Lagrangian Duality for Efficient Large-Scale Reinforcement Learning

Joan Bas Serrano

TESI DOCTORAL UPF / 2022

DIRECTOR DE LA TESI
Dr. Gergeley Neu

Departament de Tecnologies de la Informació i les Comunicacions

upf. Universitat Pompeu Fabra Barcelona

Acknowledgements

És difícil decidir per on començar a agraïr després de cinc anys acompanyat per tanta gent fantàstica que han fet possible que aquest text fos possible.

I will first thank my supervisor Gergely Neu. Thank you for your effort and dedication, for always finding time to help and guide me and for making me feel that I was not alone in this project. I think I would have never finished this project without the optimism, the hope and the energy that you always project. It has been an honor to have such a brilliant and caring supervisor.

From the university, I also want to thank all the great people from the AI&ML group. Especial thanks to Julia, Miquel, Lorenzo and Germano, who have accompanied me during the whole trip. I feel sad for leaving now that the group is growing and becoming so active in both work and fun-wise. Gràcies també a la gent del GTI i als respectius satèl·lits (Adri i Celi). Gràcies en especial a la Bea per estar sempre preparada per ajudar-me en tot. I no puc acabar aquest paràgraf sense donar gràcies a la Lydia, per salvar-me dels embolics administratius en els que em poso cada dos per tres.

I also want to thank my coauthor and talented researcher Sebas for his contributions in the "Logistic Q-learning" paper, to Yasin for giving me the opportunity to spend some time doing research in Vietnam, and to my collegues from Socialpoint for showing me how cool can it be doing applied research in a company.

De fora de la universitat, gràcies a la meva familia, en especial als meus pares, David i Lluïsa, per, simplement, estar sempre al meu costat acompanyant-me. Seguint amb la família, gràcies a tots els meus amics. Gracies per ajudar-me a desconectar (massa i tot a vegades) i a relativitzar tots els problemes i preocupacions derivats d'aquest projecte que trobareu a continuació. Especial gràcies a la Maria i al Marçal per aguantar sempre les meves turres i fer-me costat en tot. També a l'Anna, pel suport sobretot en la primera part del doctorat. I moltes gràcies Ari pel teu afecte, per la teva sabiduria i per ser tant bona companyia; sense tu aquest últim any de recta final hauria estat molt més dur.

No poso els noms de la resta de persones importantíssimes en la meva vida perquè se'm desmadra el text (sí, ho he provat). Moltes gràcies a tots per fer que els moments difícils no ho fossin tant i que els bons ho fossin molt més.

Abstract

Reinforcement learning is an expanding field where very often there is a mismatch between the high performance of the algorithms and their poor theoretical justification. For this reason, there is a need of algorithms that are well grounded in theory, with strong mathematical guarantees and that are efficient in solving large-scale problems. In this work we explore the linear programming approach for optimal control in MDPs. In order to develop novel reinforcement learning algorithms, we apply tools from constrained optimization to this linear programming framework. In concrete, we propose a variety of new algorithms using techniques like constraint relaxation, regularization or Lagrangian duality. We provide a formal performance analysis for all of these algorithms, and evaluate them in a range of benchmark tasks.

Keywords : Reinforcement learning, Lagrangian duality, linear programming, constraint relaxation, convex optimization, entropy regularization.

Resum

L'aprenentatge per reforç (en anglès, reinforcement learning) és un camp en expansió on tot sovint la gran eficàcia dels algorismes no va de la mà d'una bona justificació teòrica d'aquests. Per aquest motiu, hi ha la necessitat d'algorismes ben fonamentats en la teoria, amb garanties matemàtiques robustes, i que a la vegada siguin eficients a l'hora de resoldre problemes de gran escala. En aquest treball explorem la formulació basada en programació lineal per al control òptim en problemes de decisió de Markov. Per tal de desenvolupar nous algorismes d'aprenentatge per reforç, apliquem eines del camp de l'optimització de funcions convexes a la formulació basada en programació lineal. En concret, utilitzem tècniques com la relaxació de condicions, la regularització, o la dualitat Lagrangiana. També elaborem una anàlisi formal del rendiment d'aquests algorismes i els avaluem en diferents tasques de referència.

Paraules clau : Aprenentatge per reforç, dualitat Lagrangiana, programació lineal, relaxació de condicions, optimització convexa, regularització entròpica.

Table of contents

Chapter 1

INTRODUCTION

1.1 An expanding field

Not many areas have received so much excitement and hype during the last decade as artificial intelligence (AI). It has attracted the attention of both industry and general population, and anything related to AI has been linked in the collective imaginary to "future" and "progress" (not always in the good sense of these words). AI is present in robotics, healthcare, finance, E-commerce, marketing or gaming among others. It is there every time that a platform recommends us a movie, a song or a product, when you get a fraud alarm from your bank, when an e-mail goes to the spam folder or when you receive personalized advertisements as if someone was reading your mind and knew your taste and interests.

AI is everywhere and, what is more important, it is still a young and growing field, which makes it very exciting and promising. From regular people to world-leading companies, AI has gained everyone's interest. Actually, as if it was a modern version of the space race, some of the most important and popular companies of the world like Google, Meta (Facebook), Amazon, Apple or Microsoft invest every year insane amounts of resources in research in AI, and in most cases it is not even for improving or developing products and services, but just for leading the AI race.

And why now? There are two main reasons that justify why the fast growth of artificial intelligence has been possible this last decade and not before. The first reason is that thanks to the Internet and Internet of things (IoT) devices, the amount of generated data that can be used to train the AI models has grown exponentially. The second reason is that the increase in computational power makes it possible to process large amounts of data, which was totally impossible before. These two factors together with the huge increase in research, has made AI to grow surprisingly fast, and bring groundbreaking news every year, in both fundamental and applied AI.

It is a fact that AI is changing the world and the way how we interact with it. Nevertheless, in this maelstrom of hype and promises, it is usually difficult to distinguish between real progress and made up results that seem to have solved artificial intelligence. As pointed out before, AI is still a very young field with a lot of open questions and exciting challenges, and we are still *very far* from anything with a flavor of a general AI.

Despite it is out of the scope of this work, we do not want to end this section without pointing out that the usage of the powerful set of tools that AI brings is not always aligned with the interests of the population, being sometimes harmful and dangerous. During the last years we have seen how AI can be used to manipulate large scale volumes of people based on their personal data, and even alter the result of elections. This is allegedly the case of Trump's or Brexit's campaigns among *many* others. Also, all the popular social networks use AI to maximize the time spent in their platforms by creating dependence in their users. The used methods are known to spread fake news and polarize opinion as a biproduct of the final goal. These are just some examples to which we could add automatic weapons, vigilance or super aggressive and personalized marketing campaigns. We think it is of a vital importance to tackle these problems if we do not want to end up in a dystopia where concepts as democracy, freedom or privacy start losing their meaning. As a beloved uncle once said: ”with great power comes great responsibility“.

While the present thesis does not aspire to address the possible misuses of AI systems, it does aim to humbly contribute to the development of more reliable and robust AI tools. Indeed, the goal of this thesis is to develop algorithms with a solid theoretical backing and in particular to provide tools that are well-rooted in theoretical foundations and come with verifyable guarantees on their performance. This effort arguably addresses some of the most basic concerns associated with learning systems: before we can make sure that our systems do not harm humans, we need to make sure that they at least act according to their specifications and achieve their intended goals efficiently.

1.2 The RL problem

The work of this thesis lies in a particular area of AI related to sequential decision-making that is known as reinforcement learning (RL) [Szepesvári, 2010; Sutton and Barto, 2018]. In concrete, the objective in RL is to learn how to behave in a complex and unknown environment by interacting with it, in order to maximize some measure of the reward collected in that environment.

RL is formalized mathematically using the *Markov Decision Process* (MDP) framework that will be described in detail later in this text. In the meanwhile, we can understand it as a mathematical framework consisting of the following parts:

- States: A state is a configuration of the environment, and it should contain all the important information of the problem that is modeled. For example, in a chess game, the configuration of the board in a given moment is a state.

- Actions: The actions are used by the agent to modify the state. Every time an agent is in a given state and applies an action, it moves to a next state.

- Transition probabilities: These are the probabilities of going from one state to another when a given action is taken. They encode the dynamics of the MDP.

- Rewards: Every time the agent picks an action in a given state, it receives a reward.

