a properly chosen log barrier or indicator function tailored to the constraints) to explicitly account for the constraints (Zhan et al. [2023]).

Thus, the presence of regularizer enables reinforcement learning to handle constrained optimization problems and to model more complex decision-making scenarios, enhancing the robustness and versatility of policy optimization in finite and general state spaces. In (Lan [2022], Ju and Lan [2022] Li et al. [2022]), Guanghui Lan and colleagues demonstrated that by using a strongly convex function with respect to a Bregman divergence as a regularizer, a linear convergence rate can be achieved in optimization. From this point onward, we focus on a general class of reinforcement learning problems that involve convex or strongly convex regularizers within their cost functions. The subsequent material is primarily based on the framework and methodologies outlined in Lan [2022].

Consider a function over the policy space $h : \Delta_{|\mathcal{A}|} \times \mathcal{S} \to \mathbb{R}$ which is closed convex w.r.t its first argument, i.e., $\pi(\cdot|s)$. By the definition of stronge convexity and its corresponding

equation (1.2.4), $\forall \pi, \pi'$ at fixed state s, there exists some $\mu \geq 0$ such that

$$h^{\pi}(s) - [h^{\pi'}(s) + \langle (h')^{\pi'}(s, \cdot), \pi(\cdot|s) - \pi'(\cdot|s) \rangle] \geq \mu D^{\pi}_{\pi'}(s) \tag{2.3.1}$$

Where $\langle \cdot, \cdot \rangle$ denotes the inner product over the action space $\mathcal{A}$, $(h')^{\pi'}(s, \cdot)$ denotes a subgradient of $h(s)$ at π' and $D^{\pi}_{\pi'}(s)$ is the Bregman divergence or Kullback-Leibler divergence in this context between $\pi(\cdot|s)$ and $\pi'(\cdot|s)$. Based on this regularizer, we define regularized Value function, Q-function and advantage function as follows:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t [c(s_t, a_t) + h^{\pi}(s_t)] \mid s_0 = s, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\right] \tag{2.3.2}$$

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t [c(s_t, a_t) + h^{\pi}(s_t)] \mid s_0 = s, a_0 = a, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\right] \tag{2.3.3}$$

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s) \tag{2.3.4}$$

Example 2.1(Negative Entropy Regularizer): As mentioned in the previous section, the negative entropy function $H(p) = \sum_{i=0}^{n} p_i \log(p_i)$ can serve as a regularizer. According to the definition of strong convexity in (1.2.4) and the arguments leading to (1.2.3), the negative entropy function is 1-strongly convex with respect to the KL divergence. Moreover, for the uniform distribution $\pi_0 \in \Delta_n \subset \mathbb{R}^n$, and for any $p \in \Delta_n$, we have:

$$\begin{aligned} \mathrm{KL}(p \,||\, \pi_0) = \sum_{i=0}^{n} p_i \log(\frac{p_i}{1/n}) &= \sum_{i=0}^{n} p_i \log(p_i) + p_i \log(n) \\ &= \log(n) + H(p) \end{aligned}$$

As indicated in Example 1.1, the Kullback-Leibler divergence $\mathrm{KL}(p \,||\, \pi_0)$ is strongly convex with modulus 1. Also, It is straightforward to demonstrate that in $\tau\,\mathrm{KL}(p \,||\, \pi_0)$, the coefficient τ is the strong convexity modulus for this term. Therefore, in optimization objectives, we can use $\tau \mathrm{KL}(p \,||\, \pi_0)$ or equivalently $\tau H(p)$ as weighted negative entropy regularizer with modulus 1, since $\log(n)$ is constant and independent of variable p. This fact appears in the analysis of APMD in Chapter 4, and also in the numerical experiment we have conducted, detailed in Chapter 6.

As stated in (2.2.8), the main objective in RL is to find a policy $\pi^{\star}(\cdot|\cdot) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that

$$V^{\pi^{\star}}(s) \leq V^{\pi}(s), \forall \pi(\cdot|s) \in \Delta_{|\mathcal{A}|}, \forall s \in \mathcal{S} \tag{2.3.5}$$

As already mentioned, the existence of a unique optimal policy $\pi^{\star}$ that satisfies (2.2.8)

is guaranteed (Puterman [1994]). Hence, we can formulate (2.2.8) as a single objective optimization problem by taking the expectation of V^π with respect to an arbitrarily chosen distribution ρ with $\rho_s > 0$ over S (Lan [2022], Tomar et al. [2020]). One reasonable choice of ρ would be stationary state distribution induced by optimal policy $\pi^\star$, denoted by $\nu^\star$ as follows

$$\min_{\pi} \{f(\pi) := \mathbb{E}_{s\sim\nu^\star}[V^\pi(s)]\}, \quad \text{s.t. } \pi(\cdot \mid s) \in \Delta_{|\mathcal{A}|}, \forall s \in \mathcal{S} \tag{2.3.6}$$

It has been observed recently (e.g., Liu et al. [2019], Li et al. [2022], Yuan et al. [2023] Appendix H.) that one can simplify the analysis of various algorithms by setting $\rho = \nu^\star$ (Lan [2022]), which avoids any distribution mismatch coefficient in the analysis (Xiao [2022]). Although the objective defined in equation (2.3.6) depends on $\nu^\star$ and, consequently, the unknown optimal policy $\pi^\star$, the algorithms developed to solve equations (2.3.5) and (2.3.6) do not require explicit knowledge of $\nu^\star$. This is demonstrated by the objective function in equation (3.1.1) of Algorithm 2, as well as in the SPMD and APMD algorithms, which will be analyzed in subsequent chapters. Under a mild conditions, in this work we have shown that change of distribition ρ would not affect the convergence rate to the optimal policy.

It is well-known that the value function $V^\pi(s)$ is highly non-convex with respect to π, due to the fact that the components of $\pi(\cdot \mid s)$ are multiplied by each other in their definitions. However, in this section, we will demonstrate that problem (2.3.6) can be reformulated as a variational inequality (VI), which adheres to certain generalized monotonicity properties, guaranteeing the existence of a solution. We begin by computing the gradient of the value function $V^\pi(s)$ in equation (2.3.2). For simplicity, h^π is assumed to be differentiable at this stage and later relax this assumption.

Lemma 2.1 (Lan [2022], Lemma 1). For any $(s_0, s, a) \in S \times S \times \mathcal{A}$, we have

$$\frac{\partial V^\pi(s_0)}{\partial \pi(a|s)} = \frac{1}{1-\gamma} d^\pi_{s_0}(s) \left[Q^\pi(s,a) + \nabla h^\pi(s,a)\right], \tag{2.3.7}$$

where $\nabla h^\pi(s,\cdot)$ denotes the gradient of $h^\pi(s)$ with respect to π.

Proof It follows from (2.2.4) that

$$\frac{\partial V^\pi(s_0)}{\partial \pi(a|s)} = \frac{\partial}{\partial \pi(a|s)} \sum_{a' \in \mathcal{A}} \pi(a'|s_0) Q^\pi(s_0, a'),$$

$$= \sum_{a' \in \mathcal{A}} \frac{\partial \pi(a'|s_0)}{\partial \pi(a|s)} Q^\pi(s_0, a') + \pi(a'|s_0) \frac{\partial Q^\pi(s_0, a')}{\partial \pi(a|s)}.$$

Also, the relation in (2.2.3) implies that

$$\frac{\partial Q^{\pi}(s_0,a)}{\partial \pi(a|s)} = \nabla h^{\pi}(s,a) + \gamma \sum_{s' \in S} P(s'|s_0,a) \frac{\partial V^{\pi}(s')}{\partial \pi(a|s)}.$$

Combining the above two relations, we obtain

$$\begin{aligned}
\frac{\partial V^{\pi}(s_0)}{\partial \pi(a|s)} &= \sum_{a' \in \mathcal{A}} \left[\frac{\partial \pi(a'|s_0)}{\partial \pi(a|s)} Q^{\pi}(s_0,a') + \pi(a'|s_0) \frac{\partial h^{\pi}(s_0)}{\partial \pi(a|s)} \right] \\
&\quad + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s_0) \sum_{s' \in S} \mathcal{P}(s'|s_0,a') \frac{\partial V^{\pi}(s')}{\partial \pi(a|s)} \\
&= \sum_{x \in S} \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}^{\pi}(s_t = x|s_0) \sum_{a' \in \mathcal{A}} \left[\frac{\partial \pi(a'|x)}{\partial \pi(a|s)} Q^{\pi}(x,a') + \pi(a'|x) \frac{\partial h^{\pi}(x)}{\partial \pi(a|s)} \right] \\
&= \frac{1}{1-\gamma} \sum_{x \in S} d_{s_0}^{\pi}(x) \left[\sum_{a' \in \mathcal{A}} \frac{\partial \pi(a'|x)}{\partial \pi(a|s)} Q^{\pi}(x,a') + \frac{\partial h^{\pi}(x)}{\partial \pi(a|s)} \right] \\
&= \frac{1}{1-\gamma} d_{s_0}^{\pi}(s) \left[Q^{\pi}(s,a) + \frac{\partial h^{\pi}(s)}{\partial \pi(a|s)} \right],
\end{aligned}$$

where the second equality follows by expanding $\frac{\partial V^{\pi}(s')}{\partial \pi(a|s)}$ recursively, and the third equality follows from the definition of $d_{s_0}^{\pi}(s)$ in (2.2.11), and the last identity follows from $\frac{\partial \pi(a'|x)}{\partial \pi(a|s)} = 0$ for $x \neq s$ or $a' \neq a$, and $\frac{\partial h^{\pi}(x)}{\partial \pi(a|s)} = 0$ for $x \neq s$. □

In view of Lemma 2.1, the gradient of the objective $f(\pi)$ in (2.3.6) at the optimal policy $\pi^{\star}$ is given by

$$\begin{aligned}
\frac{\partial f(\pi^{\star})}{\partial \pi(a|s)} &= \mathbb{E}_{s_0 \sim \nu^{\star}} \left[\frac{1}{1-\gamma} d_{s_0}^{\pi^{\star}}(s) \left[Q^{\pi^{\star}}(s,a) + \nabla h^{\pi^{\star}}(s,a) \right] \right], \\
&\sum_{t=0}^{\infty} \gamma^t (\nu^{\star})^T (\mathcal{P}^{\pi^{\star}})^t e_s \left[Q^{\pi^{\star}}(s,a) + \nabla h^{\pi^{\star}}(s,a) \right] \\
&= \frac{1}{1-\gamma} (\nu^{\star})^T e_s \left[Q^{\pi^{\star}}(s,a) + \nabla h^{\pi^{\star}}(s,a) \right] \\
&= \frac{1}{1-\gamma} \nu^{\star}(s) \left[Q^{\pi^{\star}}(s,a) + \nabla h^{\pi^{\star}}(s,a) \right], \qquad (2.3.7)
\end{aligned}$$

where the third identity follows from (2.2.11) and the last one follows from (2.2.15) which implies $(\nu^{\star})^T (\mathcal{P}^{\pi^{\star}})^t = (\nu^{\star})^T$ for any $t \geq 0$ since $\nu^{\star}$ is the steady-state distribution of $\pi^{\star}$. Therefore, the optimality condition of (2.3.6) suggests that we solve the following

variational inequality:

$$\mathbb{E}_{s\sim\nu^\star}\left\langle Q^{\pi^\star}(s,\cdot)+\nabla h^{\pi^\star}(s,\cdot),\pi(\cdot|s)-\pi^\star(\cdot|s)\right\rangle\geq 0. \tag{2.3.8}$$

However, the above VI requires h^π to be differentiable. In order to handle the possible non-smoothness of h^π, we instead solve the following problem:

$$\mathbb{E}_{s\sim\nu^\star}\left[\left\langle Q^{\pi^\star}(s,\cdot),\pi(\cdot|s)-\pi^\star(\cdot|s)\right\rangle+h^\pi(s)-h^{\pi^\star}(s)\right]\geq 0. \tag{2.3.9}$$

It turns out this variational inequality satisfies certain generalized monotonicity properties thanks to the following performance difference lemma obtained by generalizing some previous results (e.g., Lemma 6.1 of Kakade and Langford [2002]).

Lemma 2.2 (Performance difference, Lan [2022] Lemma 2). For any two feasible policies π and π', we have

$$V^{\pi'}(s)-V^\pi(s)=\frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi'}}\left[\langle A^\pi(s',\cdot),\pi'(\cdot|s')\rangle+h^{\pi'}(s')-h^\pi(s')\right],$$

where

$$A^\pi(s',a):=Q^\pi(s',a)-V^\pi(s'). \tag{2.3.10}$$

Proof For simplicity, let us denote $\xi^{\pi'}(s_0)$ the random process $(s_t,a_t,s_{t+1}),t\geq 0$, generated by following the policy π' starting with the initial state s_0. It then follows from the definition of $V^{\pi'}$ that

$$V^{\pi'}(s)-V^\pi(s)=\mathbb{E}_{\xi^{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(c(s_t,a_t)+h^{\pi'}(s_t)\right)\right]-V^\pi(s)$$

$$=\mathbb{E}_{\xi^{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(c(s_t,a_t)+h^{\pi'}(s_t)+V^\pi(s_t)-V^\pi(s_t)\right)\right]-V^\pi(s)$$

$$\overset{(a)}{=}\mathbb{E}_{\xi^{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(c(s_t,a_t)+h^{\pi'}(s_t)+\gamma V^\pi(s_{t+1})-V^\pi(s_t)\right)\right]+\mathbb{E}_{\xi^{\pi'}(s)}\left[V^\pi(s_0)-V^\pi(s)\right]$$

$$\overset{(b)}{=}\mathbb{E}_{\xi_{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(c(s_t,a_t)+h^{\pi'}(s_t)+\gamma V^\pi(s_{t+1})-V^\pi(s_t)\right)\right]$$

$$=\mathbb{E}_{\xi_{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(c(s_t,a_t)+h^\pi(s_t)+\gamma V^\pi(s_{t+1})-V^\pi(s_t)\right)+h^{\pi'}(s_t)-h^\pi(s_t)\right]$$

$$\overset{(c)}{=}\mathbb{E}_{\xi^{\pi'}(s)}\left[\sum_{t=0}^\infty\gamma^t\left(Q^\pi(s_t,a_t)-V^\pi(s_t)+h^{\pi'}(s_t)-h^\pi(s_t)\right)\right]$$

where (a) follows by taking the term $V^{\pi}(s_0)$ outside the summation, (b) follows from the fact that $\mathbb{E}_{\xi^{\pi'}}[V^{\pi}(s_0)] = V^{\pi}(s)$ since the random process starts with $s_0 = s$, and (c) follows from (2.2.3) which clearly holds for the regularized definitions of the Q-function in the analogous setting. The previous conclusion, together with (2.3.10) and the definition $d_s^{\pi'}$ in (2.2.12), then implies that

$$
\begin{aligned}
&V^{\pi'}(s) - V^{\pi}(s) \\
&= \frac{1}{1-\gamma} \sum_{s' \in S} \sum_{a' \in A} d_s^{\pi'}(s') \pi'(a'|s') \left[A^{\pi}(s', a') + h^{\pi'}(s') - h^{\pi}(s') \right] \\
&= \frac{1}{1-\gamma} \sum_{s' \in S} d_{s'}^{\pi'}(s') \left[\langle A^{\pi}(s', \cdot), \pi'(\cdot|s') \rangle + h^{\pi'}(s') - h^{\pi}(s') \right].
\end{aligned}
$$

□

Lemma 2.3(Lan [2022] Lemma 3). The VI problem in (2.3.9) satisfies

$$
\begin{aligned}
&\mathbb{E}_{s \sim \nu^\star} \left[\langle Q^{\pi}(s, \cdot), \pi(\cdot|s) - \pi^\star(\cdot|s) \rangle + h^{\pi}(s) - h^{\pi^\star}(s) \right] \\
&= \mathbb{E}_{s \sim \nu^\star} \left[(1-\gamma)(V^{\pi}(s) - V^{\pi^\star}(s)) \right]. \qquad (2.3.11)
\end{aligned}
$$

Proof. It follows from Lemma 2.2 (with $\pi' = \pi^\star$) that

$$
\begin{aligned}
&(1-\gamma)[V^{\pi^\star}(s) - V^{\pi}(s)] \\
&= \mathbb{E}_{s' \sim d_s^{\pi^\star}} \left[\langle A^{\pi}(s', \cdot), \pi^\star(\cdot|s') \rangle + h^{\pi^\star}(s') - h^{\pi}(s') \right].
\end{aligned}
$$

Let e denote the vector of all 1's. Then, we have

$$
\begin{aligned}
\langle A^{\pi}(s', \cdot), \pi^\star(\cdot|s') \rangle &= \langle Q^{\pi}(s', \cdot) - V^{\pi}(s')e, \pi^\star(\cdot|s') \rangle \\
&= \langle Q^{\pi}(s', \cdot), \pi^\star(\cdot|s') \rangle - V^{\pi}(s') \\
&= \langle Q^{\pi}(s', \cdot), \pi^\star(\cdot|s') \rangle - \langle Q^{\pi}(s', \cdot), \pi(\cdot|s') \rangle \\
&= \langle Q^{\pi}(s', \cdot), \pi^\star(\cdot|s') - \pi(\cdot|s') \rangle. \qquad (2.3.12)
\end{aligned}
$$

The first identity follows from the definition of $A^{\pi}(s', \cdot)$ in (2.3.10), the second equality follows from the fact that $\langle e, \pi^\star(\cdot|s') \rangle = 1$, and the third equality follows from the definition of V^{π} in (2.3.2). Combining the above two relations and taking the expectation w.r.t. $\nu^\star$, we obtain

$$
\begin{aligned}
&(1-\gamma)\mathbb{E}_{s \sim \nu^\star}[V^{\pi^\star}(s) - V^{\pi}(s)] \\
&= \mathbb{E}_{s \sim \nu^\star, s' \sim d_s^{\pi^\star}} \left[\langle Q^{\pi}(s', \cdot), \pi^\star(\cdot|s') - \pi(\cdot|s') \rangle + h^{\pi^\star}(s') - h^{\pi}(s') \right]
\end{aligned}
$$

$$= \mathbb{E}_{s\sim\nu^\star}\left[\langle Q^\pi(s,\cdot), \pi^\star(\cdot|s) - \pi(\cdot|s)\rangle + h^\pi(s) - h^{\pi^\star}(s)\right],$$

Where the second identity follows similarly to (2.3.7) since $\nu^\star$ is the steady state distribution induced by $\pi^\star$. The result is then followed by rearranging the terms. □

Since $V^\pi(s) - V^{\pi^\star}(s) \geq 0$ for any feasible policy π, as $V^{\pi^\star}(s)$ is the solution to (2.3.6), consequently (2.3.9) holds, then we conclude from Lemma 2.3 that

$$\mathbb{E}_{s\sim\nu^\star}\left[\langle Q^\pi(s,\cdot), \pi(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^\pi(s) - h^{\pi^\star}(s)\right] \geq 0.$$

Therefore, the VI in (2.3.9) satisfies the generalized monotonicity (1.4.3) (see, also Lan [2021] 3.8.14).

Chapter 3

Policy mirror descent and stochastic policy mirror descent

3.1 Deterministic policy mirror descent

Mirror descent is a theoretically well-understood optimization framework (A. Nemirovski and D. B. Yudin [1983], Beck and Teboulle [2003], Bubeck [2015], Hazan [2023]), yet only recently it has been investigated for policy optimization in RL Geist et al. [2019], Liu et al. [2019], Shani et al. [2020]), and there remains considerable scope for further research in this area (Tomar et al. [2020]).