In RL, an agent interacts with an environment defined as an MDP and tries to select the best sequence of actions in order to maximize some notion of cumulative rewards. The main challenge is that the reward function and the transition probabilities are unknown. For this reason, it is important to explore the environment to get some knowledge about the rewards and problem dynamics. The balance between exploring the environment and exploiting those behaviors that we have already seen to be useful is a very well known topic in RL: the *exploration-exploitation trade-off*. Another feature that makes the RL problem particularly challenging is the fact that the actions in the present have impact in the future. Since different actions make the agent to move to different states, the sequence of visited states depend on the actions taken by the agent. For this reason, it is important to realize that maximizing the immediate reward is often not the best strategy, and that the agent should learn strategies that collect large amounts of reward in the long run.

1.3 A problem of RL

As it often happens in science and engineering, in RL (and AI in general), theory and applications do not evolve at the same speed. RL is a very young field with a lot of opportunities, where the cost of trying new ideas is very low and where lot of new results appear every day. In such a field, it is easy to get lost in the fight for beating benchmarks and forget about the foundations and mathematical understanding. This produces a huge amount of literature based on improving the performance of certain algorithms on particular datasets by tuning parameters or introducing intuition-driven modifications.

In many cases, this "trial and error" approach has worked very well and has produced algorithms with impressive empirical performance, but with poor theoretical justification and analysis. This phenomena is increased by the usage of deep neural networks (DNN), an incredibly powerful tool for which the theoretical understanding is still very limited. As an example, we have the DQN algorithm of Mnih et al. [2015], that is one of the most popular algorithms in RL due to its simplicity and its great performance. Despite this, DQN is only partially justified by theory and has only extremely limited convergence guarantees. This is not an isolated case; there are plenty of algorithms and tricks to "make things work" that are present in the basic RL toolkit and that do not come from a very well-founded theory. For example, the squared Bellman error is a broadly used loss function in RL that is not directly motivated by theory and has a number of undesirable properties. Furthermore, methods based on its recursive optimization are known to be unstable. Despite this, the squared Bellman error appears in most of the state-of-the-art algorithms like DQN [Mnih et al., 2015], TRPO [Schulman et al., 2015], SAC [Haarnoja et al., 2018], A3C [Mnih et al., 2016], TD3 [Fujimoto et al., 2018], MPO [Abdolmaleki et al., 2018] or POLITEX [Abbasi-Yadkori et al., 2019] among others.

On the other hand, algorithms that are well-grounded in theory are usually not very practical, and their performance is not comparable to the ones mentioned above, which makes them receive less attention in the performance-oriented world described above.

Despite not being intrinsically bad, this situation has two major problems that motivate the direction of our work. The first of these problems is related to the heuristic used to push the progress of RL. While intuition and trial-and-error are very useful for doing some exploratory work and motivating insightful questions, a well developed theory is fundamental to keep progressing in promising directions. Furthermore, with the huge amount of applied research being published, theory is crucially important to unify and explain the different and independent experimental results and methods. It is not until the theory and mathematical proofs appear, that the beliefs, intuitions and observations become trustworthy knowledge that can be used safely. Without a solid understanding of the algorithms and the problems that we want to solve, it seems difficult to face the important challenges of RL. We need a solid and robust theoretical basis to build on top of if we do not want to end up working in an alchemy-like discipline.

The second problem is that for some tasks, a perfect understanding of the algorithm and its limitations is needed, in the form of a mathematical description and characterization of its performance and convergence behaviour. This is the case of tasks where safety is a must like applications in health, self-driving vehicles, economy, or any other field where the cost of a mistake is very high.

For these reasons, the aim of this work is to develop algorithms that are well rooted in theory and provide a mathematical analysis of their performance.

1.4 Our approach

RL has its roots in the theory of control and sequential decision making, where it is commonly assumed that the decision-maker has full access to a model of the environment that governs the state dynamics. There, the most classical approach to compute optimal policies in MDPs is through the method of *dynamic programming*, understood in this context as computing fixed points of certain operators [Bellman, 1957; Howard, 1960; Bertsekas, 2007].

The use and influence of dynamic-programming methods like value iteration and policy iteration extend well beyond the world of decision and control theory, as the underlying ideas serve as foundations for most algorithms for *learning* optimal policies in unknown MDPs: the setting of RL.

While being hugely successful, DP-based methods have the downside of being somewhat incompatible with classical machine-learning tools that are rooted in convex optimization. Indeed, most of the popular reductions of dynamic programming to (non-)convex optimization are based on heuristics that are not directly motivated by theory, which is, in our opinion, one of the root causes of the general lack of theoretical and mathematical understanding explained in the previous section. Examples include algorithms already mentioned like the celebrated DQN that reduces value-function estimation to minimizing the “squared Bellman error”, or the TRPO algorithm that reduces policy updates to minimizing a “regularized surrogate objective”. While these methods can be justified to a certain extent, it is technically unknown if solving the resulting optimization problems actually leads to a desirable solution to the original sequential decision-making problem.

In this work we explore an alternative approach based on linear programming (LP) that was first proposed roughly at the same time as the DP methods of Bellman [1957]; Howard [1960]: the idea of LP-based methods for sequential decision-making goes back to the works of de Ghellinck [1960]; Manne [1960]; Denardo [1970]. While LP-based methods seem to be more obscure in present day than DP methods, they have the clear advantage that they lead to an objective function directly amenable to modern large-scale optimization methods. Recent reinforcement-learning methods inspired by the LP perspective include policy-gradient and actor-critic methods [Sutton et al., 1999; Konda and Tsitsiklis, 1999] and various “entropy-regularized” learning algorithms (e.g., Peters et al., 2010; Zimin and Neu, 2013; Neu et al., 2017). While these methods promise to directly tackle the policy-optimization problem through solving the underlying linear program, most of them still require the computation of certain value functions through approximate dynamic programming, which is typically done through a minimization of a heuristic objective like the squared Bellman error.

In our work, we argue for the viability of methods fully based on convex optimization, rooted in the LP approach. Such way of working allow us to develop theoretically grounded practical algorithms for which it is possible to show performance guarantees.

1.5 Thesis structure

The dissertation is structured in the following parts: the background (Chapters 2 and 3), the results regarding a linearly relaxed saddle-point problem for finding optimal policies presented in [Bas-Serrano and Neu, 2020] (Chapter 4), the results regarding the new `Q-REPS` algorithmic scheme presented in [Bas-Serrano et al., 2021] (Chapter 5), and a very brief conclusions chapter (Chapter 6). These chapters are structured as follows:

- In Chapter 2 we cover the necessary background in convex optimization. In the first part of this chapter we go through basic concepts and definitions, present a formal framework for constrained optimization problems and introduce Lagrangian duality and its implications. The second part of this chapter is dedicated to convex optimization algorithms. There we present two specific algorithms called "mirror descent" and "mirror prox", and provide some theoretical guarantees regarding their convergence. We also show how these algorithms can be used for saddle-point optimization.

- In Chapter 3 we cover the background regarding reinforcement learning. We start by formulating the RL problem as a Markov decision process. We then describe the two settings that are used in this work. We also provide some tools for evaluating policies and finding the optimal policy in those settings. Finally, we introduce some approximate dynamic programming methods for finding optimal policies.

- In Chapter 4 we present our first set of contributions: an approach based on a linearly relaxed version of a saddle-point problem that characterizes the optimal solution in MDPs. We first introduce the bilinear saddle-point formulation of the MDP optimization problem, and present a linearly parameterized version of this problem that enables to reduce the dimensionality of the problem. We characterize a set of assumptions that allow a reduced-order saddle-point representation of the optimal policy, and propose an algorithm with convergence guarantees that shows the sufficiency of the assumptions.

- In Chapter 5 we present our second set of contributions: a new reinforcement learning algorithm derived from a regularized linear-programming formulation of optimal control in MDPs. We first present the constrained optimization problem that we aim to solve and from which we derive a new loss function for policy evaluation that serves as an alternative to the widely used squared Bellman error. We then use this new loss function that we call logistic Bellman error to build the new algorithmic scheme called `Q-REPS`. We also analyze the error propagation of `Q-REPS`. After that, we provide a practical saddle-point algorithm (with two variants) and derive bounds on their performance. Finally, we show the effectiveness of our method on a range of benchmark problems.

- In Chapter 6 we extract some conclusions, point out the main directions for future work and highlight the main takeaways of the work.

1.6 Contributions

The main contributions of this thesis are the results of the works published under the names “Faster saddle-point optimization for solving large-scale Markov decision processes” [Bas-Serrano and Neu, 2020] and “Logistic Q-Learning” [Bas-Serrano et al., 2021], that are presented in Chapters 4 and 5 respectively.

In the first of these works we study the saddle-point formulation of MDPs under linear approximation. The first main contribution of this work is the characterization of a set of assumptions that allow a reduced-order saddle-point representation of the optimal policy. These include a realizability assumption and a newly identified coherence assumption about the subspaces used for approximation. The second main contribution is the design of a mirror-prox based optimization algorithm with fast convergence rates that are independent of the size of the state space. We show that the algorithm outputs an ε-optimal policy with runtime guarantees of $\widetilde{\mathcal{O}}\left(\tau_{mix}^2 m^2/\varepsilon\right)$, where m is the number of variables in the relaxed optimization problem, and τ_{mix} is a notion of mixing time. The analysis of this algorithm shows some useful tools for connecting the duality gap of the solution output by our algorithm with the suboptimality gap of the extracted policy.

In the second work we develop and analyze a new reinforcement learning algorithm called `Q-REPS` that is derived from a regularized linear programming formulation of optimal control in MDPs. The algorithm is closely related to `REPS` (Peters et al. [2010]) but significantly more practical due to (1) the introduction of a Q-function that enables efficient model-free implementation and (2) a convex loss function for policy evaluation that serves as a theoretically sound alternative to the widely used squared Bellman error heuristic due to its favourable properties.

We call this new loss function the logistic Bellman error (LBE), and we provide an empirical version of the LBE that comes with a bound on its bias in terms of the regularization parameters used in `Q-REPS`. Furthermore, we propose a semi-empirical version of the LBE that is unbiased thanks to the usage of a simulator. We also provide a practical implementation of the algorithm based on saddle-point optimization, and an error propagation analysis that shows convergence of the algorithm under some conditions. The analysis presented in this thesis differs from the one in the original paper [Bas-Serrano et al., 2021] because some of the results provided there were incorrect due to errors in the proofs. In this chapter we develop a new analysis fixing those errors at the price of introducing some more restrictive assumptions. The results are supported by a set of experiments testing the performance of `Q-REPS` in different standard environments, and comparing them with the performance of some state-of-the-art on-policy algorithms. The experiments show great performance of `Q-REPS` that in all cases is comparable or outperforms the competing algorithms.

1.7 Notation

We denote inner products over vector spaces by $\langle \cdot, \cdot \rangle$. The set of probability distributions on the finite set $\mathcal{S}$ is denoted as $\mathcal{P}_{\mathcal{S}} = \left\{ p \in \mathbb{R}_+^{\mathcal{S}} : \sum_{s \in \mathcal{S}} p(s) = 1 \right\}$, or just $\mathcal{P}$ if the set ξ is clear by context. Sums spanning over the spaces $x \in \mathcal{X}$ and $a \in \mathcal{A}$ are simply denoted by $\sum_x$ or $\sum_a$, and we write $p(x, a) \propto q(x, a)$ to signify that $p(x, a) = q(x, a) / \sum_{x', a'} q(x', a')$ for a nonnegative function q over $\mathcal{X} \times \mathcal{A}$. We denote by $\mathbf{1}_N$ the N-dimensional vector with all the entries equal to 1, or just $\mathbf{1}$ when N is clear by the context.

Chapter 2

CONVEX OPTIMIZATION

This chapter serves as an overview of the concepts about convex optimization that are used in the rest of the work. The presented results are quite standard and we make an informal presentation. For a more rigorous treatment one can check Boyd et al., 2004, Chapters 1 to 6 for Sections 2.1 to 2.3 and Bubeck, 2014, Chapters 4 and 5 for Section 2.4.

2.1 Basic concepts

Let's start from the beginning: presenting *convex sets*, *convex functions*, and some useful definitions and properties regarding them. We say that a set $\mathcal{S}$ is convex if for all $x, y \in \mathcal{S}$ and for all $\alpha \in (0, 1)$

$$\alpha x + (1 - \alpha)y \in \mathcal{S}.$$

We also introduce the following definitions regarding a set $\mathcal{S}$:

- The *affine hull* of a set $\mathcal{S}$, denoted as aff$(\mathcal{S})$, is the set of all affine combinations of points in $\mathcal{S}$:

$$\text{aff}(\mathcal{S}) = \{\theta_1 x_1 + \cdots \theta_k x_k | x_1, .., x_k \in \mathcal{S}, \theta_1 + \cdots + \theta = 1\}.$$

- The *interior* of a set $\mathcal{S}$, denoted as int$(\mathcal{S})$, is defined as follows: Let's define the ball of radious $\epsilon > 0$ and centered in $x \in \mathcal{S}$ as $B_\epsilon(x) = \{y : \|x - y\|_2 \leq \epsilon\} \subset \mathcal{S}$. Then $x \in \mathcal{S}$ is an *interior point* of $\mathcal{S}$ if there exists an ϵ such that $B_\epsilon(x) \subset \mathcal{S}$, and $int(\mathcal{S}) = \{x \in \mathcal{S} : x$ is an interior point$\}$.

- The *relative interior* of a set $\mathcal{S}$, denoted as relint$(\mathcal{S})$, is the interior of $\mathcal{S}$ relative to aff$(\mathcal{S})$:

$$\text{relint}(\mathcal{S}) = \{x \in \mathcal{S} | B_\epsilon(x) \cap \text{aff}(\mathcal{S}) \subseteq \mathcal{S} \text{ for some } \epsilon \geq 0\}.$$

- The *boundary* of a set $\mathcal{S}$ denoted as $\partial\mathcal{S}$ is the set of points in $\mathbb{R}^d \setminus \mathcal{S}$ such that for all $\epsilon > 0$ the set $B_\epsilon(x)$ contains points from $\mathcal{S}$ and $\mathcal{S}^c = \mathbb{R}^d \setminus \mathcal{S}$.
- Open and closed sets: A set $\mathcal{S}$ is open if $int(\mathcal{S}) = \mathcal{S}$, and is closed if its complement $\mathcal{S}^c$ is open.

Regarding convex functions, we say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $x, y \in \text{dom}(f)$ and for all $\alpha \in (0, 1)$

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) - (1-\alpha)f(y). \tag{2.1}$$

If the above condition is satisfied with strict inequality whenever $x \neq y$, we then say that the function is strictly convex. We say that f is concave if $-f$ is convex, and strictly concave if $-f$ is strictly convex.

Let's now define the dual norm of a norm $\|\cdot\|$ as

$$\|g\|_* = \sup_{x\in\mathbb{R}^d:\|x\|\leq 1} \langle g, x\rangle \,.$$

For example, if the norm is the ℓ_1 norm, then the dual norm is the ℓ_∞ norm and vice-versa, and if the norm is the ℓ_2 norm, the dual norm is itself. For a convex function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$, we say that f is

- L-Lipschitz w.r.t. $\|\cdot\|$ if $|f(x) - f(y)| \leq L\|x - y\|$.
- β-smooth w.r.t. $\|\cdot\|$ if $\|\nabla f(x) - \nabla f(y)\|_* \leq \beta\|x - y\|$.
- σ-strongly convex w.r.t. $\|\cdot\|$ if $f(x) - f(y) \leq \langle\nabla f(x), x - y\rangle - \frac{\sigma}{2}\|x - y\|^2$ or equivalently $\langle\nabla f(x) - \nabla f(y), x - y\rangle \geq \sigma\|x - y\|^2$.

2.2 Constrained optimization problems

Let's write the problem of minimizing a function $f(x)$ with $x \in \mathbb{R}^d$ subject to a set of inequality constraints $g_i(x) \leq 0$ for $i = 1, ..., m$ and a set of equality constrains $h_j(x) = 0$ for $j = 1, ..., n$ as

$$\begin{aligned} \text{minimize}_{x\in\mathcal{D}} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, ..., m \\ & h_j(x) = 0 \quad j = 1, ..., n, \end{aligned} \tag{2.2}$$

where we assume that $\mathcal{D} = \text{dom}(f) \cap \bigcap_i \text{dom}(g_i) \cap \bigcap_j \text{dom}(h_j)$ is nonempty. We say that a point $x \in \mathcal{D}$ is feasible if it satisfies the set of constraints, and we say that the optimization problem is feasible if there exists at least one feasible point. The feasible set $\mathcal{C}$ is the set of all feasible points.

The optimization problem 2.2 is a *convex optimization problem* if the following conditions hold:

- The objective function $f(x)$ is convex.

- The inequality constraints $g_i(x) \leq 0$ are convex.

- The equality constraints are affine $h_j(x) = a_j^\intercal x + b_j$.

Furthermore, if all the objective function, the inequality constraints and the equality constraints are affine, then we call it a *linear program* (LP):

$$\begin{aligned} \text{minimize}_{x \in \mathcal{D}} \quad & \langle c, x \rangle + d \\ \text{s.t.} \quad & Gx \leq h \\ & Ax = b. \end{aligned} \tag{2.3}$$

In this work, we will mainly work with convex (sometimes also linear) problems, which will allow us to use tools from convex optimization.