The **Policy Mirror Descent (PMD)** method, introduced by Lan [2022], is a noble application of mirror descent in RL. The PMD method is designed to update policies in reinforcement learning problems by minimizing a *proximal objective* that combines both the policy's expected cost and a divergence term that measures the "distance" between consecutive policies in terms of Bregman divergence. The proximal mapping incorporates a mirror map that leverages the non-Euclidean Kullback-Leibler (KL) divergence, effectively adapting updates to the underlying geometry of the optimization problem.

In PMD (Lan [2022], 3.1), each policy update is obtained by solving an optimization problem of the form:

$$\pi_{k+1}(\cdot|s) = \underset{p(\cdot|s)\in\Delta_{|\mathcal{A}|}}{\arg\min} \left\{\eta_k \left[\langle Q^{\pi_k}(s,\cdot), p(\cdot|s)\rangle + h^p(s)\right] + D^p_{\pi_k}(s)\right\}, \quad \forall s \in \mathcal{S} \tag{3.1.1}$$

where:

- $Q^{\pi_k}(s,\cdot)$ represents the Q-function under the current policy π_k,
- $h^p(s)$ is a $\mu-$strongly convex function as the regularizer,

- $D^p_{\pi_k}(s)$ is the Bregman divergence (KL-divergence in this context) measuring the "distance" between the current policy π_k and the candidate policy $p(\cdot|s)$,
- η_k is the step size controlling the update magnitude.

The structural difference between this proximal update and the mirror(1.3.1) descent update can be observed due to the appearance of $h^p(s)$ in the expression, which makes Algorithm 1 infeasible in this case for general h, and one needs to rely on iterative convex optimization methods to solve the optimization problem. For the special case of weighted negative entropy regularizer $h^p(s) = \tau H(p(\cdot \mid s))$, discussed in Example 2.1, we still can obtain a closed form update by rephrasing the objective function. The PMD algorithm (Lan [2022], Algorithm 1) is as follows

Algorithm 2 The policy mirror descent method

Require: Initial points π_0 and step sizes $\eta_k \geq 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: $\pi_{k+1}(\cdot|s) = \underset{p(\cdot|s)\in\Delta_{|\mathcal{A}|}}{\arg\min} \left\{\eta_k \left[\langle Q^{\pi_k}(s,\cdot), p(\cdot|s)\rangle + h^p(s)\right] + D^p_{\pi_k}(s)\right\}, \quad \forall s \in S$

3: **end for**

From this point on, for the sake of simplicity, we consider

$$\pi_0(a|s) = 1/|\mathcal{A}|, \quad \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$$

which is the uniform distribution, i.e. maximum entropy distribution, over actions given a state s. In this case, we have

$$D^{\pi}_{\pi_0} = \sum_{a\in\mathcal{A}} \pi(a|s)\log(\pi(a|s)) + \log(|\mathcal{A}|) \leq \log(|\mathcal{A}|), \quad \forall \pi(\cdot|s) \in \Delta_{|\mathcal{A}|}$$

In the following, the general convergence properties for the PMD method are stated. Unlike the classical policy iteration or value iteration methods used in Markov Decision Processes (Puterman [1994](6.3, 6.4), Sutton and Barto [2020] (4), this analysis does not rely on the contraction properties of Bellman's operator (Lan [2022]). Instead, it leverages the three-point lemma associated with the optimality condition of the problem (3.1.1) to prove the convergence of the algorithm. This approach also deviates significantly from the standard mirror descent method in convex optimization. First, while the classic mirror descent method requires the convexity of the objective function, the analysis of PMD is based on the generalized monotonicity outlined in Lemma 2.3. Second, whereas mirror descent typically exploits the Lipschitz or smoothness properties of the objective function, the PMD method demonstrates iterative progress by employing both the performance difference lemma and the three-point lemma. Consequently, no assumptions regarding the

smoothness of the objective function are required in the analysis (Lan [2022]).

Lemma 3.1 (Extension of the Three-Point Lemma (1.1), Lan [2022] Lemma 4). For any $p(\cdot|s) \in \Delta_{|\mathcal{A}|}$ we have

$$\eta_k \left[\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - p(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^p(s)\right]$$

$$+D_{\pi_k}^{\pi_{k+1}}(s) \leq D_{\pi_k}^p(s) - (1+\eta_k\mu)D_{\pi_{k+1}}^p(s) \tag{3.1.2}$$

Proof. By the optimality condition of (3.1.1),

$$\langle \eta_k[Q^{\pi_k}(s,\cdot)+(h')^{\pi_{k+1}}(s,\cdot)]+\nabla D_{\pi_k}^{\pi_{k+1}}(s,\cdot), p(\cdot|s)-\pi_{k+1}(\cdot|s)\rangle \geq 0, \quad \forall p(\cdot|s) \in \Delta_{|\mathcal{A}|}, \tag{3.1.3}$$

where $(h')^{\pi_{k+1}}$ denotes the subgradient of h at π_{k+1} and $\nabla D_{\pi_k}^{\pi_{k+1}}(s,\cdot)$ denotes the gradient of $D_{\pi_k}^{\pi_{k+1}}(s)$ at π_{k+1}. Using the The Three-point lemma (lemma 1.1) for

$$a = \pi_{k+1,}, \quad b = p(\cdot|s), \quad c = \pi_k, \text{ we have}$$

$$D_{\pi_k}^p(s) = D_{\pi_k}^{\pi_{k+1}}(s) + \langle \nabla D_{\pi_k}^{\pi_{k+1}}(s,\cdot), p(\cdot|s) - \pi_{k+1}(\cdot|s)\rangle + D_{\pi_{k+1}}^p(s). \tag{3.1.4}$$

The result is immediately followed by combining (3.1.3) and (3.1.4) together with strong convexity inequality in (2.3.1) to replace $(h')^{\pi_{k+1}}$. □

3.2 PMD convergence analysis

In this section, we observe that, when employing a constant step size rule, the policy mirror descent method (PMD) achieves a linear convergence rate for solving reinforcement learning problems with strongly convex regularizers. This linear convergence is particularly noteworthy as it ensures that the algorithm converges to an optimal policy efficiently, even in high-dimensional or complex environments commonly encountered in RL applications. The strongly convex regularizers play a crucial role in this context by providing a well-behaved optimization landscape, which facilitates faster convergence as will be demonstrated later (Lan [2022]).

The following lemma establishes the descent property of PMD in each iteration. Specifically, it demonstrates that the PMD update rule consistently reduces the objective function value, thereby ensuring progress towards the optimal policy. This descent property is fundamental as it underpins the theoretical guarantees of Algorithm 2, which will be revisited shortly in the proof of Theorem 1.

Lemma 3.2 (Descent property of PMD, Lan [2022], Lemma 5). For any $s \in \mathcal{S}$, we have

$$V^{\pi_{k+1}}(s) \leq V^{\pi_k}(s), \tag{3.2.1}$$

$$\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s) \geq V^{\pi_{k+1}}(s) - V^{\pi_k}(s). \tag{3.2.2}$$

Proof. It follows from Lemma 2.2 (with $\pi' = \pi_{k+1}, \pi = \pi_k$) that

$$V^{\pi_{k+1}}(s) - V^{\pi_k}(s) =$$

$$\frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\mathcal{A}^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') + h^{\pi_{k+1}}(s') - h_{\pi_k}(s')\right]. \tag{3.2.3}$$

Similarly to (2.3.11), we can show that

$$\langle A^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s')\rangle = \langle Q^{\pi_k}(s',\cdot) - V^{\pi_k}(s'), \pi_{k+1}(\cdot \mid s')\rangle$$

$$= \langle Q^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle.$$

where the last equality follows from (2.2.4), then combining the above two identities, we then obtain

$$V^{\pi_{k+1}}(s) - V^{\pi_k}(s) =$$

$$\frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right]. \tag{3.2.3}$$

Now we conclude from Lemma 3.1 applied to (3.1.1) with $p(\cdot|s') = \pi_k(\cdot|s')$ that

$$\langle Q^{\pi_k}(s'), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')$$

$$\leq -\frac{1}{\eta_k}\left[(1+\eta_k\mu)D_{\pi_k+1}^{\pi_k}(s') + D_{\pi_k}^{\pi_{k+1}}(s')\right]. \tag{3.2.3}$$

The previous two conclusions then clearly imply the result in (3.2.1). It also follows from (3.2.3) that

$$\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right]$$

$$\leq d_s^{\pi_k}(s)\left[\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s)\right]$$

$$\leq (1-\gamma)\left[\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s)\right], \tag{3.2.4}$$

where the last inequality follows from the fact that $d_s^{\pi_{k+1}}(s) \geq (1-\gamma)$ due to the definition of $d_s^{\pi_{k+1}}$ in (2.2.11). The result in (3.2.1) then follows immediately from (3.2.3) and the above inequality.

Now, in the next theorem with a constant stepsize rule, the PMD method is proven to achieve a linear rate of convergence for solving RL problems with strongly convex regularizers (i.e., $\mu > 0$).

Theorem 1 (Lan [2022], Theorem 1). Suppose that $\eta_k = \eta$ for any $k \geq 0$ in the PMD method with

$$1 + \eta\mu \geq \frac{1}{\gamma}. \tag{3.2.5}$$

Then we have

$$f(\pi_k) - f(\pi^\star) + \frac{\mu}{1-\gamma}\mathcal{D}(\pi_k, \pi^\star) \leq \gamma^k \left[f(\pi_0) - f(\pi^\star) + \frac{\mu}{1-\gamma}\log|\mathcal{A}| \right]$$

for any $k \geq 0$, where

$$\mathcal{D}(\pi_k, \pi^\star) := \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s) \right]. \tag{3.2.6}$$

Proof. By Lemma (3.1) applied to (3.1.1) (with $\eta_k = \eta$ and $p = \pi^\star$), we have

$$\eta\left[\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi^\star}(s) \right]$$

$$+ D^{\pi_{k+1}}_{\pi_k}(s) \leq D^{\pi^\star}_{\pi_k}(s) - (1+\eta\mu) D^{\pi^\star}_{\pi_{k+1}}(s),$$

which, in view of (3.2.2), then implies that

$$\eta\left[\langle Q^{\pi_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^{\pi_k}(s) - h^{\pi^\star}(s) \right]$$

$$+\eta\left[V^{\pi_{k+1}}(s) - V^{\pi^k}(s) \right] + D^{\pi_{k+1}}_{\pi_k}(s) \leq D^{\pi^\star}_{\pi_k}(s) - (1+\eta\mu) D^{\pi^\star}_{\pi_{k+1}}(s).$$

Taking expectation w.r.t. $\nu^\star$ on both sides of the above inequality and using Lemma (2.3), we arrive at

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta(1-\gamma)(V^{\pi_k}(s) - V^{\pi^\star}(s)) \right] + \eta\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_{k+1}}(s) - V^{\pi_k}(s) \right] + \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi_{k+1}}_{\pi_k}(s) \right]$$

$$\leq \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s) - (1+\eta\mu) D^{\pi^\star}_{\pi_{k+1}}(s) \right].$$

Noting $V^{\pi_{k+1}}(s) - V^{\pi^k}(s) = V^{\pi_{k+1}}(s) - V^{\pi^\star}(s) - [V^{\pi_k}(s) - V^{\pi^\star}(s)]$ and rearranging the terms in the above inequality, we have

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta(V^{\pi_{k+1}}(s) - V^{\pi^\star}(s)) + (1+\eta\mu) D^{\pi^\star}_{\pi_{k+1}}(s) \right] + \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi_{k+1}}_{\pi^k}(s) \right]$$

$$\leq \gamma\mathbb{E}_{s\sim\nu^\star}\left[\eta(V^{\pi_k}(s) - V^{\pi^\star}(s)) + \frac{1}{\gamma} D^{\pi^\star}_{\pi_k}(s) \right], \tag{3.2.7}$$

which, in view of the assumption (3.2.5) and the definition of f in (2.3.6), gives

$$f(\pi_{k+1}) - f(\pi^\star) + \frac{\mu}{1-\gamma}\mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_{k+1}}(s)\right]$$

$$\leq \gamma\left[f(\pi_k) - f(\pi^\star) + \frac{\mu}{1-\gamma}\mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s)\right]\right].$$

Applying this relation recursively and using the upper bound of $\log(|\mathcal{A}|)$ on $D^{\pi_k}_{\pi_0}(s)$, we then conclude the result. □

Remark 1. Theorem 1 highlights the role of a strongly convex regularizer in achieving linear convergence. In equation (3.2.7), it is necessary to align the coefficients of $D^{\pi^\star}_{\pi_{k+1}}$ on the left-hand side and $D^{\pi^\star}_{\pi_k}$ on the right-hand side of the inequality. This alignment allows us to apply a recursive argument that iteratively reduces the inequality down to $f(\pi_0) - f(\pi^\star)$, with a factor of γ introduced at each iteration. By multiplying both sides by $1/\eta$, and solving

$$\frac{1+\eta\mu}{\eta} = \frac{1}{\gamma\eta}$$

for $\eta \geq 0$, we obtain $\eta = \frac{1-\gamma}{\gamma\mu}$. Considering this value for η the coefficient of both $D^{\pi^\star}_{\pi_{k+1}}$ and $D^{\pi^\star}_{\pi_k}$ will be $\frac{\mu}{1-\gamma}$. In order to be able to make this choice of stepsize to unify the coefficient of the two mentioned terms, we need the following inequality to hold

$$\eta \geq \frac{1-\gamma}{\gamma\mu} \Rightarrow \eta\mu\gamma \geq 1-\gamma$$

$$\Rightarrow (\eta\mu+1)\gamma \geq 1 \geq \frac{1}{\gamma}$$

Consequently, this analysis justifies the specific choice of stepsize in Theorem 1. Moreover, this scheme provides the rationale for the stepsize selection in the convergence guarantees of algorithms APMD and SPMD, which will be discussed later. However, in Theorem 2, we observe that the absence of strong convexity in the regularizer precludes the use of this recursive argument, resulting in only a sub-linear convergence guarantee within Guanghui Lan's proof framework.

Remark 2. Theorem 1 not only asserts that the optimality gap converges to zero but also demonstrates that the expected Kullback-Leibler (KL) divergence in equation(3.2.6),

$$\mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s)\right],$$

approaches zero. This convergence in expected KL divergence indicates that the distributions induced by the policies π_k and $\pi^\star$ become increasingly similar. Consequently, as the iteration k progresses, the policy π_k becomes an excellent approximation of the optimal

policy $\pi^\star$.

According to Theorem 1, the PMD method converges linearly in terms of both function value and the distance to the optimal solution for solving RL problems with strongly convex regularizers. Now we will see that a direct application of the PMD method only achieves a sublinear rate of convergence for the case when $\mu = 0$.

Theorem 2 (Lan [2022], Theorem 2) Suppose that $\eta_k = \eta$ in the PMD method. Then we have

$$f(\pi_{k+1}) - f(\pi^\star) \leq \frac{\eta\gamma\left[f(\pi_0) - f(\pi^\star)\right] + \log|\mathcal{A}|}{\eta(1-\gamma)(k+1)}. \tag{3.2.8}$$

for any $k \geq 0$

Proof. It follows from (3.2.7) with $\mu = 0$ that

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta(V^{\pi_{k+1}}(s) - V^{\pi^\star}(s)) + D^{\pi^\star}_{\pi_{k+1}}(s)\right] + \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s)\right]$$

$$\leq \eta\gamma\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_k}(s) - V^{\pi^\star}(s)\right] + \mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s)\right].$$

Taking the telescopic sum of the above inequalities and using the fact that $V^{\pi_{k+1}}(s) \leq V^{\pi^k}(s)$ due to (3.2.1), we obtain

$$(k+1)\eta(1-\gamma)\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_{k+1}}(s) - V^{\pi^\star}(s)\right]$$

$$\leq \mathbb{E}_{s\sim\nu^\star}\left[\eta V^{\pi_0}(s) - V^{\pi^\star}(s)\right] + D^{\pi^\star}_{\pi_0}(s),$$

which clearly implies the result in view of the definition of f in (2.3.6) and the upper bound of $\log(|\mathcal{A}|)$ on $D^{\pi^\star}_{\pi_0}$ in (3.4). □

According to Lan [2022], the result in Theorem 2 shows that the PMD method requires $\mathcal{O}(1/(1-\gamma)\epsilon)$ iterations to find an ϵ-solution for general RL problems. This bound already matches, in terms of its dependence on $(1-\gamma)$ and ϵ, the previously best-known complexity for natural policy gradient methods (Agarwal et al. [2021]). This theorem addressed the scenario where the regularizer is not strictly convex. Within PMD framework, achieving a linear convergence rate is not established for a non-strictly convex regularizer, and this theorem proves only a sublinear convergence rate. We will further analyze an enhancement of the PMD method, APMD (Lan [2022]), which can achieve a linear rate of convergence for the case when $\mu = 0$, in the next chapter.

3.3 SPMD: Stochastic policy mirror descent

In the preceding section, we investigated the exact Policy Mirror Descent (PMD) method. The exact policy mirror descent method requires the input of the exact action-value functions, which is hardly realizable in real-world settings. In this section, we focus on Stochastic Policy Mirror Descent (SPMD), Lan [2022] (Section 4), wherein a certain level of error in Q-function estimation is assumed that propagates with each iteration. Under relatively strong assumptions regarding these estimation errors, Lan [2022] again establishes linear convergence. The lemmas and argument structures employed in the proof of the relevant theorem closely mirror those presented in the previous section, adhering to the same logical framework while incorporating adjustments to account for the noise introduced by estimation inaccuracies.

We assume that for a given policy π_k, there exists a stochastic estimator Q^{π_k,ξ_k} such that

$$\mathbb{E}_{\xi_k}\left[Q^{\pi_k,\xi_k}\right] = \bar{Q}^{\pi_k}, \tag{3.3.1}$$

$$\mathbb{E}_{\xi_k}\left[\|Q^{\pi_k,\xi_k} - Q^{\pi_k}\|_\infty^2\right] \leq \sigma_k^2, \tag{3.3.2}$$

$$\|\bar{Q}^{\pi_k} - Q^{\pi_k}\|_\infty \leq \zeta_k, \tag{3.3.3}$$

$$\delta_k = Q^{\pi_k,\xi_k} - Q^{\pi_k} \tag{3.3.4}$$

then, the Stochastic Policy Mirror Descent algorithm is derived by substituting Q^{π_k} in equation (3.1.1) with its stochastic estimator Q^{π_k,ξ_k}, namely,

$$\pi_{k+1}(\cdot|s) = \underset{p(\cdot|s)\in\Delta_{|\mathcal{A}|}}{\arg\min}\left\{\Phi_k(p) := \eta_k\left[\langle Q^{\pi_k,\xi_k}(s,\cdot), p(\cdot|s)\rangle + h^p(s)\right] + D^p_{\pi_k}(s)\right\}.$$

This substitution integrates the stochastic nature of the estimator into the PMD framework, allowing the algorithm to operate in settings where the exact computation of Q^{π_k} is infeasible. By carefully managing the bias ζ_k and variance σ_k^2 introduced by the stochastic estimator, the SPMD method achieves a balance between computational tractability and convergence guarantees, paving the way for efficient policy optimization in reinforcement learning real-world applications.