The following lemma presents an important result regarding the characteristics of the solution of convex optimization problems :

Proposition 2.2.1. (First-order optimality condition, Bubeck, 2014, Proposition 1.3) *Let f be convex and $\mathcal{C}$ a closed convex set on which f is differentiable. Then*

$$x^* \in \underset{x \in \mathcal{C}}{\mathit{arg\,min}}\, f(x)$$

if and only if

$$\langle x^* - x, \nabla f(x^*) \rangle \leq 0 \quad \forall x \in \mathcal{C}. \tag{2.4}$$

2.3 Lagrangian duality

Lagrangian duality (or just duality) is a principle that says that optimization problems can be viewed from two different prespectives: the *primal* problem, which is the original constrained optimization problem that we want to solve, and the *dual* problem, a problem related to the primal in a very special way that we explain below. The concept of duality and its implications are extremely important in optimization (specially in convex optimization), and as we will see in later chapters, duality is one of the key tools of this work. There, in order to develop algorithms and analyze them, we will navigating between the primal and the dual and take advantage of both.

Consider an optimization problem (not necessarily convex) in the form of (2.2). Its *Lagrangian* $\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is obtained by summing to the objective function a wheighted sum of the constraint functions $g_{1,...,m}(x)$ and $h_{1,...,n}(x)$:

$$\mathcal{L}(x,\lambda,\nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{n} \nu_j h_j(x), \tag{2.5}$$

with $\text{dom}(\mathcal{L}) = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^n$ and where the weights $\lambda_1, ..., \lambda_m$ and $\nu_1, ..., \nu_n$ are variables of the Lagrangian function called *dual variables* or *Lagrange multipliers*. To gain some intuition about this new function, let's take a look at the following min-max problem:

$$\min_{x\in\mathbb{R}^d} \max_{\lambda\in(\mathbb{R}^+)^m, \nu\in\mathbb{R}^n} \mathcal{L}(x,\lambda,\nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{n} \nu_j h_j(x). \tag{2.6}$$

where we have imposed the Lagrange multipliers of the inequality constraints to be positive. Thinking about this problem as a min-max game, we can easily see that if the constraints in (2.2) are not satisfied, then the $\max$ player can assign values to λ and ν to make the Lagrangian arbitrarily large. It is obvious that the $\min$ player does not want this to happen, and since he is the first one playing, the best strategy for him will always be to satisfy the constraints. This observation shows that the solution of problems (2.2) and (2.6) are the same. So far this is not very helpful since solving (2.6) is a hard problem. Nevertheless, under some conditions the order of the $\min$ an the $\max$ can be swapped, which opens very interesting possibilities. Once we have given some intuition about the results that we are going to present, we can move to a more formal explanation.

We start with the definition of the *Lagrange dual function* $\mathcal{G} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{G}(\lambda,\nu) = \inf_{x\in\mathcal{D}} \mathcal{L}(x,\lambda,\nu). \tag{2.7}$$

Since the Lagrange dual function is the pointwise infimum of affine functions of λ and ν, it is always a concave function.

Proposition 2.3.1. *Let x^* be an optimal point of the primal problem. Then, for all (λ,ν) with $\lambda \geq 0$, we have that $\mathcal{G}(\lambda,\nu)$ is a lower bound on the optimal value of the primal problem:*

$$\mathcal{G}(\lambda,\nu) \leq f(x^*).$$

Proof. This can be easily verified by realizing that for any feasible point $\widetilde{x}$, we have

$$\mathcal{L}(\widetilde{x},\lambda,\nu) = f(\widetilde{x}) + \sum_{i=1}^{m} \lambda_i g_i(\widetilde{x}) + \sum_{j=1}^{n} \nu_j h_j(\widetilde{x}) \leq f(\widetilde{x}),$$

since the terms $\lambda_i g_i(\widetilde{x})$ are negative for $i = 1, ..., m$ and the terms $\nu_j h_j(\widetilde{x}) \leq f(\widetilde{x})$ are zero for $j = 1, ..., n$. Hence

$$\mathcal{G}(\lambda, \nu) = \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) \leq \mathcal{L}(\widetilde{x}, \lambda, \nu) \leq f(\widetilde{x}).$$

Since this last inequality holds for any feasible point $\widetilde{x}$, it also holds for the optimal x^*. □

To find the values of λ and ν for which the bound is more tight (i.e. where the dual $\mathcal{G}$ takes the maximum value), one has to solve the Lagrange dual problem (usually called the *dual*):

$$\begin{aligned} \text{maximize} \quad & \mathcal{G}(\lambda, \nu) \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{2.8}$$

The fact that the solution of the dual is a lower bound on the solution of the primal is known as *weak duality*. In this work, we will actually use a stronger but less general result called *strong duality*, that implies that the optimal value of the primal equals the optimal value of the dual. The following proposition shows two sufficient conditions for strong duality to hold.

Proposition 2.3.2. *(*Slater's condition*) When the primal is a convex problem and there exists an $x \in$ relint$(\mathcal{D})$ such that*

$$\begin{aligned} g_i(x) < 0 \qquad & i = 1, ..., m \\ h_j(x) = 0 \qquad & j = 1, ..., n \end{aligned}$$

then strong duality holds.

If any of the inequality constraints g_i is affine, the Slater's condition can be refined to only require non-strict feasibility for that constraint: $g_i(x) \leq 0$. There are many other results that establish conditions on the problem beyond convexity under which strong duality holds. Those conditions are known as *constraint qualifications*.

Going back to the intuition that brought equation (2.6), strong duality is equivalent to saying that the order of the $\min$ and the $\max$ players can be switched, i.e., that

$$\min_{x \in \mathcal{D}} \max_{\lambda \in (\mathbb{R}^+)^m, \nu \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) = \max_{\lambda \in (\mathbb{R}^+)^m, \nu \in \mathbb{R}^n} \min_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu).$$

This result is very closely connected to a more general result presented in Sion's minimax Theorem, that states that for (most) continuous functions $f(x, y)$ that are convex in the first argument x on a compact convex set $\mathcal{X}$, and concave in the second argument y on a convex set $\mathcal{Y}$, it holds that

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y).$$

The following proposition shows another very interesting result regarding the relation between the dual variables λ_i and the equality constraints $g_i(x)$:

Proposition 2.3.3. *(*Complementary slackness*) Let x^* be an optimal point of the primal problem and (λ^*, ν^*) be optimal points of the dual problem. Then, for $i = 1, ..., m$, whether $\lambda_i^* = 0$ or $g_i(x^*) = 0$.*

Proof. Let x^* be an optimal point of the primal problem and (λ^*, ν^*) the optimal point of the dual problem. Let's also assume that strong duality holds. Then, we can write the following:

$$\begin{aligned}
f(x^*) &= \mathcal{G}(\lambda^*, \nu^*) \\
&= \inf_x \left(f(x) + \sum_{i=1}^m \lambda_i^* g_i(x) + \sum_{j=1}^n \nu_j^* h_j(x) \right) \\
&\leq f(x^*) + \sum_{i=1}^m \lambda_i^* g_i(x^*) + \sum_{j=1}^n \nu_j^* h_j(x^*) \\
&\leq f(x^*).
\end{aligned}$$

Where the first equality is due to strong duality and the last inequality is because the terms that disappear are all negative or zero. In the above derivation we can see that both inequalities have to hold with equality. For the first inequality, this implies that x^* minimizes $\mathcal{L}(x, \lambda^*, \nu^*)$ over x. For the second inequality, the implication is that

$$\sum_{i=1}^m \lambda_i^* g_i(x^*) = 0.$$

Since all the terms of the above sum are nonpositive, this implies that $\lambda_i^* g_i(x^*) = 0$ for $i = 1, ..., m$, which concludes the proof. □

Another important result in Lagrangian duality theory is presented in the following proposition that establish necessary (and sufficient under some conditions) optimality conditions:

Proposition 2.3.4. *(*Karush–Kuhn–Tucker (KKT) conditions*, Boyd et al., 2004, Section 5.5.3.) Let x^* be an optimal point of the primal problem and (λ^*, ν^*) the optimal point of the dual problem. Assume also that strong duality holds. Then, for all $i = 1, ..., m$ and $j = 1, ..., n$, the following conditions are satisfied:*

$$
\begin{aligned}
&g_i(x^*) \leq 0 \\
&h_j(x^*) = 0 \\
&\lambda_i \geq 0 \\
&\lambda_i g_i(x^*) = 0 \\
&\nabla_x f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla_x g_i(x^*) + \sum_{j=1}^{n} \nu_j^* \nabla_x h_j(x^*) = 0
\end{aligned}
$$

If the primal problem is convex, the KKT conditions become also suficient, meaning that if a set of points x, λ, ν satisfy the KKT conditions, then x is primal optimal and λ, ν are dual optimal.

It is interesting to see that the last condition comes from the fact that the gradient of $L(x, \lambda^*, \nu^*)$ w.r.t. x must vanish at $x = x^*$.

2.4 Algorithms

The objective of this section is *not* to make an overview of existing convex optimization algorithms nor a review of the state of the art. The aim is to introduce the foundations to understand mirror descent and other algorithms based on the same principles, since the algorithms presented in this work are closely related to them.

Bregman divergence and Legendre functions

We start presenting a tool that will constitute a key part of the algorithms presented in this chapter and the rest of this work: the *Bregman divergence*. For a convex function $\Phi : \mathbb{R}^d \to \mathbb{R}$ with $\mathcal{D} = \text{dom}(\Phi)$, the Bregman divergence between $x \in \mathbb{R}^d$ and $y \in \mathcal{D}$ is defined as

$$D_\Phi(x\|y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y \rangle \tag{2.9}$$

Intuitively, a Bregman divergence gives us a measure of difference between two points x and y with respect to some function Φ. In this sense, Bregman divergences are similar to distances where the function Φ acts as the distance generating norm. This analogy is useful from the intuitive point of view, but Bregman divergences are not distances since they are not symmetric (in general) nor satisfy the triangle inequality.

Geometrically, $D_\Phi(x\|y)$ can be understood as the distance between $\Phi(x)$ and the Taylor approximation of $\Phi(x)$ from y, and gives us a notion of how convex the function Φ is between the points x and y: if the function is very flat between the two points, then the Taylor approximation at x will be "good" and the Bregman divergence will be small, and if the function is very convex, the approximation will be "bad" so we will have a large Bregman divergence. We will often denote D_Φ as D when Φ is clear by the context.

Here we show some useful properties of the Bergman divergence $D_\Phi(x\|y)$:

- Is *strictly convex* in his first argument x.

- Is *nonnegative*, meaning that $D_\Phi(x\|y) \geq 0$, and $D_\Phi(x\|y) = 0$ if an only if $x = y$.

- Is *linear* in Φ: $D_{\Phi + a\Phi'}(x\|y) = D_\Phi(x\|y) + aD_{\Phi'}(x\|y)$.

- The gradient of $D_\Phi(x\|y)$ w.r.t. x satisfies $\frac{\partial D(x\|y)}{\partial x} = \nabla\Phi(x) - \nabla\Phi(y)$.

We also present a very useful equality known as the three points identity:

$$D(x\|y) + D(z\|x) - D(z\|y) = \langle \nabla\Phi(x) - \nabla\Phi(y), x - z \rangle . \tag{2.10}$$

To gain some insight about it we can realize that $D(z\|x)$ is the distance between $\Phi(z)$ and the Taylor approximation of $\Phi(z)$ from x, and $D(z\|y) - D(x\|y) = \Phi(z) - \nabla\Phi(y)(z - x)$ is the distance between $\Phi(z)$ and the Taylor approximation of $\Phi(z)$ but from y instead of x. Therefore, the term in the right hand side of (2.10), is exactly the difference between using the gradient in y instead of x.

As a direct consequence of the three points identity together with Proposition 2.2.1, we have another useful result known as the generalized Pythagorean theorem for Bregman divergences that says the following: let $\mathcal{C}$ be a convex set, $y \in \mathcal{C}$, and $x = \arg\min_{x \in \mathcal{C}} D(x\|x')$ be the projection of a point x' in $\mathcal{C}$. Then,

$$D(x\|x') + D(y\|x) - D(y\|x') \leq 0. \tag{2.11}$$

We say that the convex function Φ is a *Legendre function* if:

- int($\mathcal{D}$) is non-empty,

- Φ is differentiable and strictly convex in int($\mathcal{D}$), and

- $\lim_{n\to\infty} \|\nabla\Phi(x_n)\| = \infty$ for any sequence of $\{x_n\}_n$ with $x_n \in \mathcal{C}$ for all n and $\lim_{n\to\infty} x_n$ is in the boundary of int($\mathcal{D}$).

Roughly speaking, a Legendre function is a strongly convex function whose gradient blows up in the boundaries of its domain. Defining the *Fenchel conjugate* of a function Φ as $\Phi^*(u) = \sup_x \langle x, u \rangle - \Phi(x)$, we have that if Φ is Legendre, the following statements are true:

- The Fenchel conjugate Φ^* is Legendre.

- $\nabla\Phi$ is a bijection between $\mathrm{int}(\mathrm{dom}(\Phi))$ and $\mathrm{int}(\mathrm{dom}(\Phi^*))$ with $\nabla\Phi^{-1} = \nabla\Phi^*$.

- $D_\Phi(x, y) = D_{\Phi^*}(\nabla\Phi(y), \nabla\Phi(x))$ for all $x, y \in \mathrm{dom}(\Phi)$.

Theorem 2.4.1. *(Lattimore and Szepesvári, 2020, Theorem 26.15). Let $\Phi : \mathbb{R}^d \to R$ be Legendre, $\mathcal{C} \subset \mathbb{R}^d$ a non-empty, closed, convex set with $\mathcal{C} \cap dom(\Phi)$ non-empty, and assume that $\widetilde{x} = \arg\min_{x\in\mathbb{R}^d} \Phi(x)$ exists. Then,*

1. *$x^* = \arg\min_{x\in\mathcal{C}} \Phi(x)$ exists and in unique.*

2. *$x^* = \arg\min_{x\in\mathcal{C}} D_\Phi(z, \widetilde{x})$.*

Theorem 2.4.1 shows that minimizing a Legendre function Φ in a convex set $\mathcal{C}$ is equivalent to finding the unconstrained minimum in $\mathrm{dom}(\Phi)$ and projecting that point to $\mathcal{C}$. As we will see soon, this property is very useful in the implementation of the algorithms presented below.

2.4.1 Mirror Descent

Consider the constrained optimization problem $\arg\min_{x\in\mathcal{C}} f(x)$ with $f : \mathbb{R}^d \to \mathbb{R}$. *Mirror descent* (MD) needs two parameters which are a convex (and usually Legendre) function $\Phi : \mathbb{R}^d \to \mathbb{R}$ with domain $\mathcal{D}$ such that $\mathcal{C}$ is in its closure, and a learning rate $\eta \geq 0$. Then, at $k = 0$ MD proposes

$$x_0 \in \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} \Phi(x),$$

and for $k \geq 1$ it selects

$$x_k = \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} \langle\nabla f(x_{k-1}), x\rangle + \frac{1}{\eta} D_\Phi(x\|x_{k-1})$$

After K iterations, we define $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K} x_k$. With this procedure, mirror descent creates a sequence of points x_0, x_1, x_2... that (hopefully) converge to $x^* = \arg\min_{x\in\mathcal{C}} f(x)$.

At each iteration k, MD looks for a trade-off between minimizing the linearization of the function (go as far as possible in the direction of $-\nabla f(x)$), and not going too far from the previous point x_k in terms of the Bregman divergence induced by Φ.

The update performed by mirror descent at each time step k for $k \geq 1$ is equivalent to the following two-step procedure:

$$\nabla\Phi(y_k) = \nabla\Phi(x_{k-1}) - \eta\nabla f(x_{k-1})$$

and

$$x_k = \arg\min_{x \in \mathcal{C} \cap \mathcal{D}} D_\Phi(y_k \| x_{k-1}).$$

We find this alternative presentation of the algorithm less insightful but we add it here because it is used in the proof of the following theorem that gives a bound on the convergence rate of mirror descent:

Theorem 2.4.2. *(Bubeck, 2014, Theorem 4.2) Let Φ be a mirror map σ-strongly convex on $\mathcal{C} \cap \mathcal{D}$ w.r.t. $\|\cdot\|$. Let $R^2 = \sup_{x \in \mathcal{C} \cap \mathcal{D}} \Phi(x) - \Phi(x_0)$, and f be a convex and L-Lipschitz w.r.t. $\|\cdot\|$. Then, after K iterations, mirror descent with $\eta = \frac{R}{L}\sqrt{\frac{2}{\sigma T}}$ satisfies*

$$f\left(\bar{x}_K\right) - f(x^*) \leq RL\sqrt{\frac{2}{\sigma K}}$$

Proof. Let $x \in \mathcal{C} \cap \mathcal{D}$. Then, we have

$$\begin{aligned}
f(x_k) - f(x) &\leq \nabla \left\langle f(x_k), (x_k - x\right\rangle \\
&\quad \text{(due to convexity of } f) \\
&= \frac{1}{\eta}(\nabla\Phi(x_k) - \nabla\Phi(y_{k+1}))^\top (x_k - x) \\
&\quad \text{(by definition of mirror descent)} \\
&= \frac{1}{\eta}(D_\Phi(x \| x_k) + D_\Phi(x_k \| y_{k+1}) - D_\Phi(x \| y_{k+1})) \\
&\quad \text{(using the three points identity (2.10))} \\
&\leq \frac{1}{\eta}(D_\Phi(x \| x_k) + D_\Phi(x_k \| y_{k+1}) - D_\Phi(x \| x_{k+1}) - D_\Phi(x_{k+1} \| y_{k+1})) \\
&\quad \text{(using the generalized Pythagorean theorem (2.11))}
\end{aligned}$$

Summing over epochs some of the terms telescope and we get

$$\sum_{k=0}^{K} f(x_k) - f(x) \leq \frac{1}{\eta}\left(D(x \| x_0) + \sum_{k=0}^{K}\left(D_\Phi(x_k \| y_{k+1}) - D_\Phi(x_{k+1} \| y_{k+1})\right)\right). \tag{2.12}$$

It remains to bound the terms $D_\Phi(x_k \| y_{k+1}) - D_\Phi(x_{k+1} \| y_{k+1})$. To do so, we use the σ-strong convexity of Φ and the inequality $az - bz^2 \leq \frac{a^2}{4b}, \forall z \in \mathbb{R}$:

$$
\begin{aligned}
D_\Phi(x_k, y_{k+1}) - D_\Phi(x_{k+1}, y_{k+1}) &= \Phi(x_k) - \Phi(x_{k+1}) - \nabla\Phi(y_{k+1})^\top (x_k - x_{k+1}) \\
&\le (\nabla\Phi(x_k) - \nabla\Phi(y_{k+1}))^\top (x_k - x_{k+1}) - \frac{\sigma}{2}\left\|x_k - x_{k+1}\right\|^2 \\
&= \eta g_k^\top (x_k - x_{k+1}) - \frac{\sigma}{2}\left\|x_k - x_{k+1}\right\|^2 \\
&\le \eta L \left\|x_k - x_{k+1}\right\| - \frac{\sigma}{2}\left\|x_k - x_{k+1}\right\|^2 \\
&\le \frac{(\eta L)^2}{2\sigma}
\end{aligned}
$$

Plugging this result into (2.12) we get

$$\sum_{k=1}^{K}(f(x_k) - f(x)) \le \frac{D_\Phi(x, x_1)}{\eta} + \eta\frac{L^2 K}{2\sigma}.$$

Plugging the choice of η from the statement of the theorem concludes the proof.

□

2.4.2 Mirror prox

Mirror prox (MP) was first introduced by Nemirovski [2004] and can be seen like a more sophisticated version of mirror descent. As in mirror descent, to solve an optimization problem $\arg\min_{x\in\mathcal{C}\cap\mathcal{D}} f(x)$, mirror prox is initialized with a mirror map Φ with domain $\mathcal{D}$ such that $\mathcal{C}$ is in its closure and a learning rate η, and at $k = 0$

$$x_0 \in \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} \Phi(x).$$

For $k \ge 1$ is when mirror prox differs from mirror descent, since it performs the following two-step update:

$$\widehat{x}_k = \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} \langle\nabla f(x_{k-1}), x\rangle + \frac{1}{\eta} D_\Phi(x, x_{k-1}),$$

$$x_k = \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} \langle\nabla f(\widehat{x}_k), x\rangle + \frac{1}{\eta} D_\Phi(x, x_{k-1}).$$

After K iterations, we define $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K} \widehat{x}_{k+1}$. In the first step of this two-step update, mirror prox performs a regular step as mirror descent would do. But after this, mirror prox goes back to the previous point x_{k-1} and makes another step but with the gradient of $\widehat{x}_k$ instead of x_{k-1}. The first of these steps is often referred to as an *extrapolation step* and it serves to enhance the stability of the algorithm.

As in mirror descent, the update rule of mirror prox can also be rewritten in the following alternative way:

$$\begin{aligned}
&\nabla\Phi(\widehat{x}'_k) = \nabla\Phi(x_{k-1}) - \eta\nabla f(x_{k-1}) \\
&\widehat{x}_k = \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} D_\Phi(x\|\widehat{x}'_k)
\end{aligned}$$

$$\begin{aligned}
&\nabla\Phi(x'_k) = \nabla\Phi(x_{k-1}) - \eta\nabla f(\widehat{x}_k) \\
&x_k = \arg\min_{x\in\mathcal{C}\cap\mathcal{D}} D_\Phi(x\|x'_k)
\end{aligned}$$

We now present the following theorem and subsequent corollaries that show interesting results regarding the convergence of mirror prox:

Theorem 2.4.3. *(Rakhlin and Sridharan, 2013, Lemma 1) Let Φ be σ-strongly convex and ∇f be L-Lipschitz. Then, for all k, Mirror Prox guarantees*

$$\eta\left\langle \widehat{x}_{k+1} - x, \nabla f(\widehat{x}_{k+1})\right\rangle \leq D(x\|x_k) - D(x\|x_{k+1}) - \frac{\sigma - \eta L}{4}\left\|x_{k+1} - x_k\right\|^2.$$

holds for every $x \in \mathcal{C}\cap\mathcal{D}$ and $k > 0$.

Proof. The proof will rely on repeatedly using the three points identity (2.10). We first use it to show

$$\begin{aligned}
D(x\|x_{k+1}) &= D(x\|x_k) - D(x_{k+1}\|x_k) + \left\langle x_{k+1} - x, \nabla\Phi(x_{k+1}) - \nabla\Phi(x_k)\right\rangle \\
&\leq D(x\|x_k) - D(x_{k+1}\|x_k) + \eta\left\langle x - x_{k+1}, \nabla f(\widehat{x}_{k+1})\right\rangle,
\end{aligned}$$

where we also used the first-order optimality condition for x_{k+1} in the second step:

$$\left\langle \nabla\Phi(x_k) - \nabla\Phi(x_{k+1}) - \eta\nabla f(\widehat{x}_{k+1}), x_{k+1} - x\right\rangle \geq 0.$$

Furthermore, we have

$$\left\langle x - x_{k+1}, \nabla f(\widehat{x}_{k+1})\right\rangle = \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1})\right\rangle + \left\langle \widehat{x}_{k+1} - x_{k+1}, \nabla f(\widehat{x}_{k+1})\right\rangle.$$

Using this bound together with the three-points identity

$$\begin{aligned}
D(x_{k+1}\|x_k) =& D(x_{k+1}\|\widehat{x}_{k+1}) + D(\widehat{x}_{k+1}\|x_k) \\
&+ \left\langle \nabla\Phi(x_k) - \nabla\Phi(\widehat{x}_{k+1}), \widehat{x}_{k+1} - x_{k+1}\right\rangle,
\end{aligned}$$

we obtain

$$\begin{aligned}D(x\|x_{k+1}) \leq & D(x\|x_k) - D(x_{k+1}\|x_k) + \eta \left\langle \widehat{x}_{k+1} - x_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ & + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ =& D(x\|x_k) - D(x_{k+1}\|\widehat{x}_{k+1}) - D(\widehat{x}_{k+1}\|x_k) + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ & + \left\langle \nabla\Phi(x_k) - \nabla\Phi(\widehat{x}_{k+1}) - \eta \nabla f(\widehat{x}_{k+1}), x_{k+1} - \widehat{x}_{k+1} \right\rangle \\ =& D(x\|x_k) - D(x_{k+1}\|\widehat{x}_{k+1}) - D(\widehat{x}_{k+1}\|x_k) \\ & + \left\langle \nabla\Phi(x_k) - \nabla\Phi(\widehat{x}_{k+1}) - \eta \nabla f(x_k), x_{k+1} - \widehat{x}_{k+1} \right\rangle \\ & + \eta \left\langle \nabla f(x_k) - \nabla f(\widehat{x}_{k+1}), x_{k+1} - \widehat{x}_{k+1} \right\rangle + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ \leq& D(x\|x_k) - D(x_{k+1}\|\widehat{x}_{k+1}) - D(\widehat{x}_{k+1}\|x_k) \\ & + \eta \left\langle \nabla f(x_k) - \nabla f(\widehat{x}_{k+1}), x_{k+1} - \widehat{x}_{k+1} \right\rangle + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle,\end{aligned}$$

where the last step follows from the fact that $\widehat{x}_{k+1}$ satisfies the first-order optimality condition

$$\left\langle \nabla\Phi(x_k) - \nabla\Phi(\widehat{x}_{k+1}) - \eta \nabla f(x_k), x_{k+1} - \widehat{x}_k \right\rangle \leq 0.$$