First, by assumptions (3.3.1) and (3.3.3), we can make the following decomposition (Lan [2022]):

$$\begin{aligned}\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle &= \langle Q^{\pi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle \\ &+\langle \bar{Q}^{\pi_k}(s,\cdot) - Q^{\pi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle\end{aligned}$$

$$+\langle Q^{\pi_k,\xi_k}(s,\cdot) - \bar{Q}^{\pi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle,$$

and by applying the expectation, it's straightforward to see

$$\mathbb{E}_{\xi_k}\left[\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle \,|\, \xi_{[k-1]}\right]$$

$$\geq \langle Q^{\pi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle - 2\zeta_k. \tag{3.3.5}$$

Similar to Lemma 3.2 (Descent property of PMD), below, we have some general convergence properties of the SPMD method. Unlike PMD, SPMD does not guarantee the non-increasing property of $V^{\pi_k}(s)$ anymore.

Lemma 3.3 (Lan [2022], Lemma 12.) For any $s \in \mathcal{S}$, we have

$$V^{\pi_{k+1}}(s) - V^{\pi_k}(s) \leq \langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_{k+1}(\cdot \mid s) - \pi_k(\cdot \mid s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s)$$

$$+\frac{1}{\eta_k} D_{\pi_k}^{\pi_{k+1}}(s) + \frac{\eta_k \|\delta_k\|_\infty^2}{2(1-\gamma)}. \tag{3.3.6}$$

Proof. The equation (3.2.3) in the proof of Lemma 3.2 still holds in this case; therefore, we have

$$V^{\pi_{k+1}}(s) - V^{\pi_k}(s)$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right]$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k,\xi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle\right.$$

$$\left.-\langle \delta_k, \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right]$$

$$\leq \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k,\xi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right]$$

$$+\frac{1}{2\eta_k}\|\pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\|_1^2 + \frac{\eta_k\|\delta_k\|_\infty^2}{2}$$

$$\leq \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k,\xi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s')\right.$$

$$\left.+\frac{1}{\eta_k} D_{\pi_k}^{\pi_{k+1}}(s') + \frac{\eta_k\|\delta_k\|_\infty^2}{2}\right], \tag{3.3.7}$$

where the first inequality follows from Young's inequality and the second one follows from the strong convexity of $D_{\pi_k}^{\pi_{k+1}}$ w.r.t. $\|\cdot\|_1$ (by using Pinsker's inequality on (1.2.4). Plus, from Lemma 3.1 (Extension of the Three-Point Lemma) applied to $\arg\min_{p(\cdot|s)\in\Delta_{|\mathcal{A}|}}\{\Phi_k(p)\}$ with Q^{π_k} replaced by Q^{π_k,ξ_k} and $p(\cdot \mid s') = \pi_k(\cdot \mid s')$ we have

$$\langle Q^{\pi_k,\xi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s') + \frac{1}{\eta_k} D_{\pi_k}^{\pi_{k+1}}(s')$$

$$\leq -\frac{1}{\eta_k}\left[(1+\eta_k\mu) D_{\pi_k}^{\pi_{k+1}}(s')\right] \leq 0,$$

which implies that

$$\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q^{\pi_k,\xi_k}(s',\cdot), \pi_{k+1}(\cdot \mid s') - \pi_k(\cdot \mid s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s') + \frac{1}{\eta_k} D_{\pi_{k+1}}^{\pi_k}(s')\right]$$

$$\leq d_s^{\pi_{k+1}}(s)\left[\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_{k+1}(\cdot \mid s) - \pi_k(\cdot \mid s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s) + \frac{1}{\eta_k} D_{\pi_k}^{\pi_{k+1}}(s)\right]$$

$$\leq (1-\gamma)\left[\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_{k+1}(\cdot \mid s) - \pi_k(\cdot \mid s) + h^{\pi_{k+1}}(s) - h^{\pi_k}(s) + \frac{1}{\eta_k} D_{\pi_k}^{\pi_{k+1}}(s)\right],$$

where the last inequality follows from (2.2.14), and the result in (3.3.6) follows immediately from (3.3.7) and the above inequality. □

The following lemma establishes an important recursion about the SPMD method.

Lemma 3.4 (Lan [2022], Lemma 13) For any $k \geq 0$, we have

$$\mathbb{E}_{\xi_{[k]}}\left[f(\pi_{k+1}) - f(\pi^\star) + \left(\frac{1}{\eta_k} + \mu\right) D(\pi_{k+1}, \pi^\star)\right]$$

$$\leq \mathbb{E}_{\xi_{[k-1]}}\left[\gamma\left(f(\pi_k) - f(\pi^\star)\right) + \frac{1}{\eta_k} D(\pi_k, \pi^\star)\right] + 2\zeta_k + \frac{\eta_k \sigma_k^2}{2(1-\gamma)}. \tag{3.3.4}$$

Proof. By applying Lemma 3.1 (Extension of the Three-Point Lemma) to (3.1.1), with Q^{π_k} replaced by Q^{π_k,ξ_k} and setting $p = \pi^\star$, we have

$$\eta_k[\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_{k+1}(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi^\star}(s)] + D_{\pi_k}^{\pi_{k+1}}(s)$$

$$\leq D_{\pi_k}^{\pi^\star}(s) - (1+\eta_k\mu) D_{\pi_{k+1}}^{\pi^\star}(s),$$

which, by lemma 3.3 implies that

$$\langle Q^{\pi_k,\xi_k}(s,\cdot), \pi_k(\cdot \mid s) - \pi^\star(\cdot \mid s)\rangle + h^{\pi_k}(s) - h^{\pi^\star}(s) + V^{\pi_{k+1}}(s) - V^{\pi_k}(s)$$

$$\leq \frac{1}{\eta_k} D_{\pi_k}^{\pi^\star}(s) - \left(\frac{1}{\eta_k} + \mu\right) D_{\pi_{k+1}}^{\pi^\star}(s) + \frac{\eta_k \|\delta_k\|_\infty^2}{2(1-\gamma)}.$$

Taking expectation w.r.t. $\xi_{[k]}$ and $\nu^\star$ on both sides of the above inequality, and using Lemma 2.3 and the relation in (3.3.5), we have

$$\mathbb{E}_{s\sim v^\star,\xi_{[k]}}\left[(1-\gamma)(V^{\pi_k}(s)-V^{\pi^\star}(s))+V^{\pi_{k+1}}(s)-V^{\pi_k}(s)\right]$$
$$\leq \mathbb{E}_{s\sim v^\star,\xi_{[k]}}\left[\frac{1}{\eta_k}D^{\pi^\star}_{\pi_k}(s)-\left(\frac{1}{\eta_k}+\mu\right)D^{\pi\star}_{\pi_{k+1}}(s)\right]+2\zeta_k+\frac{\eta_k\sigma_k^2}{2(1-\gamma)}.$$

By using the equation $V^{\pi_{k+1}}(s)-V^{\pi_k}(s)=V^{\pi_{k+1}}(s)-V^{\pi^\star}(s)-[V^{\pi_k}(s)-V^{\pi^\star}(s)]$, and rearranging the terms in the above inequality, and minding the definition of f in (2.3.6), we obtain the result. □

Lemma 3.4 is instrumental in proving Theorem 3, as expression (3.2.7) is utilized in establishing Theorem 1. By incorporating a specific choice of step size, we can employ a recursive argument that leads to the final result in Theorem 1, yet a noticeable difference in (3.3.4) is the added term of

$$2\zeta_k+\frac{\eta_k\sigma_k^2}{2(1-\gamma)} \tag{3.3.5}$$

which is due to the stochastic estimation of Q-function, the possibility of factorizing a term of γ in each iteration to obtain the linear convergence is not guaranteed. However, by the assumption of exponentially diminishing error terms in (3.3.1) and (3.3.2), we will still be able to bypass this issue, as proposed in the next theorem.

Theorem 3 (Lan [2022], Theorem 5). Suppose that $\eta_k=\eta=\frac{1-\gamma}{\gamma\mu}$ in the SPMD method. If $\zeta_k=2^{-(k/l)+2}$ and $\sigma_k^2=2^{-(k/l)+2}$ for any $k\geq 0$ with $l:=\lceil\log_\gamma(1/4)\rceil$, then

$$\mathbb{E}_{\xi_{[k-1]}}\left[f(\pi_k)-f(\pi^\star)+\frac{\mu}{1-\gamma}D(\pi_k,\pi^\star)\right]$$
$$\leq 2^{-k/l}\left[f(\pi_0)-f(\pi^\star)+\frac{1}{1-\gamma}\left(\mu\log|\mathcal{A}|+\frac{5}{2}+\frac{5}{8\gamma\mu}\right)\right]. \tag{3.3.6}$$

Proof. By Lemma 3.4 and the selection of η, we have

$$\mathbb{E}_{\xi_{[k]}}\left[f(\pi_{k+1})-f(\pi^\star)+\frac{\mu}{1-\gamma}\mathcal{D}(\pi_{k+1},\pi^\star)\right]$$
$$\leq \gamma\,\mathbb{E}_{\xi_{[k-1]}}\left[f(\pi_k)-f(\pi^\star)+\frac{\mu}{1-\gamma}\mathcal{D}(\pi_k,\pi^\star)\right]+2\zeta_k+\frac{\sigma_k^2}{2\gamma\mu},$$

which, in view of Lemma 4.6 (Lan [2022], Lemma 11) with

$$X_k=\mathbb{E}_{\xi_{[k-1]}}\left[f(\pi_k)-f(\pi^\star)+\frac{\mu}{1-\gamma}\mathcal{D}(\pi_k,\pi^\star)\right],\quad Z_k=2\zeta_k+\frac{\sigma_k^2}{2\gamma\mu}$$

we have

$$\mathbb{E}_{\xi_{[k-1]}}\left[f(\pi_k)-f(\pi^\star)+\frac{\mu}{1-\gamma}\mathcal{D}(\pi_k,\pi^\star)\right]$$

$$\leq \gamma^{\lfloor k/l \rfloor} \left[f(\pi_0) - f(\pi^\star) + \frac{\mu \mathcal{D}(\pi_0, \pi^\star)}{1-\gamma} + \frac{5}{4} \left(\frac{2}{1-\gamma} + \frac{1}{2\gamma(1-\gamma)\mu} \right) \right]$$

$$\leq \gamma^{\lfloor k/l \rfloor} \left[f(\pi_0) - f(\pi^\star) + \frac{1}{1-\gamma} \left(\mu \log |\mathcal{A}| + \frac{5}{2} + \frac{5}{8\gamma\mu} \right) \right].$$

□

By a specific assumption of exponentially diminishing error terms made in Theorem 3, as can be seen in (3.3.6), a linear convergence guarantee for the SPMD algorithm is established. In the multiple trajectory setting stated in (2.2.6), one can compute $Q^{\pi_k, \xi_k}(s, \cdot)$ by sampling trajectories.

Assuming the following upper bounds on the cost function and regularizer

$$c(s, a) \leq \bar{c}, \, \forall (s, a) \in \mathcal{S} \times \mathcal{A},$$

$$h^\pi(s) \leq \bar{h}, \, \forall s \in \mathcal{S}, \pi \in \Delta_{|\mathcal{A}|}.$$

For estimation error assumptions (3.3.2) and (3.3.3), Monte Carlo estimation introduced in (2.2.6)

$$Q^{\pi_k, \xi_k}(s, a) = \frac{1}{M_k} \sum_{i=1}^{M_k} \sum_{t=0}^{T_k - 1} \gamma^t \left[c(s_t^i, a_t^i) \right],$$

satisfies

$$\zeta_k = \frac{(\bar{c} + \bar{h}) \gamma^{T_k}}{1-\gamma} \quad \text{and} \quad \sigma_k^2 = \frac{(\bar{c} + \bar{h})^2}{(1-\gamma)^2} \left[\gamma^{2T_k} + \frac{\kappa (\log(|\mathcal{S}||\mathcal{A}|) + 1)}{M_k} \right], \tag{3.3.7}$$

For some absolute constant $\kappa > 0$ (see Lan [2022] Proposition 7 in the appendix). By choosing T_k and M_k properly, we can show the convergence of the SPMD method employed with stepsize rules as stated in Theorems 3.

Proposition 1 (Lan [2022]). Suppose that $\eta_k = \frac{1-\gamma}{\gamma\mu}$ in the SPMD method. If T_k and M_k are chosen such that

$$T_k \geq \frac{l}{2} \left(\lfloor k/l \rfloor + \log_2 \frac{\bar{c} + \bar{h}}{1-\gamma} + 2 \right) \quad \text{and} \quad M_k \geq \frac{(\bar{c} + \bar{h})^2 \kappa (\log(|\mathcal{S}||\mathcal{A}|) + 1)}{(1-\gamma)^2} 2^{\lfloor k/l \rfloor + 4}$$

with $l := \lceil \log_\gamma(1/4) \rceil$, then the relation in (3.3.6) holds. As a consequence, an ϵ-solution $\bar{\pi}$ such that $\mathbb{E}[f(\bar{\pi}) - f(\pi^\star) + \frac{\mu}{1-\gamma} D(\bar{\pi}, \pi^\star)] \leq \epsilon$, can be found in at most $\mathcal{O}(\log_\gamma \epsilon)$ SPMD iterations. In addition, the total number of samples for (s_t, a_t) pairs can be bounded by

$$\mathcal{O} \left(\frac{|\mathcal{S}||\mathcal{A}| \log |\mathcal{A}| \log(|\mathcal{S}||\mathcal{A}|) \log_\gamma(1/2) \log_\gamma \epsilon}{\mu (1-\gamma)^3 \epsilon} \right). \tag{}$$

Proof. (Proposition 1, Lan [2022])

This implies the sample complexity of $\mathcal{O}\left(\log(1/\epsilon)/\epsilon\right)$ of the SPMD algorithm. According to Lan [2022], this is the first time in the literature that such a sample complexity, after disregarding all constant factors, has been obtained for solving RL problems with strongly convex regularizers, even though problem (2.3.6) is still highly nonconvex. The previously best-known sampling complexity for RL problems with entropy regularizer was $\tilde{\mathcal{O}}(|\mathcal{S}||\mathcal{A}|^2/\epsilon^3)$ by Shani et al. [2020].

3.4 Exploration Dynamics and Distribution Mismatch in Policy Optimization

For any two probability distributions $\rho, \mu \in \Delta_{|\mathcal{S}|}$, where $\Delta_{|\mathcal{S}|}$ denotes the simplex over the state space $\mathcal{S}$, the distribution mismatch between ρ and μ quantifies the extent to which ρ deviates from μ across all states. This mismatch is formally defined using the infinity norm of the element-wise ratio of the distributions:

$$\left\| \frac{\rho}{\mu} \right\|_\infty := \max_{s \in \mathcal{S}} \frac{\rho(s)}{\mu(s)}$$

with the convention $0/0 = 1$. The infinity norm captures the largest relative discrepancy between ρ and μ across all states, effectively highlighting the worst-case scenario of distributional divergence. This measure is finite if and only if the support of μ contains the support of ρ. If μ is the uniform distribution, then the mismatch is bounded by $|\mathcal{S}|$. (Xiao [2022]). The convergence properties of policy gradient methods often depend on the distribution mismatch coefficients between two discounted state-visitation distributions (e.g., Kakade and Langford [2002], Agarwal et al. [2021].

In Theorem 1, the expected value function is being evaluated with regard to the steady state distribution of the optimal policy, $\nu^\star$. Assuming that we would like to extend this result to the case with another distribution over the states, ρ, given that $supp(\rho) = supp(\nu^\star)$, the following generalization could be made

$$\begin{aligned}
&\mathbb{E}_{s\sim\rho}\left[V^{\pi_k}(s) - V^{\pi^\star} + \frac{\mu}{1-\gamma} D_{\pi_k}^{\pi^\star}\right] \\
&= \sum_{s\in\mathcal{S}} \rho(s) \left[V^{\pi_k} - V^{\pi^\star} + \frac{\mu}{1-\gamma} D_{\pi_k}^{\pi^\star}\right] \frac{\nu^\star(s)}{\nu^\star(s)} \\
&\leq \sum_{s\in\mathcal{S}} \nu^\star(s) \left[V^{\pi_k} - V^\star + \frac{\mu}{1-\gamma} D_{\pi_k}^{\pi^\star}\right] \left\| \frac{\rho}{\nu^\star} \right\|_\infty
\end{aligned}$$

$$= \mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_k}(s) - V^{\pi^\star}(s) + \frac{\mu}{1-\gamma} D_{\pi_k}^{\pi^\star}\right] \left\|\frac{\rho}{\nu^\star}\right\|_\infty$$

Which is the LHS of Theorem 1 scaled by the distribution mismatch $\left\|\frac{\rho}{\nu^\star}\right\|_\infty$. Now, by Theorem 1, and using similar change of measure, we have

$$\mathbb{E}_{s\sim\rho}\left[V^{\pi_k}(s) - V^{\pi^\star} + \frac{\mu}{1-\gamma} D_{\pi_k}^{\pi^\star}\right]$$

$$\leq \left\|\frac{\rho}{\nu^\star}\right\|_\infty \gamma^k \left[\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_0} - V^{\pi^\star}\right] + \frac{\mu}{1-\gamma}\log(|\mathcal{A}|)\right]$$

$$\leq \left\|\frac{\rho}{\nu^\star}\right\|_\infty \gamma^k \left[\mathbb{E}_{s\sim\rho}\left[V^{\pi_0} - V^{\pi^\star}\right] \left\|\frac{\nu^\star}{\rho}\right\|_\infty + \frac{\mu}{1-\gamma}\log(|\mathcal{A}|)\right]$$

given that $\left\|\frac{\nu^\star}{\rho}\right\|_\infty \geq 1$, we have

$$\leq \left\|\frac{\rho}{\nu^\star}\right\|_\infty \left\|\frac{\nu^\star}{\rho}\right\|_\infty \gamma^k \left[\mathbb{E}_{s\sim\rho}\left[V^{\pi_0} - V^{\pi^\star}\right] + \frac{\mu}{1-\gamma}\log(|\mathcal{A}|)\right]$$

Impact of Distribution Mismatch in Stochastic Policy Mirror Descent. Regarding the Stochastic Policy Mirror Descent (SPMD) algorithm, the same measure-change strategy employed in the exact PMD framework can be effectively utilized. Under the assumptions on the error terms ζ_k and σ_k^2, we are able to factor out an exponentially decaying coefficient. This coefficient plays a pivotal role by counteracting the impact of the distribution mismatch term $\left\|\frac{\rho}{\nu^\star}\right\|_\infty \left\|\frac{\nu^\star}{\rho}\right\|_\infty$, thereby preserving the desired rate of convergence. Specifically, the exponential decay mitigates the amplification of stochastic errors introduced by the distribution mismatch, ensuring that the convergence rate remains linear. However, in scenarios where such strong assumptions on the error terms do not hold, the exponentially decaying factor may fail to emerge. Consequently, the coefficient $\left\|\frac{\rho}{\nu^\star}\right\|_\infty \left\|\frac{\nu^\star}{\rho}\right\|_\infty$ directly scales the stochastic error, thereby exacerbating its effect on the convergence rate. This highlights the critical significance of managing distribution mismatch, especially when employing stochastic estimations for Q-function as in (2.2.6) or Temporal Difference (TD). In the absence of a controlled distribution mismatch, the variance introduced by stochastic estimators can lead to substantial error propagation, undermining the stability and efficiency of the SPMD algorithm.

Chapter 4

Approximate policy mirror descent

4.1 Setup

In this chapter, we explore an enhancement to the foundational PMD method introduced by Lan [2022], which is integrated by an adaptive diminishing perturbation term into the definitions of the value function, Q-function, and proximal mapping.

Considering h to be a general convex function, i.e., not necessarily strictly convex, and for some $\tau \geq 0$ and a given initial policy $\pi_0(a|s) > 0, \forall s \in \mathcal{S}, a \in \mathcal{A}$, we have the perturbed action-value and state-value functions, respectively, by

$$Q_\tau^\pi(s,a) := \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t \left[c(s_t,a_t) + h^\pi(s_t) + \tau D_{\pi_0}^\pi(s_t)\right]$$
$$|s_0 = s, a_0 = a, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t,a_t)], \tag{4.1.1}$$
$$V_\tau^\pi(s) := \langle Q_\tau^\pi(s,\cdot), \pi(\cdot|s)\rangle. \tag{4.1.2}$$

Clearly, if $\tau = 0$, then the perturbed value functions reduce to the usual value functions, i.e.,

$$Q_0^\pi(s,a) = Q^\pi(s,a) \quad \text{and} \quad V_0^\pi(s) = V^\pi(s).$$

The following result relates the value functions with different τ.

Lemma 4.1 (Lan [2022], Lemma 6). For any given $\tau, \tau' \geq 0$, we have

$$V_\tau^\pi(s) - V_{\tau'}^\pi(s) = \frac{\tau - \tau'}{1-\gamma} \mathbb{E}_{s' \sim d_s^\pi} \left[D_{\pi_0}^\pi(s')\right]. \tag{4.1.3}$$

As a consequence, if $\tau \geq \tau' \geq 0$ then

$$V_{\tau'}^\pi(s) \leq V_\tau^\pi(s) \leq V_{\tau'}^\pi(s) + \frac{\tau - \tau'}{1-\gamma} \log|\mathcal{A}|. \tag{4.1.4}$$

Proof. By the definitions of V_τ^π and d_s^π, we have

$$V_\tau^\pi(s) = \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t \left(c(s_t, a_t) + h^\pi(s_t) + \tau D_{\pi_0}^\pi(s)\right)$$

$$|s_0 = s,\ a_t \sim \pi(\cdot|s_t),\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\big]$$

$$= \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t (c(s_t, a_t) + h^\pi(s_t) + \tau' D_{\pi_0}^\pi(s))$$

$$|s_0 = s,\ a_t \sim \pi(\cdot|s_t),\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\big]$$

$$+\mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t (\tau - \tau') D_{\pi_0}^\pi(s)|s_0 = s,\ a_t \sim \pi(\cdot|s_t),\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)\big]$$

$$= V_{\tau'}^\pi(s) + \frac{\tau - \tau'}{1-\gamma} \mathbb{E}_{s' \sim d_s^\pi}\left[D_{\pi_0}^\pi(s')\right],$$

which together with the bound on $D_{\pi_0}^\pi$, we get (4.1.4). □

As will be presented in Algorithm 3, the Approximate Policy Mirror Descent (APMD) method is derived by substituting $Q^{\pi_k}(s, \cdot)$ with its approximation $Q_{\tau_k}^{\pi_k}(s, \cdot)$ and incorporating the perturbation term $\tau_k D_{\pi_0}^\pi(s_t)$ when updating π_{k+1} in the basic Policy Mirror Descent (PMD) method. In fact, the APMD method serves as a generalized form of the PMD method since it reverts to the standard PMD approach when $\tau_k = 0$. Moreover, by flexibly selecting the convex function $h^\pi(s)$ and maintaining the strict convexity of the overall sum through a perturbation term that decreases over time, we can develop an accurate approximation approach. This methodology is illustrated by a key example examined in section (4.3).

Algorithm 3 The approximate policy mirror descent (APMD) method

Input: initial points π_0, stepsizes $\eta_k \geq 0$ and perturbation $\tau_k \geq 0$.

for $k = 0, 1, \ldots,$ **do**

$$\pi_{k+1}(\cdot|s) = \underset{p(\cdot|s) \in \Delta_{|\mathcal{A}|}}{\arg\min} \left\{\eta_k \left[\langle Q_{\tau_k}^{\pi_k}(s, \cdot), p(\cdot|s)\rangle + h^p(s) + \tau_k D_{\pi_0}^p(s_t)\right] + D_{\pi_k}^p(s)\right\}, \quad \forall s \in \mathcal{S}. \tag{4.1.5}$$

end for

In the remainder of this section, we observe that the Approximate Policy Mirror Descent method Lan [2022], when appropriately selecting τ_k, can achieve a linear convergence rate for solving general reinforcement learning problems. It is important to note that in the classic mirror descent method, introducing a perturbation term into the objective function typically does not enhance the convergence rate from sublinear to linear. However, in the Policy Mirror Descent (PMD) framework, the linear convergence rate is influenced

by the discount factor rather than the strong convexity parameter of the regularization term, as demonstrated in Theorem 1. This distinction enables us to establish a linear convergence rate for the APMD method.

4.2 APMD convergence analysis

In this section, we note that Lan [2022] extends and adapts the lemmas used in the convergence analysis of Policy Mirror Descent to the Approximate Policy Mirror Descent framework. First, we observe that Lemma 2.3 can still be applied to the perturbed value functions. The difference between the following result and Lemma 2.3 exists in that the RHS of (4.1.6) is no longer nonnegative, i.e., $V_\tau^\pi(s) - V_\tau^{\pi^\star}(s) \not\geq 0$. However, this relation will be approximately satisfied if τ is small enough.

Lemma 4.2 (Lan [2022], Lemma 7). The VI problem in (2.3.9) satisfies

$$\mathbb{E}_{s\sim\nu^\star}\left[\langle Q_\tau^\pi(s,\cdot), \pi(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^\pi(s) - h^{\pi^\star}(s) + \tau\left(D_{\pi_0}^\pi(s) - D_{\pi_0}^{\pi^\star}(s)\right)\right]$$

$$= \mathbb{E}_{s\sim\nu^\star}\left[(1-\gamma)\left(V_\tau^\pi(s) - V_\tau^{\pi^\star}(s)\right)\right]. \tag{4.2.1}$$

Proof. The proof is the same as that for Lemma 2.3 except that we will apply the performance difference lemma (i.e., Lemma 2.2) to the perturbed value function V_τ^π. □

Next, we establish some general convergence properties about the APMD method. Lemma (4.3) below will play the role of lemma (3.1) in convergence analysis.

Lemma 4.3 (Derivation of the Three-Point Lemma for AMPD, Lan [2022] Lemma 8). Let $\pi_{k+1}(\cdot|s)$ be defined in (4.1.5). For any $p(\cdot|s) \in \Delta_{|\mathcal{A}|}$, we have

$$\eta_k\left[\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - p(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^p(s)\right] + \eta_k\tau_k\left[D_{\pi_0}^p(s_t) - D_{\pi_0}^{\pi_{k+1}}(s_t)\right]$$

$$+D_{\pi_k}^p(s) - D_{\pi_k}^{\pi_{k+1}}(s) \leq D_{\pi_k}^p(s) - (1+\eta_k\tau_k)D_{\pi_{k+1}}^p(s).$$

Proof. The proof is similar to the proof of 4.3, by considering the function $\tilde{h}_\tau^p(s) = h^p(s) + \tau D_{\pi_0}^p(s)$, which is $\tau-$strongly convex with respect to KL-divergence as demonstrated in Example 1.1.
The lemma below is similar to the Descent property of PMD in lemma 3.2.

Lemma 4.4 (Lan [2022], Lemma 9). For any $s \in \mathcal{S}$, we have

$$\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi_k}(s)$$

$$+\tau_k \left[D_{\pi_0}^{\pi_{k+1}}(s) - D_{\pi_0}^{\pi_k}(s)\right] \geq V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s). \tag{4.2.2}$$

Proof. By applying Lemma 2.2 to the perturbed value function V_τ^π and using an argument similar to (3.2.3), we can show that

$$V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s)$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q_{\tau_k}^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle\right.$$

$$\left.+h^{\pi_{k+1}}(s') - h^{\pi_k}(s') + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s) - D_{\pi_0}^{\pi_k}(s)\right]\right]. \tag{4.2.3}$$

Now we conclude from Lemma 4.3 with $p(\cdot|s') = \pi_k(\cdot|s')$ that

$$\langle Q_{\tau_k}^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle + h^{\pi_{k+1}}(s') - h^{\pi_k}(s') + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s') - D_{\pi_0}^{\pi_k}(s')\right]$$

$$\leq -\frac{1}{\eta_k}\left[(1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi_k}(s') + D_{\pi_k}^{\pi_{k+1}}(s')\right], \tag{4.2.4}$$

which implies that

$$\mathbb{E}_{s'\sim d_s^{\pi_{k+1}}}\left[\langle Q_{\tau_k}^{\pi_k}(s',\cdot), \pi_{k+1}(\cdot|s') - \pi_k(\cdot|s')\rangle\right.$$

$$\left.+h^{\pi_{k+1}}(s') - h^{\pi_k}(s') + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s') - D_{\pi_0}^{\pi_k}(s')\right]\right]$$

$$\leq d_s^{\pi_{k+1}}(s)\left[\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right.$$

$$\left.+h^{\pi_{k+1}}(s) - h^{\pi_k}(s) + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s) - D_{\pi_0}^{\pi_k}(s)\right]\right]$$

$$\leq (1-\gamma)\left[\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right.$$

$$\left.+h^{\pi_{k+1}}(s) - h^{\pi_k}(s) + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s) - D_{\pi_0}^{\pi_k}(s)\right]\right], \tag{4.2.5}$$

where the last inequality follows from (2.2.14) that $d_s^{\pi_{k+1}}(s) \geq (1-\gamma)$ due to the definition of $d_s^{\pi_{k+1}}$ in (2.2.11). The result in (4.2.2) then follows immediately from (4.2.3) and the above inequality.

Lemma 4.5 (Lan [2022], Lemma 10).

Suppose $1 + \eta_k\tau_k = 1/\gamma$ and $\tau_k \geq \tau_{k+1}$ in the APMD method. Then for any $k \geq 0$, we have

$$\mathbb{E}_{s\sim\nu^\star}\left[V_{\tau_{k+1}}^{\pi_{k+1}}(s) - V_{\tau_{k+1}}^{\pi^\star}(s) + \frac{\tau_{k+1}}{1-\gamma}D_{\pi_{k+1}}^{\pi^\star}(s)\right]$$

$$\leq \mathbb{E}_{s\sim\nu^\star}\left[\gamma\left(V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma}D_{\pi_k}^{\pi^\star}(s)\right) + \frac{\tau_k - \tau_{k+1}}{1-\gamma}\log|\mathcal{A}|\right]. \tag{4.2.6}$$

Proof. By Lemma 4.3 with $p = \pi^\star$, we have

$$\eta_k\left[\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi^\star}(s)\right]$$
$$+\eta_k\tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s) - D_{\pi_0}^{\pi^\star}(s)\right] + D_{\pi_k}^{\pi_{k+1}}(s) \leq D_{\pi_k}^{\pi^\star}(s) - (1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi^\star}(s).$$

Moreover, by the previous lemma, we have

$$\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle + h^{\pi_{k+1}}(s) - h^{\pi^k}(s) + \tau_k\left[D_{\pi_0}^{\pi_{k+1}}(s_t) - D_{\pi_0}^{\pi_k}(s_t)\right]$$
$$\geq V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s).$$

Combining the above two relations, we obtain

$$\eta_k\left[\langle Q_{\tau_k}^{\pi_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\rangle + h^{\pi_k}(s) - h^{\pi^\star}(s)\right] + \eta_k\tau_k\left[D_{\pi_0}^{\pi_k}(s_t) - D_{\pi_0}^{\pi^\star}(s_t)\right]$$
$$+\eta_k\left[V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s)\right] + D_{\pi_k}^{\pi_{k+1}}(s) \leq D_{\pi_k}^{\pi^\star}(s) - (1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi^\star}(s).$$

Taking expectation w.r.t. $\nu^\star$ on both sides of the above inequality and using Lemma 4.2, we arrive at

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta_k(1-\gamma)(V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s))\right] + \eta_k\mathbb{E}_{s\sim\nu^\star}\left[V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s)\right]$$
$$+\mathbb{E}_{s\sim\nu^\star}\left[D_{\pi_k}^{\pi_{k+1}}(s)\right] \leq \mathbb{E}_{s\sim\nu^\star}\left[D_{\pi_k}^{\pi^\star}(s) - (1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi^\star}(s)\right].$$

Noting $V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi_k}(s) = V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi^\star}(s) - [V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s)]$ and rearranging the terms in the above inequality, we have

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta_k(V_{\tau_k}^{\pi_{k+1}}(s) - V_{\tau_k}^{\pi^\star}(s)) + (1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi^\star}(s) + D_{\pi_k}^{\pi_k+1}(s)\right]$$
$$\leq \eta_k\gamma\mathbb{E}_{s\sim\nu^\star}\left[V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s)\right] + \mathbb{E}_{s\sim\nu^\star}\left[D_{\pi_k}^{\pi^\star}(s)\right]. \tag{4.2.7}$$

Using the above inequality, the assumption $\tau_k \geq \tau_{k+1}$ and (3.19), we have

$$\mathbb{E}_{s\sim\nu^\star}\left[\eta_k(V_{\tau_{k+1}}^{\pi_{k+1}}(s) - V_{\tau_{k+1}}^{\pi^\star}(s)) + (1+\eta_k\tau_k)D_{\pi_{k+1}}^{\pi^\star}(s) + D_{\pi_k}^{\pi_{k+1}}(s)\right]$$
$$\leq \mathbb{E}_{s\sim\nu^\star}\left[\eta_k\gamma(V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s)) + D_{\pi_k}^{\pi^\star}(s)\right] + \frac{\eta_k(\tau_k - \tau_{k+1})}{1-\gamma}\log|\mathcal{A}|, \tag{4.2.8}$$

which implies the result by the assumption $1 + \eta_k\tau_k = 1/\gamma$. □

In the following theorem, we analyze the convergence rate of the APMD method by employing dynamic step size rules for selecting η_k and τ_k when solving general reinforcement learning (RL) problems.

Theorem 4 (Lan [2022], Theorem 3). Suppose that $\tau_k = \tau_0\gamma^k$ for some $\tau_0 \geq 0$ and that $1 + \eta_k\tau_k = 1/\gamma$ for any $k \geq 0$ in the APMD method. Then for any $k \geq 0$, we have

$$f(\pi_k) - f(\pi^\star) \leq \gamma^k \left[f(\pi_0) - f(\pi^\star) + \tau_0 \left(\frac{2}{1-\gamma} + \frac{k}{\gamma} \right) \log |\mathcal{A}| \right]. \tag{4.2.8}$$

Proof. Applying the result in the previous lemma recursively, we have

$$\mathbb{E}_{s\sim\nu^\star}\left[V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s)\right] \leq \gamma^k \mathbb{E}_{s\sim\nu^\star}\left[V_{\tau_0}^{\pi_0}(s) - V_{\tau_0}^{\pi^\star}(s) + \frac{\tau_0}{1-\gamma} D_{\pi_0}^{\pi^\star}(s)\right].$$

$$+ \sum_{i=1}^{k} \frac{(\tau_{i-1} - \tau_i)\gamma^{k-i}}{1-\gamma} \log |\mathcal{A}|.$$

Noting that $V_{\tau_k}^{\pi_k}(s) \geq V^{\pi_k}(s), V_{\tau_k}^{\pi^\star}(s) \leq V^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma}\log|\mathcal{A}|$, and $V_{\tau_0}^{\pi^\star}(s) \geq V^{\pi^\star}(s)$ due to (4.1.3), and that $V_{\tau_0}^{\pi_0}(s) = V^{\pi_0}(s)$ due to $D_{\pi_0}^{\pi_0}(s) = 0$, we conclude from the previous inequality that

$$\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_k}(s) - V^{\pi^\star}(s)\right] \leq \gamma^k \mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_0}(s) - V^{\pi^\star}(s) + \frac{\tau_0}{1-\gamma} D_{\pi_0}^{\pi^\star}(s)\right]$$

$$+ \left[\frac{\tau_k}{1-\gamma} + \sum_{i=1}^{k} \frac{(\tau_{i-1} - \tau_i)\gamma^{k-i}}{1-\gamma}\right] \log |\mathcal{A}|. \tag{4.2.9}$$

The result in (4.2.8) immediately follows from the above relation, the definition of f in (2.3.6), and the selection of τ_k. □

According to (4.2.8), if τ_0 is a constant, then the rate of convergence of the APMD method is $\mathcal{O}(k\gamma^k)$. If the total number of iterations k is given a priori, one can improve the rate of convergence to $\mathcal{O}(\gamma^k)$ by setting $\tau_0 = 1/k$. Below is a different way to specify τ_k for the APMD method so that it can achieve this $\mathcal{O}(\gamma^k)$ rate of convergence without fixing k a prior. We first review a technical result that will also be used later for the analysis of stochastic PMD methods.

Lemma 4.6 (Lan [2022], Lemma 11). Assume that the nonnegative sequences $\{X_k\}_{k\geq 0}$,

$\{Y_k\}_{k\geq 0}$, and $\{Z_k\}_{k\geq 0}$ satisfy

$$X_{k+1} \leq \gamma X_k + (Y_k - Y_{k+1}) + Z_k. \tag{4.2.10}$$

Let us denote $l = \left\lceil \log_\gamma \frac{1}{4} \right\rceil$. *If* $Y_k = Y \cdot 2^{-(k/l+1)}$ *and* $Z_k = Z \cdot 2^{-(k/l+2)}$ *for some* $Y \geq 0$ *and* $Z \geq 0$, *then*

$$X_k \leq 2^{-\lfloor k/l \rfloor} \left(X_0 + Y + \frac{5Z}{4(1-\gamma)} \right). \tag{4.2.11}$$

Proof. Lemma 11, Lan [2022]. □

Theorem 5 (Lan [2022], Theorem 4). Let us denote $l := \left\lceil \log_\gamma \frac{1}{4} \right\rceil$. If $\tau_k = 2^{-(\lfloor k/l \rfloor + 1)}$ and $1 + \eta_k \tau_k = 1/\gamma$, then

$$f(\pi_k) - f(\pi^\star) \leq 2^{-\lfloor k/l \rfloor} \left[f(\pi_0) - f(\pi^\star) + \frac{2 \log |\mathcal{A}|}{1-\gamma} \right].$$

Proof. By using Lemma 4.5 and Lemma 4.6 (with $X_k = \mathbb{E}_{s\sim\nu^\star}[V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma} D_{\pi_k}^{\pi^\star}(s)]$ and $Y_k = \frac{\tau_k}{1-\gamma} \log |\mathcal{A}|$), we have

$$\mathbb{E}_{s\sim\nu^\star} \left[V_{\tau_k}^{\pi_k}(s) - V_{\tau_k}^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma} D_{\pi_k}^{\pi^\star}(s) \right]$$

$$\leq 2^{-\lfloor k/l \rfloor} \left\{ \mathbb{E}_{s\sim\nu^\star} \left[V_{\tau_0}^{\pi_k}(s) - V_{\tau_0}^{\pi^\star}(s) + \frac{\tau_0}{1-\gamma} D_{\pi_0}^{\pi^\star}(s) \right] + \frac{\log |\mathcal{A}|}{1-\gamma} \right\}.$$

Noting that $V_{\tau_k}^{\pi_k}(s) \geq V^{\pi_k}(s), V_{\tau_k}^{\pi^\star}(s) \leq V^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma} \log |\mathcal{A}|, V_{\tau_0}^{\pi^\star}(s) \geq V^{\pi^\star}(s)$ due to (4.1.3), and that $V_{\tau_0}^{\pi_0}(s) = V^{\pi_0}(s)$ due to $D_{\pi_0}^{\pi_0}(s) = 0$, we conclude from the previous inequality and the definition of τ_k that

$$\mathbb{E}_{s\sim\nu^\star} \left[V^{\pi_k}(s) - V^{\pi^\star}(s) + \frac{\tau_k}{1-\gamma} D_{\pi_k}^{\pi^\star}(s) \right]$$

$$\leq 2^{-\lfloor k/l \rfloor} \left\{ \mathbb{E}_{s\sim\nu^\star} \left[V^{\pi_k}(s) - V_{\tau_0}^{\pi^\star}(s) + \frac{\tau_0}{1-\gamma} D_{\pi_0}^{\pi^\star}(s) \right] + \frac{\log |\mathcal{A}|}{1-\gamma} \right\} + \frac{\tau_k \log |\mathcal{A}|}{1-\gamma}$$

$$\leq 2^{-\lfloor k/l \rfloor} \left\{ \mathbb{E}_{s\sim\nu^\star} \left[V^{\pi_0}(s) - V_{\tau_0}^{\pi^\star}(s) \right] + \frac{2 \log |\mathcal{A}|}{1-\gamma} \right\}.$$

□

In accordance with Theorem 5, a policy $\tilde{\pi}$ such that $f(\tilde{\pi}) - f(\pi^\star) \leq \epsilon$ will be identified within a maximum of $\mathcal{O}(\log(1/\epsilon))$ epochs, which corresponds to no more than $\mathcal{O}(l \log(1/\epsilon)) = \mathcal{O}(\log_\gamma(\epsilon))$ iterations. This convergence rate is comparable to that achieved in reinforcement learning problems that utilize strongly convex regularizers. However, for general RL problems, ensuring a linear convergence rate for $D_{\pi_{k+1}}^{\pi^\star}(s)$ is guaranteed be-

cause the coefficient τ_k eventually becomes very small. By considering the continuity of the objective function and the compactness of the feasible set, it is possible to demonstrate that the sequence of solutions asymptotically approaches the true optimal policy as the number of iterations grows. On the other hand, establishing a convergence rate for the solution sequence of the Policy Mirror Descent method in general RL problems remains unattainable without further exploration of the structural properties inherent to these problems (Lan [2022]). It is noteworthy that a thorough analysis of a derivation of APMD has been done by Li et al. [2022] under the title of Homotopic Policy Mirror Descent (HPMD). They show that local superlinear convergence is obtainable without any further assumptions. Also, they extend their convergence results to HPMD instantiated with a broad class of decomposable Bregman divergences, demonstrating the generality of the computational properties they demonstrated.