Now, using the σ-strong convexity of Φ and the L-Lipschitz continuity of ∇f, we obtain

$$\begin{aligned}D(x\|x_{k+1}) \leq & D(x\|x_k) - D(x_{k+1}\|\widehat{x}_{k+1}) - D(\widehat{x}_{k+1}\|x_k) \\ & + \eta \left\langle \nabla f(x_k) - \nabla f(\widehat{x}_{k+1}), x_{k+1} - \widehat{x}_{k+1} \right\rangle + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ \leq& D(x\|x_k) - \frac{\sigma}{2} \left\| x_{k+1} - \widehat{x}_{k+1} \right\|_2^2 - \frac{\sigma}{2} \left\| \widehat{x}_{k+1} - x_k \right\|_2^2 \\ & + \eta L \left\| x_k - \widehat{x}_{k+1} \right\|_2 \left\| x_{k+1} - \widehat{x}_{k+1} \right\|_2 + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ \leq& D(x\|x_k) - \frac{\sigma - \eta L}{2} \left(\left\| x_{k+1} - \widehat{x}_{k+1} \right\|_2^2 + \left\| \widehat{x}_{k+1} - x_k \right\|_2^2 \right) \\ & + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle \\ \leq& D(x\|x_k) - \frac{\sigma - \eta L}{4} \left\| x_{k+1} - x_k \right\|_2^2 + \eta \left\langle x - \widehat{x}_{k+1}, \nabla f(\widehat{x}_{k+1}) \right\rangle,\end{aligned}$$

where we also used the elementary inequalities $2ab \leq a^2 + b^2$ and $(a+b)^2 \leq 2a^2 + 2b^2$ in the last two steps, respectively. □

This theorem has two important corollaries that we will crucially use throughout the analysis of the algorithm presented in Chapter 4. The first one shows that the iterates remain bounded during the optimization procedure.

Corollary 2.4.1. *Suppose that the conditions of Theorem 2.4.3 hold and that $\eta \leq \frac{\sigma}{L}$. Then, for all k, Mirror Prox guarantees*

$$D(x^*\|x_{k+1}) \leq D(x^*\|x_k).$$

In particular, $D(x^*\|x_k) \leq D(x^*\|x_0)$ for all k.

Proof. Applying Theorem 2.4.3 with x^* being an optimal solution to the minimization problem, we get

$$\eta \langle \widehat{x}_{k+1} - x^*, \nabla f(\widehat{x}_{k+1}) \rangle + D(x^* \| x_{k+1}) \leq D(x^* \| x_k) - \frac{\sigma - \eta L}{4} \|x_{k+1} - x_k\|^2 .$$

Since x^* satisfies the variational inequality $\langle \widehat{x}_{k+1} - x^*, \nabla f(\widehat{x}_{k+1}) \rangle \geq 0$ and that the right-most term is positive, we get

$$D(x^* \| x_{k+1}) \leq D(x^* \| x_k),$$

which trivailly concludes the proof. □

The next corollary establishes a bound on the suboptimality gap evaluated at $\bar{x}_K$:

Corollary 2.4.2. *Let $x \in \mathcal{C} \cap \mathcal{D}$ be arbitrary and assume that the conditions of Theorem 2.4.3 hold and that $\eta \leq \frac{\sigma}{L}$. Then, mirror prox guarantees the following bound on the duality gap:*

$$f(\bar{x}_K) - f(x) \leq \frac{D(x \| x_0)}{\eta K}.$$

Proof.

$$\begin{aligned} f(\bar{x}_K) - f(x) &= f\left(\frac{1}{K} \sum_{k=0}^{K} \widehat{x}_{k+1} \right) - f(x) \\ &\leq \sum_{k=0}^{K} \frac{1}{K} \left(f(\widehat{x}_{k+1}) - f(x) \right) \\ &\leq \sum_{k=0}^{K} \frac{1}{K} \langle \nabla f(\widehat{x}_{k+1}), \widehat{x}_{k+1} - x \rangle \\ &\leq \frac{1}{\eta K} \sum_{k=0}^{K} \left(D(x \| x_k) - D(x \| x_{k+1}) - \frac{\sigma - \eta L}{4} \|x_{k+1} - x_k\|^2 \right) \\ &\leq \frac{1}{\eta K} D(x \| x_0), \end{aligned}$$

where in the first inequality we used Jensen's inequality, in the second one we used convexity and in the third one we used the bound in Theorem 2.4.3. □

2.4.3 Saddle-point optimization

Let $\mathcal{X}$ and $\mathcal{Y}$ be compact convex sets, and $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ a function that is convex in his first argument x and concave in his second argument y. Let's now consider the following min-max or saddle-point problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y).$$

We are interested in finding the optimal (x^*, y^*). The quality of a candidate solution (x, y) is measured through the duality gap defined as

$$\max_{y' \in \mathcal{Y}} f(x, y') - \min_{x' \in \mathcal{X}} f(x', y),$$

that by deffinition is equal to 0 at (x^*, y^*).

From now on, we use the notation $z = (x, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Furthermore, we assume that $\mathcal{X}$ is equipped with a mirror map $\Phi_{\mathcal{X}}$ (with $\mathcal{D}_{\mathcal{X}} = \mathrm{dom}(\Phi_{\mathcal{X}})$) which is 1-strongly convex w.r.t. a norm $\| \cdot \|_{\mathcal{X}}$ on $\mathcal{X} \cap \mathcal{D}_{\mathcal{X}}$, and we denote $R^2_{\mathcal{X}} = \sup_{x \in \mathcal{X}} \Phi(x) - \min_{x \in \mathcal{X}} \Phi(x)$. We define similar quantities for the space $\mathcal{Y}$. To solve the saddle-point problem, we can apply the mirror descent (or mirror prox) scheme in the space $\mathcal{Z}$ by using:

- The mirror map $\Phi(z) = a\Phi_{\mathcal{X}}(x) + b\Phi_{\mathcal{Y}}(y)$ with $a, b \in \mathbb{R}_+$ and defined on $\mathcal{D} = \mathcal{D}_{\mathcal{X}} \times \mathcal{D}_{\mathcal{Y}}$.
- The monotone operator $g(z) = (\nabla_x f(x, y), -\nabla_y f(x, y))$ instead of the real gradient.

We can realize that using this operator, the duality gap can be controlled in a similar way as the subobtimality gap in a convex optimization problem:

$$\max_{y' \in \mathcal{Y}} f(x, y') - \min_{x' \in \mathcal{X}} f(x', y) \leq \langle g(z), z - z' \rangle \tag{2.13}$$

where $z' = (x', y')$. For the *saddle-point mirror descent* (SP-MD) algorithm, we let $z_0 \in \arg\min_{z \in \mathcal{Z} \cap \mathcal{D}} \Phi(z)$ and for $k \geq 1$ we have

$$z_k \in \arg\min_{z \in \mathcal{Z} \cap \mathcal{D}} \eta \langle g(z_{k-1}), z \rangle + D_{\Phi}(z, z_{k-1}).$$

We define $\bar{z}_K = (\bar{x}_K, \bar{y}_K) = \frac{1}{K} \left(\sum_{k=0}^{K} x_k, \sum_{k=0}^{K} y_k \right)$. The following theorem gives a bound on the duality gap of $\bar{z}_K$:

Theorem 2.4.4. *(Bubeck, 2014, Theorem 5.1) Assume that $f(\cdot, y)$ is $L_{\mathcal{X}}$-Lipschitz w.r.t. the norm $\| \cdot \|_{\mathcal{X}}$, that is $\|\nabla_y f(x, y)\|^*_{\mathcal{X}} \leq L_{\mathcal{X}}, \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$. Similarly* *assume that $f(x,\cdot)$ is $L_{\mathcal{Y}}$-Lipschitz w.r.t. $\|\cdot\|_{\mathcal{Y}}$. Then SP-MD with $a = \frac{L_{\mathcal{X}}}{R_{\mathcal{X}}}, b = \frac{L_{\mathcal{Y}}}{R_{\mathcal{Y}}}$, and $\eta = \sqrt{\frac{2}{K}}$ satisfies*

$$\max_{y\in\mathcal{Y}} f\left(\frac{1}{K}\sum_{k=1}^{K} x_k, y\right) - \min_{x\in\mathcal{X}} f\left(x, \frac{1}{K}\sum_{k=1}^{K} y_k\right) \leq (R_{\mathcal{X}}L_{\mathcal{X}} + R_{\mathcal{Y}}L_{\mathcal{Y}})\sqrt{\frac{2}{K}}.$$