4.3 Entropy-regularized PMD

Given the optimization problem

$$\pi_{k+1}(\cdot|s) = \underset{p\in\Delta_{|\mathcal{A}|}}{\arg\min} \left\{\eta\left[\langle Q^{\pi_k}(s,\cdot), p\rangle + h^p(s) + \tau D^p_{\pi_0}(s)\right] + D^p_{\pi_k}(s)\right\}, \quad \forall s \in \mathcal{S}, \quad (4.3.1)$$

If we consider $h^p(s)$ to be the zero function which is indeed convex, and minding the fact that the term $\tau D^p_{\pi_0}(s)$ in the optimization is equivalent to negative entropy function weighted by τ added to a constant which can be dropped, for all actions a at any state s, the objective function in (4.3.1) becomes

$$\eta\left[\langle Q^{\pi_k}(s,a), p\rangle + \tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log p(a_i \mid s)\right] + D^p_{\pi_k}(s)$$

$$= \eta\langle Q^{\pi_k}(s,a), p\rangle + \eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log p(a_i \mid s) + \sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log\frac{p(a_i \mid s)}{\pi_k(a_i \mid s)}$$

$$= \eta\langle Q^{\pi_k}(s,a), p\rangle + \eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log p(a_i \mid s) + \sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log\frac{p(a_i \mid s)}{\pi_k(a_i \mid s)}$$

$$+\eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log\pi_k(a_i \mid s) - \eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log\pi_k(a_i \mid s)$$

$$= \eta\langle Q^{\pi_k}(s,a), p\rangle + \eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log\pi_k(a_i \mid s)$$

$$+(1+\eta\tau)(\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log p(a_i \mid s) - (1+\eta\tau)\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log \pi_k(a_i \mid s)$$

$$= \eta\langle Q^{\pi_k}(s,a), p\rangle + \eta\tau\sum_{i=1}^{|\mathcal{A}|} p(a_i \mid s)\log \pi_k(a_i \mid s) + (1+\eta\tau)D^p_{\pi_k}(s)$$

$$= \frac{\eta}{1+\eta\tau}\langle Q^{\pi_k}(s,a) + \tau\log(\pi_k), p\rangle + D^p_{\pi_k}(s)$$

The last equation is followed by multiplication by $\frac{1}{1+\eta\tau}$, which doesn't have an effect on the whole optimization. By (1.3.4) in the entropy descent algorithm setup discussed in Example 1.2, we obtain explicit policy updates for each action a at state s as follows.

$$\pi_{k+1}(a|s) = \frac{\pi_k(a|s)\exp(-\eta\frac{Q^{\pi_k}(s,a)+\tau\log(\pi_k(a|s)}{1+\eta\tau})}{\sum_{a'\in\mathcal{A}}\pi_k(a'|s)\exp(-\eta\frac{Q^{\pi_k}(s,a')+\tau\log(\pi_k(a'|s)}{1+\eta\tau})}$$

$$= \frac{\pi_k(a|s)^{(\frac{1}{1+\eta\tau})}\exp(\frac{-\eta Q^{\pi_k}(s,a)}{1+\eta\tau})}{\sum_{a'\in\mathcal{A}}\pi_k(a'|s)^{(\frac{1}{1+\eta\tau})}\exp(\frac{-\eta Q^{\pi_k}(s,a')}{1+\eta\tau})} \tag{4.3.2}$$

Which results in a closed-form update for PMD with the entropy function regularizer with a strong convexity modulus of τ. If we set τ dynamically in each iteration according to the APMD assumption on stepsize and perturbation coefficient term, It corresponds to the case of APMD with a constant convex function plus a scaled entropy function as a perturbation term, where we can drop the constant function in the optimization, keeping only an exponentially diminishing strongly convex perturbation term. This perturbation term, which serves as a strongly convex regularizer in each iteration, will enable us to use the convergence guarantees, while as iterations go by, it converges to zero exponentially fast and leads the perturbed problem to the original non-regularized problem.

In the upcoming chapter, we provide a concise overview of the Entropy-regularized Natural Policy Gradient framework as examined in Cen et al. [2022]. This framework has been proven to achieve a linear convergence rate, and their results accommodate a wide range of learning rates and shed light on the role of entropy regularization in enabling fast convergence.

Chapter 5

Entropy-regularized Natural Policy Gradient

The Natural Policy Gradient algorithm was introduced by Kakade [2001], which is a direct application of the natural gradient method (Amari [1998]) for RL. NPG methods are among the most widely utilized policy optimization algorithms in contemporary reinforcement learning (Schulman et al. [2015], Bhatnagar et al. [2007], Schulman et al. [2017], Agarwal et al. [2021]). Entropy-regularized NPG uses the Fisher information matrix (which is the Hessians of a certain divergence metric) for pre-conditioning the gradient steps and resembles a quasi-Newton method (Cayci et al. [2024a]). Entropy-regularized NPG can also be thought of as a variant of mirror descent (Nemirovski et al. [2009], Geist et al. [2019], Shani et al. [2020]) and as a smoother approximation of Policy iteration (Khodadadian et al. [2021]). This class of methods is frequently employed alongside entropy regularization, which provides stability and sufficient exploration in the optimization process (Cayci et al. [2024b], Ahmed et al. [2019]). In this chapter, the primary reference is "Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization" by Cen et al. [2022]. We review non-asymptotic convergence guarantees for Entropy-regularized NPG methods under softmax parameterization on discounted Markov Decision Processes and consider the problem of maximizing the cumulative reward, demonstrated in the mentioned paper. Therefore, instead of cost function $c(s, a)$, we deal with reward function $r(s, a)$ and consider maximizing a concave objective.

5.1 Policy parameterization

Policy parameterization refers to the method of representing and defining policies in reinforcement learning using parameterized functions. Instead of maintaining explicit map-

pings for each state-action pair, policies are expressed as functions that map states to actions (deterministic policies) or probability distributions over actions (stochastic policies) based on a set of parameters. For instance, deep learning has revolutionized reinforcement learning by providing sophisticated function approximators capable of capturing complex, high-dimensional relationships between states and actions, and more generally, it can serve as a powerful tool for parameterizing various critical components of an agent's decision-making process, enabling the development of complex and scalable reinforcement learning algorithms (Mnih et al. [2013], Peters and Schaal [2008]). By using neural networks, policies and action-value functions can be represented as parameterized functions, $\pi_\theta(a \mid s)$ and $Q(s, a; \theta)$ respectively, where θ stands for the network weights and biases (Agarwal et al. [2021]). For example, in algorithms like Deep Q-Networks (DQN, Mnih et al. [2015]) and Proximal Policy Optimization (PPO, Schulman et al. [2017]), convolutional and fully connected layers are employed to process raw input data, such as images, allowing the agent to learn effective policies directly from high-dimensional sensory inputs (Mnih et al. [2013]). This integration of deep learning not only enhances the agent's ability to learn robust and adaptive policies but also extends the applicability of reinforcement learning to complex real-world problems (Arulkumaran et al. [2017]).

A simple and widely adopted technique in reinforcement learning is softmax parametrization. It transforms preference scores for each action within a given state into a probability distribution over actions, ensuring that the probabilities are non-negative and sum to one. This parametrization is particularly advantageous for facilitating gradient-based optimization techniques and maintaining the parameters within the domain, as the softmax function inherently makes an ℓ^1 projection onto the probability simplex. For consistency, we adopt the notation and concepts of softmax parametrization as presented in Cen et al. [2022] in the subsequent sections

Formally, given a state $s \in \mathcal{S}$, function θ, and a set of actions $\mathcal{A}$, the softmax policy assigns a probability to each action $a \in A$ as follows:

$$\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

$$\pi_\theta(a \mid s) = \frac{\exp\left(\theta(s, a)\right)}{\sum_{a' \in A} \exp\left(\theta(s, a')\right)} \tag{5.1.1}$$

where:

- $\theta(s, a)$ represents the preference parameter (or logic values) for action a in state s. These preference parameters are typically the outputs of a function approximator, which takes the state s as input and produces preference scores for each possible action a in that state.

- The exponential function ensures that the probabilities are positive.
- The denominator normalizes the preferences to sum to one, yielding a valid probability distribution over action space.

Although softmax parametrization can be implemented within the tabular framework by assigning distinct preference parameters to each state-action pair, its true advantages are realized within the approximate framework (Cayci et al. [2024a], Agarwal et al. [2021]). In the tabular setting, policies are explicitly stored and managed, which limits the effectiveness of softmax in facilitating parameter sharing and generalization. Conversely, in the approximate framework, softmax parametrization enables efficient policy representation and optimization, making it well-suited for complex and high-dimensional environments.

In the following section, by slightly abuse of notation π_θ and θ are treated as vectors in $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, and suppress the subscription θ from π_θ, whenever it is clear from the context.

5.2 Entropy-regularized NPG setting

Entropy-regularized value maximization. To promote exploration and discourage premature convergence to suboptimal policies, a widely used strategy is entropy regularization, which searches for a policy that maximizes the following entropy-regularized value function

$$\mathbb{E}_{s\sim\rho} V_\tau^\pi(s) := \mathbb{E}_{s\sim\rho} V^\pi(s) + \tau \cdot \mathcal{H}(\rho, \pi). \tag{5.2.1}$$

we denote $\mathbb{E}_{s\sim\rho} V_\tau^\pi(s)$ by $V_\tau^\pi(\rho)$, and the quantity $\tau \geq 0$ stands for the regularization parameter, and $\mathcal{H}(\rho, \pi)$ denotes a sort of *discounted entropy* defined as follows

$$\mathcal{H}(\rho, \pi) := \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi(a_t|s_t)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\pi}\left[\sum_{a\in\mathcal{A}} \pi(a|s) \log \frac{1}{\pi(a|s)}\right]. \tag{5.2.2}$$

Equivalently, V_τ^π(s) can be viewed as the value function of π by adjusting the instantaneous reward to be a policy-dependent regularized version as follows

$$\forall (s,a) \in \mathcal{S}\times\mathcal{A}: \quad r_\tau(s,a) := r(s,a) - \tau \log \pi(a|s). \tag{5.2.3}$$

$V_\tau^\pi(s)$ is also defined analogously when the initial state is fixed to be any given state $s \in \mathcal{S}$. The regularized Q-function Q_τ^π of a policy π, also known as the soft Q-function, is related to V_τ^π as

$$\forall (s,a) \in \mathcal{S}\times\mathcal{A}: \quad Q_\tau^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[V_\tau^\pi(s')\right], \tag{5.2.4}$$

$$\forall s \in \mathcal{S}: \quad V_\tau^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[-\tau \log \pi(a|s) + Q_\tau^\pi(s,a)\right]. \tag{5.2.5}$$

Natural policy gradient methods. To compute the optimal policy in a parameterized form, a common approach is to perform gradient ascent with respect to the parameter θ until convergence. This first-order method is known as the policy gradient (PG) algorithm (e.g., Sutton et al. [1999]; Sutton and Barto [2020], Chapter 13). In comparison, the natural policy gradient method (Kakade [2001]) applies a pre-conditioned gradient update rule:

$$\theta \leftarrow \theta + \eta(\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta}(\rho), \tag{5.2.6}$$

with the aim of searching along a direction independent of the specific parameterization of the policy. Here, η represents the learning rate or step size, and $\mathcal{F}_\rho^\theta$ denotes the Fisher information matrix, defined by

$$\mathcal{F}_\rho^\theta := \mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\left(\nabla_\theta \log \pi_\theta(a|s)\right)\left(\nabla_\theta \log \pi_\theta(a|s)\right)^\top\right], \tag{5.2.7}$$

And the dagger sign stands for Moore Penrose pseudoinverse of a matrix. It is understood that the Entropy-regularized NPG method primarily aims to regulate or control policy changes approximately in terms of the Kullback-Leibler (KL) divergence (see, e.g., Schulman et al. [2015], Section 7).

NPG methods with entropy regularization. With entropy regularization, the Entropy-regularized NPG update rule can be expressed as

$$\theta \leftarrow \theta + \eta(\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V_\tau^{\pi_\theta}(\rho), \tag{5.2.6}$$

which employs the same update rule in (5.2.6) where $\mathcal{F}_\rho^\theta$ is defined in (5.2.7) and $V_\tau^{\pi_\theta}(\rho)$ the expectation of the regularized value function defined in (5.2.5) with respect to some distribution ρ over the states. Under a softmax parameterization as in (5.1.1), this update rule takes a relatively simple form in the policy space (Cen et al. [2022], Appendix A.1) and, is invariant to the choice of ρ. Specifically, if we let $\theta^{(t)}$ denote the t-th iterate and $\pi^{(t)} = \mathrm{softmax}(\theta^{(t)})$ represent the associated policy, then the Entropy-regularized NPG updates satisfy

$$\pi^{(t+1)}(a|s) = \frac{1}{Z^{(t)}(s)}\left(\pi^{(t)}(a|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta Q_\tau^{\pi^{(t)}}(s,a)}{1-\gamma}\right), \tag{5.2.7}$$

where $Q_\tau^{\pi^{(t)}}$ is the soft Q-function of policy $\pi^{(t)}$, and $Z^{(t)}(s)$ is a normalization factor. This update can alternatively be seen as a variant of the trust region policy optimization (TRPO) algorithm (Schulman et al. [2015]); Shani et al. [2020]). As a notable special

case, the update rule in (5.2.7) simplifies to

$$\pi^{(t+1)}(\cdot|s) = \frac{1}{Z^{(t)}(s)} \exp\left(\frac{Q_\tau^{\pi^{(t)}}(s,\cdot)}{\tau}\right) \quad \text{when } \eta = \frac{1-\gamma}{\tau}.$$

It can be interpreted as a "soft" version of the classical policy iteration algorithm (Cen et al. [2022]) To simplify the notation, $V_\tau^{\pi^{(t)}}, Q_\tau^{\pi^{(t)}}, d_\rho^{(t)}$ are denoted as $V_\tau^{(t)}, Q_\tau^{(t)}, d_\rho^{\pi^{(t)}}$, respectively. The complete Entropy-regularized natural policy gradient is summarized as follows

Algorithm 4 Entropy-regularized NPG with exact policy evaluation

1 inputs: learning rate η, initialization $\pi^{(0)}$.
2 for $t = 0, 1, 2, \ldots$ **do**
3 Compute the regularized Q-function $Q_\tau^{(t)}$ (defined in (5.2.4)) of policy $\pi^{(t)}$.
4 Update the policy:

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad \pi^{(t+1)}(a|s) = \frac{1}{Z^{(t)}(s)} \left(\pi^{(t)}(a|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta Q_\tau^{(t)}(s,a)}{1-\gamma}\right), \tag{5.2.8}$$

where $Z^{(t)}(s) = \sum_{a' \in \mathcal{A}} \left(\pi^{(t)}(a'|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta Q_\tau^{(t)}(s,a')}{1-\gamma}\right)$.

The following theorem establishes the linear convergence rate of the exact entropy-regularized Natural Policy Gradient method. This result demonstrates that, under specific conditions on the learning rate, the Entropy-regularized NPG updates converge to their optimal values at a linear rate. Such convergence guarantees that the difference between the updated Q-function and the optimal Q-function, as well as the difference between the updated policy log-probabilities and the optimal log-probabilities, diminishes exponentially over iterations. This provides a strong theoretical foundation for the efficiency of Entropy-regularized NPG in finding optimal policies within a reliable time frame.

Theorem 6 (Cen et al. [2022], Theorem 1). For any learning rate $0 < \eta \leq \frac{1-\gamma}{\tau}$, the Entropy-regularized NPG updates (5.2.8) satisfy

$$\left\|Q_\tau^\star - Q_\tau^{(t+1)}\right\|_\infty \leq C_1 \gamma (1-\eta\tau)^t \tag{5.2.9}$$

$$\left\|\log \pi_\tau^\star - \log \pi_\tau^{(t+1)}\right\|_\infty \leq 2C_1 \tau^{-1} (1-\eta\tau)^t \tag{5.2.10}$$

$$\left\|V_\tau^\star - V_\tau^{(t+1)}\right\|_\infty \leq 3C_1 \gamma (1-\eta\tau)^t \tag{5.2.11}$$

for all $t \geq 0$, where

$$C_1 := \left\|Q_\tau^\star - Q_\tau^{(0)}\right\|_\infty + 2\tau \left(1 - \frac{\eta\tau}{1-\gamma}\right) \left\|\log \pi_\tau^\star - \log \pi_\tau^{(0)}\right\|_\infty. \tag{5.2.11}$$

Proof. Theorem 1, Cen et al. [2022]. □

According to (5.2.9), to achieve $\left\|Q_\tau^\star - Q_\tau^{(t)}\right\|_\infty \leq \epsilon$, the entropy-regularized Entropy-regularized NPG method requires at most

$$\frac{1}{\eta\tau}\log\left(\frac{C_1\gamma}{\epsilon}\right)$$

iterations. Also, the iteration complexity is largely independent of the dimensions of the Markov Decision Process (MDP), with only a minimal dependence embedded in $\log(C_1)$. Moreover, unlike the unregularized case, entropy regularization ensures the uniqueness of the optimal policy, allowing for a direct analysis of policy convergence. Theorem 6 shows that the Entropy-regularized NPG method requires at most

$$\frac{1}{\eta\tau}\log\left(\frac{2C_1}{\epsilon\tau}\right)$$

iterations to achieve $\left\|\log\pi_\tau^\star - \log\pi_\tau^{(t+1)}\right\|_\infty \leq \epsilon$.

To summarize, the linear convergence guarantees established by Cen et al. [2022] underscore the effectiveness of the entropy-regularized Natural Policy Gradient (Entropy-regularized NPG) method. By ensuring the uniqueness of optimal policies and providing iteration bounds that are largely independent of the dimensions of the MDP, entropy regularization not only stabilizes the learning process but also accelerates convergence. These results offer a robust theoretical foundation for employing Entropy-regularized NPG in complex reinforcement learning tasks. Moreover, the minimal dimension-dependent iteration bounds suggest that this approach scales favorably to high-dimensional problems, making it a promising candidate for real-world applications where traditional methods may struggle due to computational limitations.

5.3 Entropy-regularized NPG as an instance of PMD

As mentioned earlier, the Policy Mirror Descent framework is designed to minimize cumulative costs, whereas the Entropy-regularized Natural Policy Gradient method focuses on maximizing cumulative rewards. According to Section 7.3 of Puterman [1994], to address the minimization of the cumulative cost objective within the PMD framework, one can construct a corresponding negative Markov Decision Process by negating the cost function, thereby deriving an equivalent reward function. This transformation effectively converts the cost minimization problem into a reward maximization problem, providing the necessary environment for applying the Entropy-regularized Natural Policy Gradient

method. In this section, we establish a connection between the Policy Mirror Descent method and Entropy-regularized Natural Policy Gradient by demonstrating that, under a specific choice of step size, the update rules for PMD align with those of Entropy-regularized NPG. This result reveals an underlying relation between the two methods, suggesting that PMD can be viewed as a generalized framework that includes Entropy-regularized NPG as a special case. By carefully selecting the step size parameter, we show that the policy updates in PMD replicate the behavior of Entropy-regularized NPG, effectively unifying these two approaches within a common theoretical framework. This alignment offers insights into the structure of policy optimization methods, highlighting the adaptability of PMD in capturing the natural gradient dynamics when parameterized appropriately.

As discussed in (4.3.2), PMD yields a closed-form update rule for negative entropy function scaled by τ as the $\tau-$strongly convex function as follows

$$\pi_{k+1}(a \mid s) = \frac{\pi_k(a|s)^{(\frac{1}{1+\eta\tau})} \exp(\frac{-\eta Q^{\pi_k}(s,a)}{1+\eta\tau})}{\sum_{a' \in \mathcal{A}} \pi_k(a'|s)^{(\frac{1}{1+\eta\tau})} \exp(\frac{-\eta Q^{\pi_k}(s,a')}{1+\eta\tau})}, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}.$$

APMD stepsize assumption of $1 + \eta_k\tau_k \geq 1/\gamma$, with $\eta_k\tau_k = c$ for some constant c, for all iteration k, yields the corresponding APMD setting update rule as well. Now, we suggest to set hyperparameters of Entropy-regularized NPG as

$$\eta'_k = \frac{\eta_k(1-\gamma)}{1+c} \tag{5.3.1}$$

$$\tau'_k = \tau_k. \tag{5.3.2}$$

Trivially, for all k, we have

$$\frac{\eta_k\tau_k}{1+\eta_k\tau_k} \leq 1 \Rightarrow \frac{\eta_k}{1+\eta_k\tau_k} \leq \frac{1}{\tau_k}$$

$$\Rightarrow \frac{\eta_k(1-\gamma)}{1+c} \leq \frac{1-\gamma}{\tau_k},$$

This relationship ensures that the step size conditions required by Entropy-regularized NPG can be satisfied within the APMD framework by transformation (5.3.1-2). Consequently, it becomes feasible to unify the dynamic step sizes employed in APMD and Entropy-regularized NPG across iterations if such synchronization is desired. In alignment with this approach, we adopt a fixed step size within APMD. This choice effectively aligns APMD with the corresponding PMD setup, ensuring consistency with the established methodologies presented by Cen et al. [2022]. Considering parameters τ and η for PMD, and the corresponding stepsizes of η' and τ' for Entropy-regularized NPG men-

tioned in (5.3.1-2), then, the numerator in (5.2.8) will become

$$\left(\pi^{(k)}(a|s)\right)^{1-\frac{\eta'\tau'}{1-\gamma}} \exp\left(\frac{\eta' Q_{\tau'}^{(k)}(s,a)}{1-\gamma}\right)$$

$$= \left(\pi^{(k)}(a|s)\right)^{1-\tau\frac{\eta(1-\gamma)}{1+c}} \exp\left(\frac{\eta(1-\gamma)}{(1+c)(1-\gamma)} Q_{\tau}^{(k)}(s,a)\right)$$

$$= \left(\pi^{(k)}(a|s)\right)^{1-\frac{c}{1+c}} \exp\left(\frac{\eta}{(1+c)} Q_{\tau}^{(k)}(s,a)\right)$$

$$= \left(\pi^{(k)}(a|s)\right)^{\frac{1}{1+\eta\tau}} \exp\left(\frac{\eta}{(1+\eta\tau)} Q_{\tau}^{(k)}(s,a)\right). \tag{5.3.3}$$

Entropy-regularized NPG with this choice of hyperparameter will boil down to (5.3.3) together with its normalization factor, which will be equivalent to the PMD update rule in (4.3.2) with $\tau D_{\pi_0}^{\pi_k}(s)$ as its $\tau-$strongly convex regularizer. Therefore, both Entropy-regularized NPG and PMD utilize the same underlying update formula. However, the implementation of action-value function regularization in PMD (2.3.3) differs from the soft-regularization introduced in Equation (5.2.4). A careful evaluation reveals that this difference does not affect the updated policies in this setting. In Equation (2.3.3), we add $h^{\pi_k}(s)$ to the initial cost of the action-value pair (s,a), which is constant across different actions taken at state s. Consequently, a constant factor of

$$\exp\left(-\frac{\eta}{1+\eta\tau} h^{\pi_k}(s)\right)$$

cancels out from both the numerator and the denominator. On the other hand, as mentioned in (5.2.3), one can equivalently consider the reward function of

$$r_\tau(s,a) := r(s,a) - \tau \log \pi(a|s)$$

for the value function (Cen et al. [2022]). Similar to the equation (2.2.4) and by definition of entropy function, we have

$$V_\tau^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[-\tau \log \pi(a|s) + Q_\tau^\pi(s,a)\right]$$

$$= \langle -\tau \log \pi(a|s) + Q_\tau^\pi(s,a), \pi(\cdot|s)\rangle$$

$$= \langle \tau H(\pi(\cdot|s)) + Q_\tau^\pi(s,a), \pi(\cdot|s)\rangle$$

The final inequality arises from the fact that $\sum_{a\in\mathcal{A}} \pi(a|s) = 1$. According to Equation (2.2.4), this relationship defines the regularized value function in PMD derived from the

regularized Q-function in Equation (2.3.3) by setting

$$h^{\pi}(s) = -\tau H\left(\pi(\cdot|s)\right),$$

Then, the value function in both cases coincides, hence the optimization objective given that a similar distribution over the states is used to evaluate the expectation of value functions. Additionally, as illustrated in Example (2.1), $h^{\pi}(s)$ differs from $\tau D_{\pi_0}^{\pi_k}$—where π_0 is the uniform distribution—by a constant factor that can be factorized and canceled out in Equation (4.3.2). Hence, despite the slight differences in the definitions of the regularized Q-function in Entropy-regularized NPG and PMD, this does not affect the unified closed-form update rule discussed earlier, and the Q-function can be calculated either way.

Moreover, Cen et al. [2022] maintained a fixed stepsize η and regularizer coefficient τ in all iterations. Motivated by the dynamic stepsize selection in APMD, which reduces the influence and potential bias of the regularizer in subsequent iterations, our analysis demonstrated that the transformed dynamic stepsize choices in equations (5.3.1) and (5.3.2) align with the hyperparameter assumptions of Entropy-regularized NPG, and one can implement a dynamic stepsize strategy at each iteration similar to APMD.

Increasing step size and the connection with policy iteration. As for the use of the increasing step size, intuitively, APMD (and indeed entropy regularized Entropy-regularized NPG) behave more and more like policy iteration (Alfano et al. [2024], Xiao [2022]), which enjoys the convergence rate of at least linear to an optimal policy (Puterman [1994], Theorem 6.4.8). For instance, in PMD (and similarly APMD), when $\eta_k \to \infty$, the KL-divergence term in (3.1.1.), $D_{\pi_k}^{p}(s)$, will be dominated (the perturbation term as well in APMD) and results in the following optimization problem

$$\pi_{(k+1)}(\cdot|s) = \underset{p\in\Delta_{|\mathcal{A}|}}{\arg\min}\left\{\langle Q^{\pi_k}(s,\cdot), p\rangle + h^{p}(s)\right\}, \quad \forall s \in \mathcal{S},$$

which is the Policy Iteration method with regularization term $h^{p}(s)$, and having $h^{p}(s) \equiv 0$ leads precisely to the classical Policy Iteration (for more discussion on the connection with policy iteration, see Xiao [2022], Section 4.4; Puterman [1994]; Bertsekas [2024]). Also according to Agarwal et al. [2021], Lemma 5.1., Entropy-regularized NPG under softmax parametrization behaves as soft policy iteration (Haarnoja et al. [2018]). Roughly speaking, although the optimal policy is not unique, as stepsize increases, one can expect to achieve the same optimization solution as Policy Iteration obtains, as well as the convergence rate.

Chapter 6

Numerical experiments

Introduction

To evaluate the impact of varying hyperparameter configurations and the choice of different convex regularizers in both stochastic and deterministic settings, we have designed a controlled numerical experiment implemented in Python using the Pytorch library. The primary objective is to evaluate and compare the performance of these setups against the optimal policy and optimal value obtained through policy iteration (Puterman [1994], 6.4) in an unregularized setting. The aim of considering an unregularized optimal value function and policy is to assess how regularized and potentially biased frameworks perform compared to the actual problem setup.

This experiment has been conducted in both exact and stochastic frameworks. The experiment utilizes the Approximate Policy Mirror Descent algorithm, as detailed in (4.1.5). APMD is employed with different convex functions and hyperparameter settings to explore their effects on learning dynamics and overall performance. Also, entropy-regularized PMD as an instance of APMD has been implemented.

To measure and compare the performance of considered settings, we have used four different metrics. The first performance metric is the expected cumulative cost gap, which assesses the long-term efficiency and effectiveness of each policy under the optimal steady-state distribution. The second metric uses the KL-divergence to measure how quickly and reliably each setup approaches the optimal policy during the iterative optimization process. The third metric is expected entropy, which quantifies the entropy of each obtained policy, allowing us to compare the relative degree of exploration across setups. The final metric evaluates the distribution mismatch between steady-state distribution of optimal policy and returned policies, as described in (3.4), offering insights into how closely these two distributions over state space align.

6.1 Experimental setup

The experiment is conducted within an environment characterized by a finite set of states S and actions A and random assignment of cost to state-action pairs and transition kernel probabilities.

The environment is defined as follows:

- **States ($\mathcal{S}$)**: A set of *10* states representing the possible configurations of the environment.
- **Actions ($\mathcal{A}$)**: A set of *4* actions available to the agent at each state.
- **Discount factor (γ)**: This factor which weights future expected costs is set to *0.9* in both experiments.
- **Cost Function**: For each state-action pair (s, a), the immediate cost $c(s, a)$ is sampled independently from the uniform distribution $\mathcal{U}(0, 1)$, where the minimum and maximum cost values are 0 and 1, respectively.
- **Transition Kernel**: The state transition probabilities $\mathcal{P}(s' \mid s, a)$ are generated by independently sampling from a normal distribution $\mathcal{N}(0, 1)$ for each potential next state s'. These samples for each possible next state given a fixed state and a fixed action are then normalized over action space using the softmax function to ensure that the transition probabilities form a valid probability distribution over the next states.

To estimate the exact state-action values $Q^\pi(s, a)$ for any given policy π, a Q-function estimator is employed. This estimator iteratively updates the value estimates based on the Bellman equation for regularized Q-function in (4.1.1); this can be derived similarly to the argument in (2.2.7). For the stochastic case, we have used a trajectory generator to sample $M_k = 50$ trajectories with a length of $T_k = 50$ for each round of Q-function estimation based on (2.2.6). The optimal policy $\pi^\star$ is derived using the policy iteration algorithm without regularization, alternating between policy evaluation and policy improvement steps until convergence.

From the optimal policy $\pi^\star$, the steady-state distribution $\nu^\star(s)$ is obtained as the solution to the equation (2.2.15).

Regarding the regularizers embedded in the optimization problems, we have employed ℓ^p norm, i.e.,

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{6.1.1}$$

which stands for convex function $h^p(s)$ in (4.1.5). Other settings consider $h^p(s) \equiv 0$, boiling down the problem to the Entropy regularized PMD discussed in (4.3), and the Log-barrier regularizer (Agarwal et al. [2021]) which employs KL-divergence with maximum entropy distribution as its first argument. For l^p norm and log-barrier settings, since a closed form update is not established, to obtain $Argmin$ in (4.1.5), we have considered softmax parametrized policy discussed in (5.1) and employed the "Adam" optimizer to obtain an approximation of the solution; we have done it for all cases to keep the optimization settings similar even in cases that the closed form solution is established (e.g., 4.3.2). Softmax ensures that the policy remains a valid probability distribution through a projection step onto the simplex with respect to KL-divergence, which is ℓ^1 normalization demonstrated in Example 1.2.

The performance of each scenario is evaluated based on Four key metrics:

- **Expected cumulative cost**. The cumulative cost of the policy π evaluated with respect to the steady-state distribution of the optimal policy.

$$\mathbb{E}_{s\sim\nu^\star}\left[V^{\pi_k}(s) - V^{\pi^\star}(s)\right]$$

- **Convergence to the optimal policy**. The distance between the current policy π_k and the optimal policy $\pi^\star$, measured using the expected Kl-divergence with respect to the steady-state distribution of the optimal policy as in (3.2.6).

$$\mathbb{E}_{s\sim\nu^\star}\left[D^{\pi^\star}_{\pi_k}(s)\right]$$

- **Expected policy entropy**. The expectation of entropy of policy over states with respect to the steady-state distribution of the optimal policy is calculated as

$$\mathbb{E}_{s\sim\nu^\star}\left[H(\pi_k(\cdot|s)\right]$$

$$H(\pi_k(\cdot|s)) = -\sum_{a\in\mathcal{A}} \pi_k(a|s)\log(\pi_k(a|s))$$

- **Distribution mismatch**. As discussed in (3.4), we have implemented the distribution mismatch of the policy returned in each distribution with respect to the optimal policy as follows

$$\left\|\frac{\nu_{\pi_k}}{\nu^\star}\right\|_\infty := \max_{s\in\mathcal{S}} \frac{\nu_{\pi_k}(s)}{\nu^\star(s)}$$

By analyzing these metrics over the iterations, we can compare how different convex regularizers and hyperparameter choices affect the learning dynamics and the overall performance of the APMD in our setup.

Implementation Details

The experiments are conducted using the PyTorch library to leverage its automatic differentiation and optimization capabilities. Key implementation details include:

- **Adam optimizer**: A fixed learning rate of 0.011 and 1500 iterations were used for all scenarios, which are obtained empirically.
- **Initialization**: The policies are initialized uniformly, assigning equal probability to all actions in each state.
- **Random Seed**: A fixed random seed "13" is set to ensure reproducibility of the results.
- **Data type**: We utilized "torch.float64" to ensure high precision in computations, minimizing rounding errors and enhancing numerical stability
- **Algorithms parameters**: Perturbation coefficient τ_k is adjusted dynamically according to Theorem 4 assumption, diminishing exponentially from its initial value of τ_0, unless specified otherwise. η_k is set dynamically in compliance with the stepsize assumption of the theorem as follows

  $$\eta_k = \frac{1-\gamma}{\gamma\tau_k}$$

  ℓ^p and log-barrier regularizers have a coefficient of 0.1 in the objective function, empirically chosen to balance their effect and prevent their value from dominating the objective function value.

Through this experimental setup, we aim to investigate the influence of convex regularization and hyperparameter settings on the performance of the APMD algorithm. The controlled environment and comprehensive evaluation metrics provide a robust framework for understanding how different configurations impact both the convergence behavior and the expected cumulative cost relative to the optimal policy.

6.2 APMD with ℓ^p Regularizers

In this experiment, we consider APMD with four different convex regularizers. Q-function is estimated by sampling, as explained in (6.1). The convex regularizers are

I. $h^\pi(s) \equiv 0$

II. $h^\pi(s) = \|\pi(\cdot|s) - \pi_0\|_1$

III. $h^\pi(s) = \|\pi(\cdot|s)\|_2$

IV. $h^\pi(s) = \|\pi(\cdot|s)\|_3$

with $\tau_0 = 0.1$, $\tau_k = \gamma^k \tau_0$, $\eta_k = \frac{1-\gamma}{\gamma\tau_k}$ in all cases. We conducted experiments for both the exact and stochastic scenarios detailed in Section(6.1). Each figure presents all eight corresponding graphs, with exact and stochastic results for each case depicted in matching colors. The exact case uses a continuous line style and is also clearly distinguishable regarding its stability.

Measure\Case	**APMD**	**AMPD(ℓ^1)**	**APMD(ℓ^2)**	**APMD(ℓ^3)**
Max Δvalue	0.047	0.055	0.059	0.057
Mean Δvalue	0.022	0.012	0.018	0.022
Max ΔKL	0.0338	0.0623	0.2315	0.1657
Mean ΔKL	0.0155	0.0185	0.0377	0.0475
Max Δentropy	0.0247	0.0564	0.0614	0.0581
Mean Δentropy	0.0119	0.0145	0.0177	0.0189
Max Δmismatch	0.046	0.088	0.146	0.076
Mean Δmismatch	0.016	0.023	0.038	0.021

Table 6.1: Contrasts in stochastic and deterministic Q-function estimation configurations

Table 6.1 compares the values of each metric between the stochastic and deterministic instances of the algorithms. Both the mean and maximum values are calculated across all iterations. For example, Max Δ KL, denotes the maximum difference in $\mathbb{E}_{s\sim\nu^\star}\left[D_{\pi_k}^{\pi^\star}(s)\right]$ observed over all iterations k, while Mean Δ KL represents the average of these differences across the iterations. This table allows us to compare the stability of the algorithms and their robustness against estimation errors in the Q-function estimation. The following four figures illustrate the performance of these settings under the mentioned metrics. The obtained values for the KL and Entropy metrics are multiplied by a factor of 10 to enhance the graph's distinguishability.

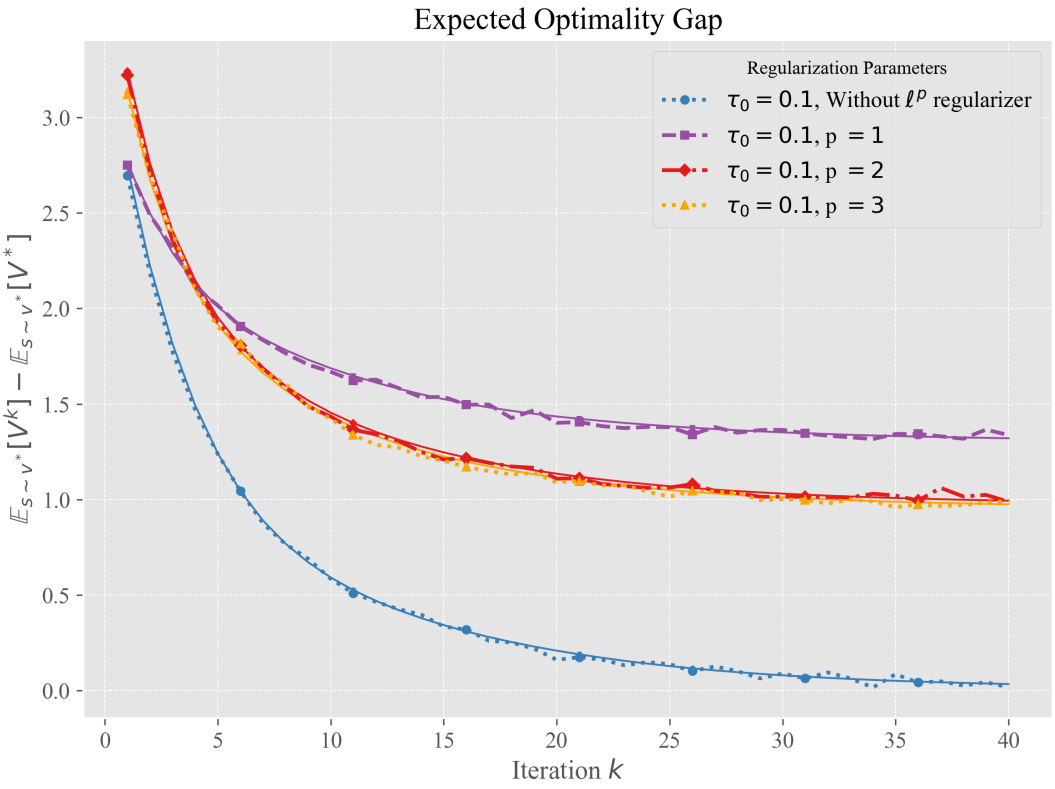

Figure 6.1

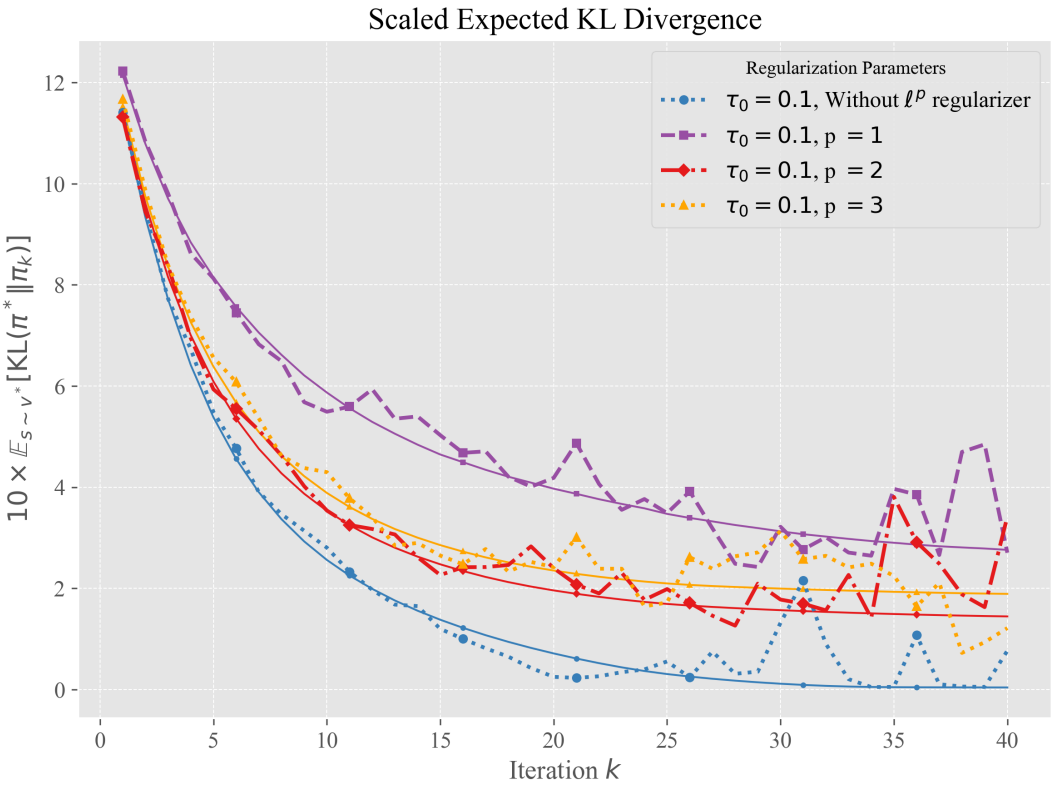

Figure 6.2

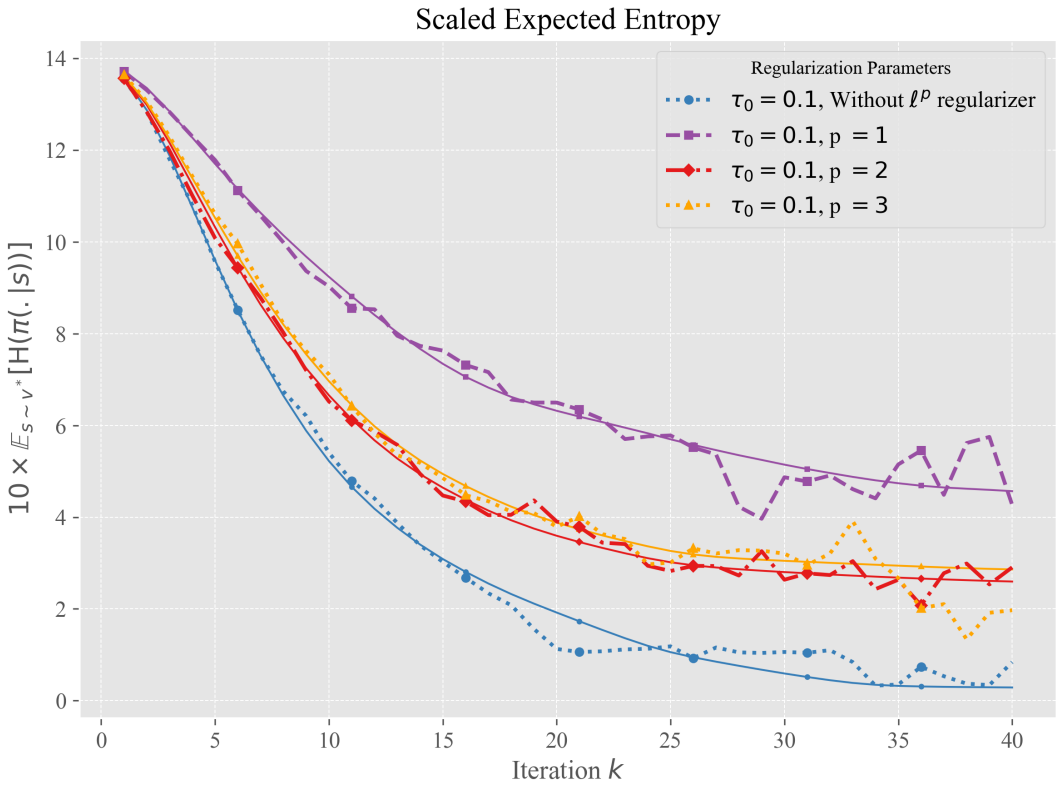

Figure 6.3

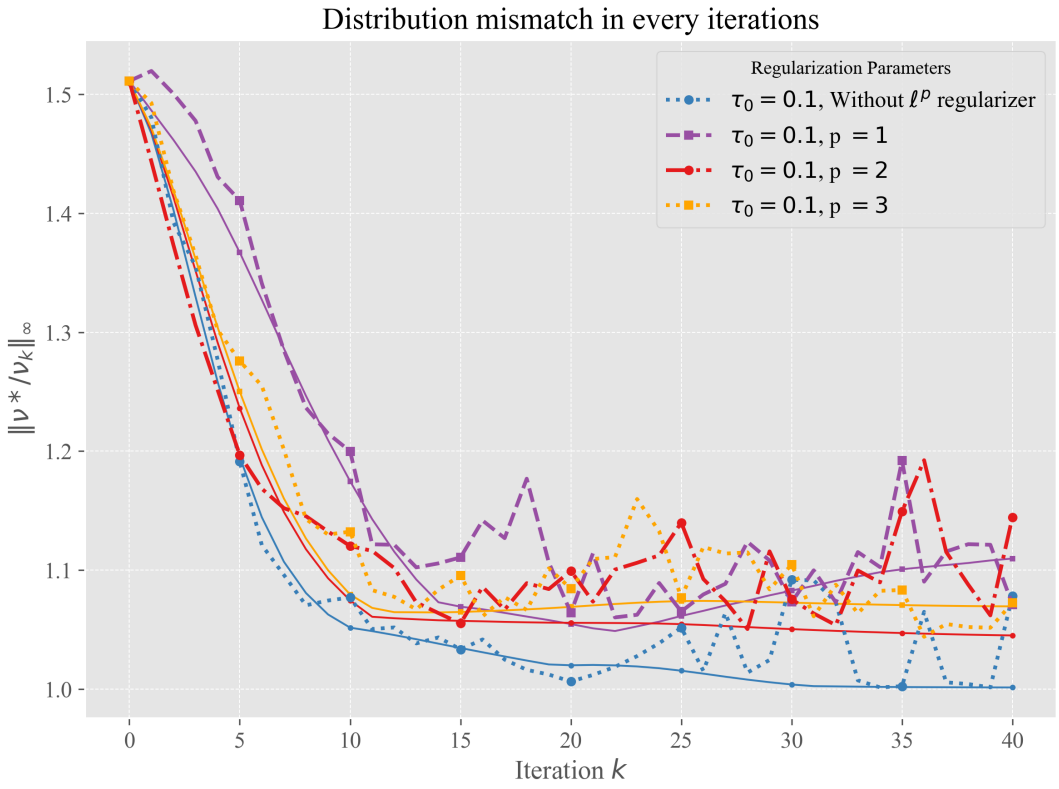

Figure 6.4

The following analysis evaluates the performance of each algorithm using Table 6.1 and Figures 6.1 to 6.4

- Case (I) Achieves the best performance across all metrics, with the fastest and smoothest convergence. The entropy is relatively high, then converges to zero after early iterations, unlike the other cases where a certain degree of exploration is still retained until the end of the process. This effect happened due to the exponentially diminishing factor of the perturbation term, while the convex regularizer effect is kept the same across iterations both in (4.1.5) and (4.1.1).

- **Case (II)** exhibits the poorest performance among all cases, characterized by high variability and slow convergence. This outcome is attributed to maintaining a relatively high entropy, which limits the algorithm's ability to achieve more effective convergence to the optimal policy in terms of both speed and optimality gap. However, this behavior may be advantageous in scenarios that require a greater degree of exploration.

- Case (III) Provides a good balance between stability and convergence. It performs steadily but does not match the speed and effectiveness of the unregularized version.

- Case (IV) Shows similar behavior to Case (III), with slight improvements in some metrics. It is stable but converges more slowly than the unregularized version, falling short of optimal efficiency but still providing consistent performance.

6.3 APMD with convex Regularizers

In this section, we examine four APMD scenarios as follows

I. $\mu = 0.2, \eta = \frac{1-\gamma}{\gamma\tau}, h^\pi(s) \equiv 0$.
In this scenario, only the perturbation term is incorporated into the objective function, utilizing fixed coefficient μ and a fixed stepsize η. By Example (1.1) and Algorithm 2, this case corresponds to PMD with μ-strongly convex regularizer.

II. $\tau_0 = 0.2$, $\tau_k = \gamma^k\tau_0$, $\eta_k = \frac{1-\gamma}{\gamma\tau_k}, h^\pi(s) \equiv 0$.
This case also incorporates the perturbation term; however, both the perturbation coefficient and the stepsizes are dynamic, adhering to the assumptions outlined in Theorem 4.

III. $\tau_0 = 0.2$, $\tau_k = \gamma^k\tau_0$, $\eta_k = \frac{1-\gamma}{\gamma\tau_k}, h^\pi(s) = \|\pi(\cdot|s)\|_2$.

IV. $\tau_0 = 0.2$, $\tau_k = \gamma^k\tau_0$, $\eta_k = \frac{1-\gamma}{\gamma\tau_k}, h^\pi(s) = \mathrm{KL}(\pi_0||\pi)$.
It Is the APMD with log-barrier regularizer as described in (Agarwal et al. [2021]), where π_0 is the uniform distribution over action space.

Similar to the previous experiment, we conducted experiments for both the exact and stochastic scenarios outlined in section (6.1). In each figure, all eight corresponding graphs are displayed, with the exact and stochastic results for each case represented using the same color. Continuous lines represent the exact scenarios.

Measure\Case	**PMD**	**AMPD**	**APMD(ℓ^2)**	**APMD(log-barrier)**
Max Δvalue	0.055	0.033	0.049	0.052
Mean Δvalue	0.022	0.012	0.02	0.018
Max ΔKL	0.0235	0.0541	0.139	0.113
Mean ΔKL	0.0099	0.0178	0.0267	0.0325
Max Δentropy	0.0169	0.0597	0.0487	0.0499
Mean Δentropy	0.0054	0.0154	0.0159	0.0144
Max Δmismatch	0.034	0.057	0.133	0.052
Mean Δmismatch	0.015	0.02	0.029	0.016

Table 6.2: Contrasts in stochastic and deterministic Q-function estimation configurations

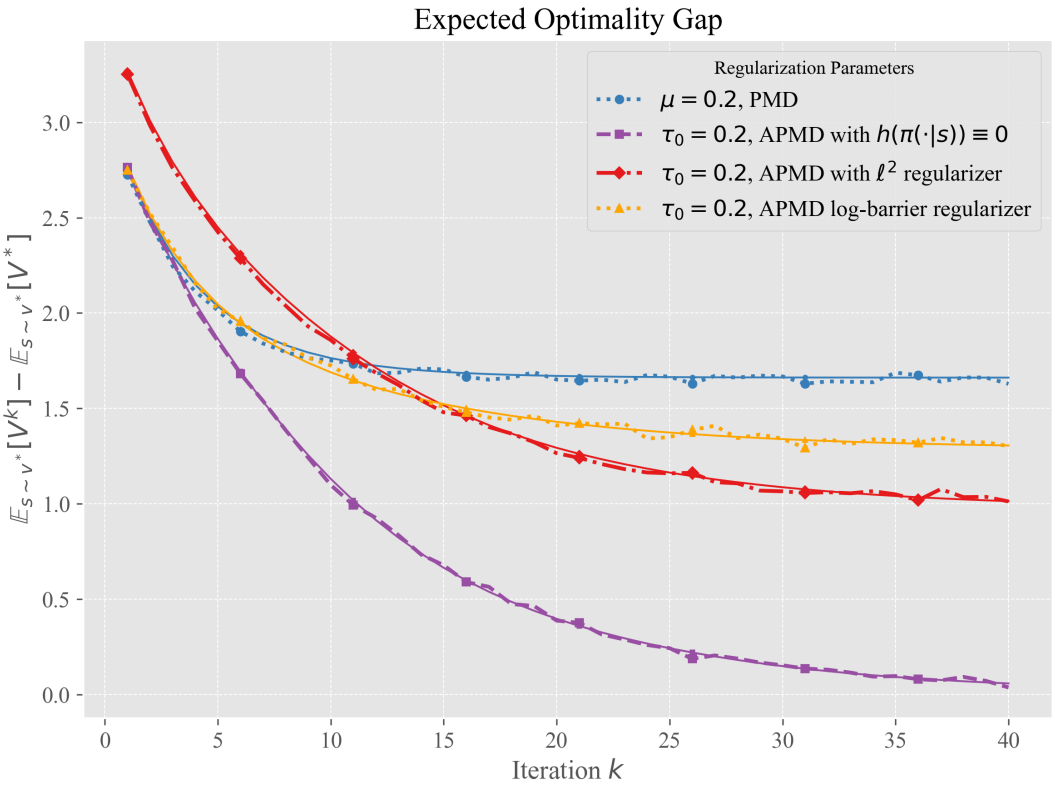

Figure 6.5

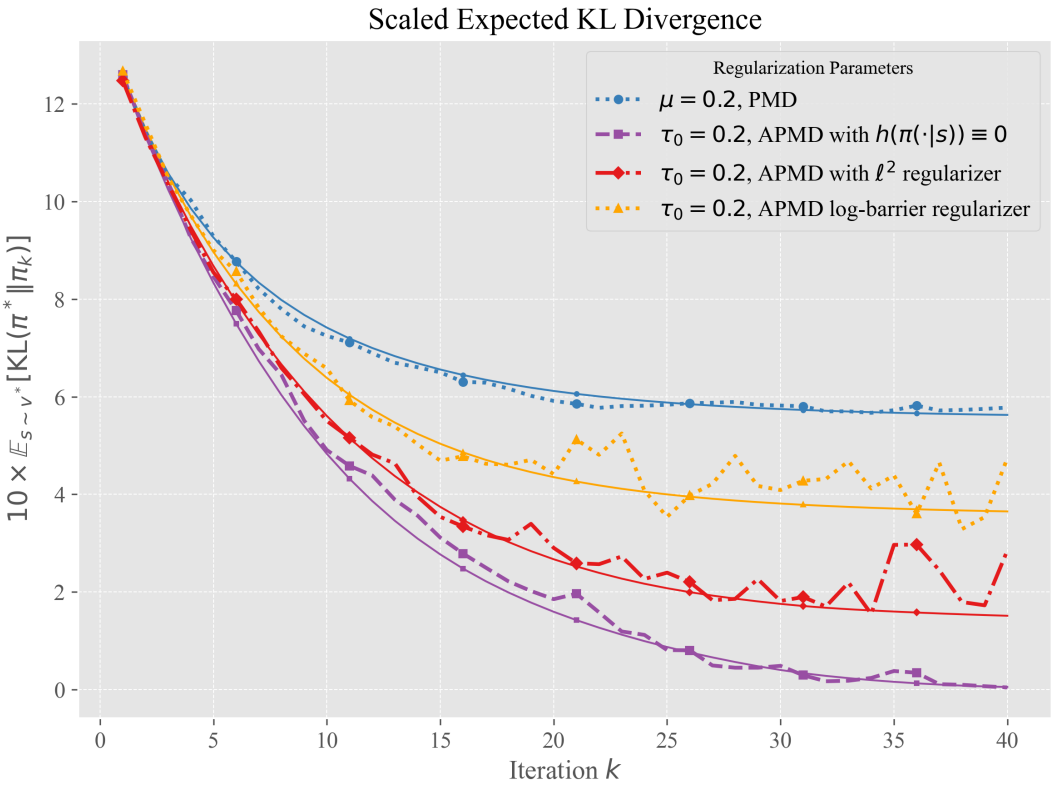

Figure 6.6

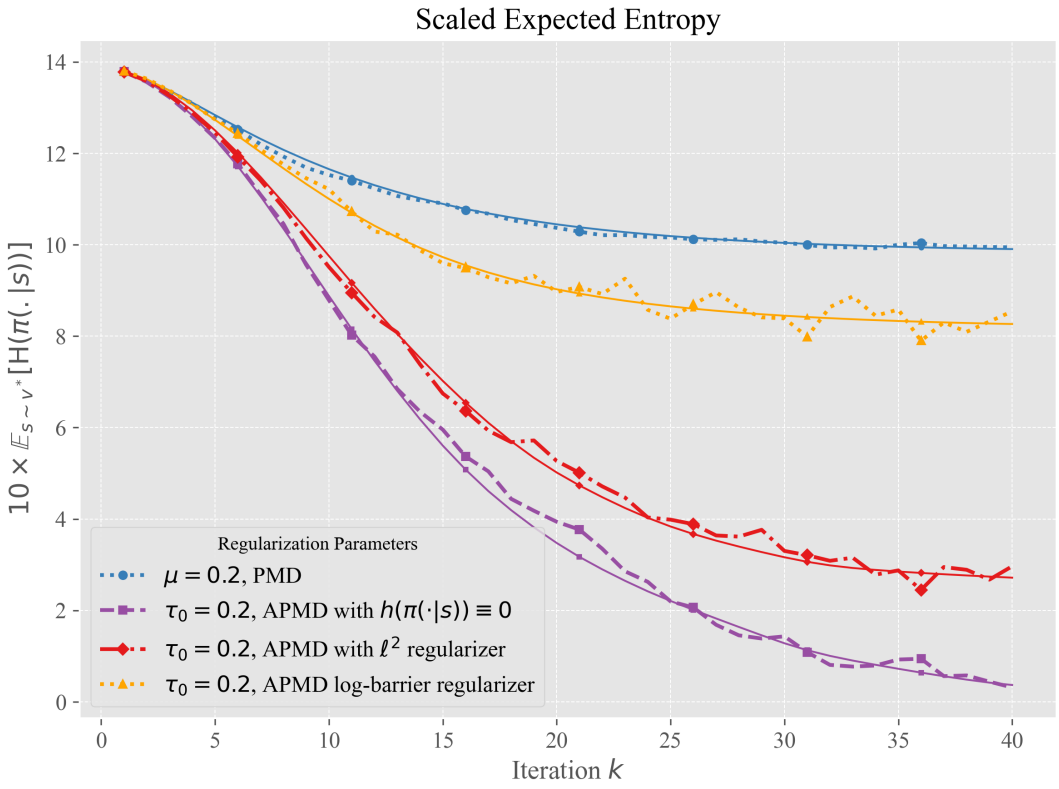

Figure 6.7

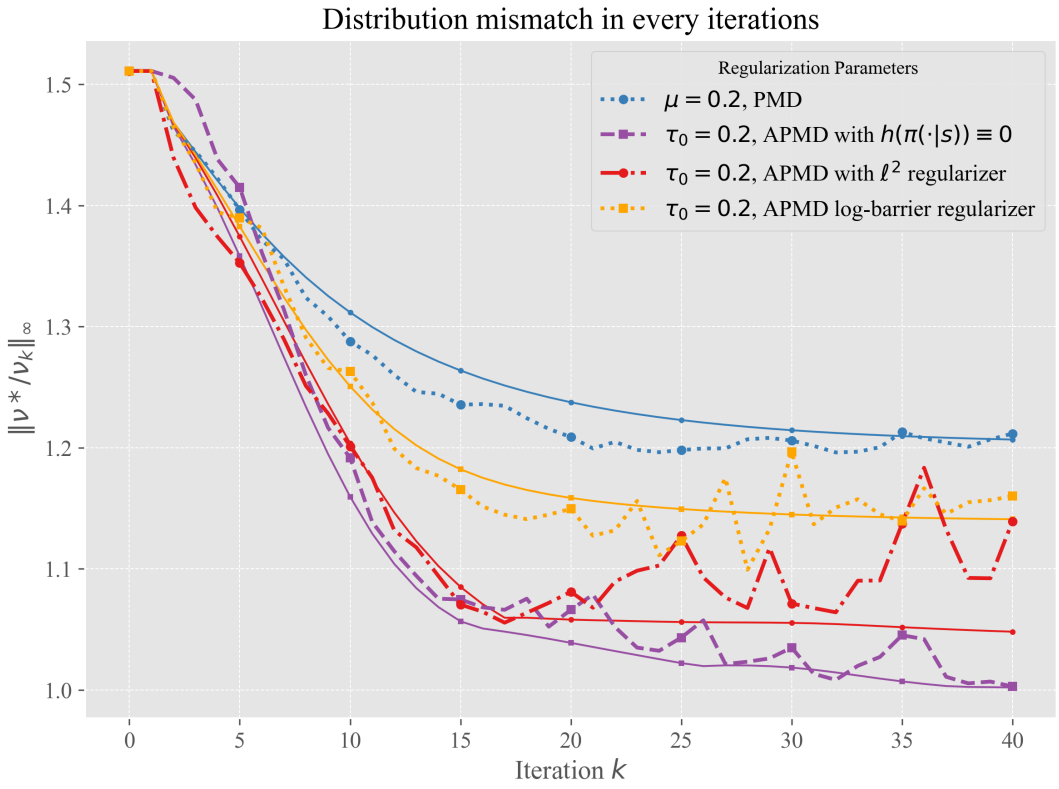

Figure 6.8

An analysis of performance of each algorithm:

- Case (I). PMD performed worse across all metrics. Notably, the behavior of this algorithm in the stochastic setting exhibited less oscillation around the exact case compared to the other approaches, indicating enhanced stability. Additionally, PMD maintained relatively high entropy throughout the iterations, suggesting that it favors exploration relatively more.

- Case (II), APMD without convex regularizer has consistently demonstrated superior performance across all metrics. It showed the fastest reduction in distribution mismatch, scaled entropy, and KL divergence, as well as near zero expected optimality gap with respect to the unregularized optimal value function. This discipline maintains relatively high entropy, hence the exploratory behavior in the early iterations. Then, by the diminishing design of the perturbation term, it tends to behave as an unregularized framework and favors exploitation after the early stages of the optimization.

- Case (III), APMD with the ℓ^2 Initially, the algorithm exhibited higher values concerning the first metric; however, it quickly recovered. It performed notably better than Cases (I) and (IV) while adopting a less aggressive approach compared to Case (II). Nevertheless, it demonstrated lower stability and robustness to noise beyond the initial phase concerning (II) and (I).

- Case (IV), APMD with the log-barrier regularizer had the most inconsistent performance across all metrics. While it achieved some reduction in distribution mismatch and entropy, it struggled to maintain stability, particularly in the stochastic setting. The stochastic version was especially prone to high variance and significant fluctuations, which impacted the effectiveness of both KL divergence reduction and the optimality gap. This behavior suggests that the log-barrier regularizer might prioritize exploration at the cost of convergence stability. Consequently, it was the least effective algorithm overall, demonstrating limited capacity to converge efficiently to the optimal value, particularly in the presence of stochasticity.

Measure\Case	PMD $\mu = 0.2$	APMD $\tau_0 = 0.1$	APMD $\tau_0 = 0.2$	with ℓ^1 $\tau_0 = 0.1$	with ℓ^2 $\tau_0 = 0.2$	with ℓ^2 $\tau_0 = 0.1$	with ℓ^3 $\tau_0 = 0.1$	with log-b $\tau_0 = 0.2$
Max Δval.	0.055	0.047	0.033	0.055	0.049	0.059	0.057	0.052
Mean Δval.	0.022	0.022	0.012	0.012	0.02	0.018	0.022	0.018
Max ΔKL	0.0235	0.0338	0.0541	0.0623	0.139	0.2315	0.1657	0.113
Mean ΔKL	0.0099	0.0155	0.0178	0.0185	0.0267	0.0377	0.0475	0.0325
Max Δent.	0.0169	0.0247	0.0597	0.0564	0.0487	0.0614	0.0581	0.0499
Mean Δent.	0.0054	0.0119	0.0154	0.0145	0.0159	0.0177	0.0189	0.0144
Max Δmis.	0.034	0.046	0.057	0.088	0.133	0.146	0.076	0.052
Mean Δmis.	0.015	0.016	0.02	0.023	0.029	0.038	0.021	0.016

Table 6.3: Combined stability comparison of algorithms

Table 6.3 merges Tables 6.1 and 6.2, and summarizes the stability comparison of the algorithms in different settings in both experiments. Comparing the results of the two experiments indicates that the initial perturbation coefficient, τ_0, directly influences the convergence rate and establishes a higher entropy level.

6.4 Conclusion

In this thesis, we have explored the theoretical foundations and practical applications of policy optimization methods in reinforcement learning, with a particular focus on Policy Mirror Descent, Approximate Policy Mirror Descent (Lan [2022]), and an instance of the Natural Policy Gradient methods (Kakade [2001], Agarwal et al. [2021]), Entropy Regularized NPG method (Cen et al. [2022]). Our work has analyzed connections between these algorithms, providing a coherent structure that bridges theoretical concepts with empirical observations.

We began by establishing the theoretical background necessary for understanding advanced optimization algorithms, including projected subgradient descent, Bregman divergence (L. Bregman [1967]), variational inequalities (Lan [2021]), and mirror descent (Nemirovski et al. [2009], Bubeck [2015], Beck and Teboulle [2003]). Building upon this foundation, we presented the reinforcement learning setup (Sutton and Barto [2020]), detailing the formulation of Markov Decision Processes and fundamental concepts such as policies, value functions, and Bellman equations (Puterman [1994]). We introduced a regularized framework to address issues related to convergence and stability, laying out key lemmas that support the theoretical analysis of policy optimization algorithms primarily based on Lan [2022] and Cen et al. [2022].

Our investigation into PMD and its stochastic variant, SPMD, provided insights into their convergence behaviors and the circumstances under which the stochastic case could achieve a similar convergence rate as the exact scenario. We thoroughly analyzed the convergence properties of PMD, reviewed its effectiveness in deterministic settings, and

the extension to stochastic environments with SPMD. We argued about how to change the performance evaluation distribution and the effect of this change on the optimality gap based on the concept of distribution mismatch.

In exploring APMD, we examined its setup in line with the PMD discipline, analyzed the convergence properties of the algorithm, and established the closed-form solution for a special case of the algorithm. A contribution of this thesis is establishing a theoretical connection between APMD and the entropy-regularized NPG method. By demonstrating that entropy-regularized NPG can be viewed as an instance of APMD under specific step-size choices, we provided a unified perspective that enhances the understanding of these algorithms, highlighting the generality of the APMD setting. This connection allows for the transfer of theoretical results between the methods and opens avenues for developing new algorithms that leverage the strengths of both approaches.

In the experimental component of our study, we conducted comprehensive numerical experiments to compare the performance of Approximate Policy Mirror Descent (APMD) using various ℓ^p regularizers alongside the log-barrier function. Building upon the foundational framework established by Lan [2022], we systematically analyzed how different regularization techniques and hyperparameter settings influence the convergence rate and stability of APMD. Our analysis focused on evaluating the algorithm's ability to efficiently reach optimal policies while maintaining robust performance across iterations. The results indicate that APMD without a convex regularizer component can significantly enhance convergence speed and offer relatively good robustness against estimation errors.

By carefully tailoring the choice of appropriate regularizer, leveraging the algorithm's flexibility and generality, and precisely tuning hyperparameters, APMD can be effectively adapted to specific use cases, thereby enhancing the efficiency and reliability of reinforcement learning applications across diverse operational settings.

Bibliography

A. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983. ISBN 0471103454.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift. *Journal of Machine Learning Research*, 22:1–76, 2021. URL http://jmlr.org/papers/v22/19-736.html.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the Impact of Entropy on Policy Optimization. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Carlo Alfano, Rui Yuan Stellantis, and Patrick Rebeschini. A Novel Framework for Policy Mirror Descent with General Parameterization and Linear Convergence. *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, 36:30681–30725, 2024.

Shun-Ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation, MIT Press*, 10(2):251–276, 2 1998. doi: 10.1162/089976698300017746.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety, 6 2016. URL http://arxiv.org/abs/1606.06565.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey, 11 2017. ISSN 10535888.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003. doi: https://doi.org/10.1016/S0167-6377(02)00231-6. URL www.elsevier.com/locate/dsw.

Richard Bellman. A Markovian Decision Process. *Indiana University Mathematics Department*, 1957.

Dimitri Bertsekas. *A Course in Reinforcement Learning*. Athena Scientific, Belmont, Massachusetts, 2 edition, 2024. ISBN 1886529299. URL http://www.athenasc.com.

Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. ISBN 1886529108.

Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Incremental Natural Actor-Critic Algorithms. *Advances in neural information processing systems,*, 20:105–112, 2007.

Mariu Bojarski, Davide Del Testa, Daniel Dworakowski, and et al. End to End Learning for Self-Driving Cars, 2016. URL https://www.researchgate.net/publication/301648615.

Boris T. Polyak. *Spectral theory of random fields*. Optimization Software, 1987. ISBN 0911575006.

Stephen P.. Boyd and Lieven. Vandenberghe. *Convex optimization*. Cambridge University Press, 2014. ISBN 9780521833783.

Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. *Theory of Convex Optimization for Machine Learning*, 8:231–357, 5 2015. doi: https://doi.org/10.48550/arXiv.1405.4980. URL http://arxiv.org/abs/1405.4980.

Semih Cayci, Niao He, and R. Srikant. Convergence of Entropy-Regularized Natural Policy Gradient with Linear Function Approximation. *SIAM Journal on Optimization*, 34:2729–2755, 6 2024a. doi: 10.1137/22M1540156.

Semih Cayci, Niao He, R Srikant, and Csl Ece. Finite-Time Analysis of Entropy-Regularized Neural Natural Actor-Critic Algorithm. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization. *Operations Research*, 70(4):2563–2578, 7 2022. ISSN 15265463. doi: 10.1287/opre.2021.2151.

Gong Chen and Marc Teboulle. CONVERGENCE ANALYSIS OF A PROXIMAL-LIKE MINIMIZATION ALGORITHM USING BREGMAN FUNCTIONS. *SIAM J. OPTIMIZATION*, 3(3):538–543, 1993. doi: https://doi.org/10.1137/0803026. URL http://www.siam.org/journals/ojsa.php.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 9 1999.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: Convergent Reinforcement Learning with Nonlinear Function Approximation. In *Proceedings of Machine Learning Research*, 2018. URL https://arxiv.org/abs/1712.10285.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization * Elad Hazan. In *Journal of Machine Learning Research*, volume 12, pages 2121–2159, 2011.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A Theory of Regularized Markov Decision Processes. In *Proceedings of Machine Learning Research*, 1 2019. URL http://arxiv.org/abs/1901.11275.

Gerald Tesauro and Tom Keith. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3), 1995.

Geoffrey J Gordon. Stable Function Approximation in Dynamic Programming. In *Machine Learning Proceedings*, pages 261–268. Morgan Kaufmann, 1995. doi: https://doi.org/10.1016/B978-1-55860-377-6.50040-2.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement Learning with Deep Energy-Based Policies. In *Proceedings of Machine Learning*, 2 2017. URL http://arxiv.org/abs/1702.08165.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Machine Learning Research*, pages 1856–1865. PMLR, 2018.

Elad Hazan. Introduction to Online Convex Optimization, 2023. URL http://arxiv.org/abs/1909.05207.

Caleb Ju and Guanghui Lan. Policy Optimization over General State and Action Spaces, 11 2022. URL http://arxiv.org/abs/2211.16715.

Sham Kakade. A Natural Policy Gradient. In *Advances in Neural Information Processing Systems*, 2001. URL http://www.gatsby.ucl.ac.uk.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. John Wiley, 2007. ISBN 978-3-527-40760-6.

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On the Linear Convergence of Natural Policy Gradient Algorithm. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2021-December, pages 3794–3799. Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781665436595. doi: 10.1109/CDC45484.2021.9682908.

Vijay R Konda and John N Tsitsiklis. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems*, 2000.

L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, pages 200–217, 1967.

Guanghui Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Springer Cham, 5 2021. doi: https://doi.org/10.1007/978-3-030-39568-1. URL http://www.springer.com/series/13852.

Guanghui Lan. Policy Mirror Descent for Reinforcement Learning: Linear Convergence, New Sampling Complexity, and Generalized Problem Classes. *Mathematical Programming*, pages 1059–1106, 4 2022. doi: https://doi.org/10.1007/s10107-022-01816-5. URL http://arxiv.org/abs/2102.00135.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-End Training of Deep Visuomotor Policies. *Journal of Machine Learning Research*, 17:1–40, 2016.

Yan Li, Guanghui Lan, and Tuo Zhao. Homotopic Policy Mirror Descent: Policy Convergence, Implicit Regularization, and Improved Sample Complexity, 1 2022. URL http://arxiv.org/abs/2201.09457.

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural Proximal/Trust Region Policy Optimization Attains Globally Optimal Policy. In *Advances in Neural Information Processing Systems*, 6 2019. URL http://arxiv.org/abs/1906.10306.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. On the Global Convergence Rates of Softmax Policy Gradient Methods. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Francisco S Melo and M Isabel Ribeiro. Q-Learning with Linear Function Approximation. In *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2007.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *CoRR*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2 2015. ISSN 14764687. doi: 10.1038/nature14236.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximat ion approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. ISSN 10526234. doi: http://dx.doi.org/10.1137/070704277.

Jan Peters and Stefan Schaal. Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190, 3 2008. ISSN 09252312. doi: 10.1016/j.neucom.2007.11.026.

M.L. Puterman. *Markov Decision Processes*. Wiley Series in Probability and Statistics, 1994. doi: https://doi.org/10.1002/9780470316887.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 7 2017. URL http://arxiv.org/abs/1707.06347.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL www.aaai.org.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 1 2016. ISSN 14764687. doi: 10.1038/nature16961.

Richard S.. Sutton and Andrew G.. Barto. *Reinforcement learning : an introduction*. The MIT Press, 2020. ISBN 9780262039246.

Richard S Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, 1999.

Manan Tomar, Lior Shani, Yonathan Efroni, Mohammad Ghavamzadeh, and Google Research. Mirror Descent Policy Optimization, 2020.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos Deepmind, and Matthieu Geist. Leverage the Average: an Analysis of KL Regularization in Reinforcement Learning. In *Neural Information Processing Systems*, 2020.

M Wiering and M van Otterlo. *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27644-6. doi: 10.1007/978-3-642-27645-3. URL https://link.springer.com/10.1007/978-3-642-27645-3.

Ronald J Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 5 1992. doi: https://doi.org/10.1007/BF00992696.

Ronald J. Williams and Jing Peng. Function Optimization using Connectionist Reinforcement Learning Algorithms. *Connection Science*, 3(3):241–268, 1 1991. ISSN 13600494. doi: 10.1080/09540099108946587.

Lin Xiao. On the Convergence Rates of Policy Gradient Methods, 1 2022. URL http://arxiv.org/abs/2201.07443.

Rui Yuan, Simon S Du, Robert M Gower, Alessandro Lazaric, and Lin Xiao. Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies. *ICLR*, 2023.

Valentina Zangirolami and Matteo Borrotti. Dealing with uncertainty: Balancing exploration and exploitation in deep recurrent reinforcement learning. *Knowledge-Based Systems*, 293, 6 2024. ISSN 09507051. doi: 10.1016/j.knosys.2024.111663.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy Mirror Descent for Regularized Reinforcement Learning: A Generalized Framework with Linear Convergence. *SIAM Journal on Optimization*, 33:1061–1091, 2023. doi: 10.1137/21M1456789. URL http://arxiv.org/abs/2105.11066.

Brian D Ziebart. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. Technical report, Carnegie Mellon University, 2010.