Proof. The proof of this theorem follow the same lines as the one from Theorem 2.4.2. We give a sketch of how to make it work. We start defining the norm

$$\|z\|_{\mathcal{Z}} = \sqrt{a\,\|x\|_{\mathcal{X}}^2 + b\,\|y\|_{\mathcal{Y}}^2}.$$

We can easily see that the mirror map Φ is 1-strongly convex w.r.t. $\|\cdot\|_{\mathcal{Z}}$. Furthermore, the dual norm of $\|\cdot\|_{\mathcal{Z}}^2$ can be shown to be

$$\|z\|_{\mathcal{Z}}^* = \sqrt{\frac{1}{a}(\|x\|_{\mathcal{X}}^*) + \frac{1}{b}(\|y\|_{\mathcal{Y}}^*)}.$$

Thus, we have that

$$\|g\|_{\mathcal{Z}}^* \leq \sqrt{\frac{L_{\mathcal{X}}^2}{a} + \frac{L_{\mathcal{Y}}^2}{b}}.$$

The theorem can now be proved in the same way as Theorem 2.4.2 by starting from equation (2.13) and using the above observations . □

Theorem 2.4.5 gives a similar result for the *saddle-point mirror prox* (SP-MP) algorithm, that can be defined following the same ideas. The presented result is regarding smooth functions. We say that a function $f(x,y)$ is $(\beta_{11}, \beta_{12}, \beta_{22}, \beta_{21})$-smooth if for any $x, x' \in \mathcal{X}, y, y' \in \mathcal{Y}$,

$$\begin{aligned}
\|\nabla_x f(x,y) - \nabla_x f(x',y)\|_{\mathcal{X}}^* &\leq \beta_{11}\,\|x - x'\|_{\mathcal{X}}\,, \\
\|\nabla_x f(x,y) - \nabla_x f(x,y')\|_{\mathcal{X}}^* &\leq \beta_{12}\,\|y - y'\|_{\mathcal{Y}}\,, \\
\|\nabla_y f(x,y) - \nabla_y f(x,y')\|_{\mathcal{Y}}^* &\leq \beta_{22}\,\|y - y'\|_{\mathcal{Y}}\,, \\
\|\nabla_y f(x,y) - \nabla_y f(x',y)\|_{\mathcal{Y}}^* &\leq \beta_{21}\,\|x - x'\|_{\mathcal{X}}\,.
\end{aligned}$$

Theorem 2.4.5. *(Bubeck, 2014, Theorem 5.2) Assume that f is $(\beta_{11}, \beta_{12}, \beta_{22}, \beta_{21})$-smooth. Then, defining $C = \max(\beta_{11}R_{\mathcal{X}}^2, \beta_{22}R_{\mathcal{Y}}^2, \beta_{12}R_{\mathcal{X}}R_{\mathcal{Y}}, \beta_{21}R_{\mathcal{X}}R_{\mathcal{Y}})$, SP-MP with $a = \frac{1}{R_{\mathcal{X}}^2}, b = \frac{1}{R_{\mathcal{Y}}^2}$, and $\eta = \frac{1}{2C}$ satisfies*

$$\max_{y\in\mathcal{Y}} f\left(\frac{1}{K}\sum_{k=1}^{K} x_{k+1}, y\right) - \min_{x\in\mathcal{X}} f\left(x, \frac{1}{K}\sum_{k=1}^{K} y_{k+1}\right) \leq \frac{4C}{K}$$

Proof. Similarly as done in the proof of Theorem 2.4.4, we only need to define the norm

$$\|z\|_{\mathcal{Z}} = \sqrt{\frac{1}{R_{\mathcal{X}}} \|x\|_{\mathcal{X}}^2 + \frac{1}{R_{\mathcal{Y}}} \|y\|_{\mathcal{Y}}^2}$$

and realize that $g(z)$ is β-Lipschitz w.r.t. $\|\cdot\|_{\mathcal{Z}}$ with $\beta = 2C$ and we can then follow the same procedure as in the proof of Theorem 2.4.3 and Corollary 2.4.2. □

Chapter 3

REINFORCEMENT LEARNING

In this chapter we give an introduction of the reinforcement learning (RL) problem, introduce the different settings that we will consider during this work, and show the main tools for evaluating and improving policies. The presented result are standard and well known so we will make a fast overview. For a more detailed description one can check Puterman, 1994, Chapters 6 and 8 for Sections 3.1, 3.2 and 3.3, and Bertsekas [2008] and Buşoniu et al. [2012] for Section 3.4.

3.1 Markov Decision Process

Mathematically, the RL problem is formalized using the Markov Decision Process framework (MDP, Puterman [1990]). An MDP is defined by a tuple $M = (\mathcal{X}, \mathcal{A}, P, r)$, where

- $\mathcal{X}$ is the state space or set of states.
- $\mathcal{A}$ is the action space or set of actions.
- P is the transition function with $P(\cdot|x, a)$ denoting the distribution of the follow-up state x' after taking action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$.
- r is the reward function that maps state-action pairs to rewards, with $r(x, a)$ denoting the reward of being in state x and taking action a.

In this work we assume that the rewards are deterministic and bounded in $[0, 1]$, and that the state space and the action space are finite (but potentially very large).

An MDP models a sequential interaction process between an agent and its environment, where in each round t, the agent observes state $x_t \in \mathcal{X}$, selects action $a_t \in \mathcal{A}$, moves to the next state $x_{t+1} \sim P(\cdot|x_t, a_t)$, and obtains reward $r(x_t, a_t)$. The goal of the agent is to select actions so as to maximize some notion of cumulative reward. The strategy that the agent follows for picking actions is called policy and is denoted as π. In this work we consider *stochastic policies*, which are a mapping from states to probability distributions over actions. For a given policy π, we denote as $\pi(a|x)$ the probability of taking action a in state x. We say that a policy is optimal if it maximizes the chosen notion of cumulative reward, and we denote it as π^*.

The way how this cumulative reward is defined determines the goal of the optimization problem, and is key when developing algorithms and performing their analysis. In this work, we focus on two different settings known as *average reward* and *normalized discounted reward*, each of them corresponding to a different notion of cumulative reward. The next two sections are dedicated to this two settings.

3.2 Average reward setting

As its name indicates, the goal in this setting is to maximize the expected average reward defined as

$$\liminf_{t\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(x_t, a_t)\right].$$

From now on, we will always make the following assumption when considering the average reward setting:

Assumption 1 (Uniform ergodicity)**.** *Every policy π generates an ergodic Markov chain. Specifically, letting P_π be the transition operator of π defined as the matrix with elements $P_\pi(x'|x) = \sum_a \pi(a|x)P(x'|x,a)$, and ν, ν' be any two distributions over $\mathcal{X}$, the following inequality is satisfied for some $C, \tau > 0$ and for all k:*

$$\left\|\left(\nu - \nu'\right) P_\pi^k\right\|_1 \leq Ce^{-k/\tau} \left\|\nu - \nu'\right\|_1 .$$

We say that our MDP is *uniformly ergodic* if it satisfies Assumption 1. Notice that this assumption is significantly weaker than the 1-step mixing assumption often made in the related literature [Even-Dar et al., 2009; Neu et al., 2014]. It is easily shown to hold when all policies induce aperiodic and irreducible Markov chains—see Theorem 4.9 in Levin and Peres [2017] for a proof.

Under Assumption 1 stated above, each policy π generates a unique *stationary state distribution* $\nu^\pi \in \Delta_\mathcal{X}$ over the state space satisfying

$$\nu^\pi(x) = \lim_{t\to\infty} \mathbb{P}\left[x_t = x\right]$$

for all x when the trajectory $(x_0, x_1, ..., x_t)$ is generated by following policy π. Similarly, each policy π generates a *stationary state-action distribution* $\mu^\pi \in$

$\Delta_{\mathcal{X}\times\mathcal{A}}$ satisfying

$$\mu^\pi(x,a) = \lim_{t\to\infty} \mathbb{P}\left[x_t = x, a_t = a\right] = \nu^\pi(x)\pi(a|x).$$

For a compact notation, we will represent the decision variables μ as vectors in $\mathbb{R}^{\mathcal{X}\times\mathcal{A}}$ and introduce the linear operator $P^\mathsf{T} : \mathbb{R}^{\mathcal{X}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{X}}$ defined for each μ through $(P^\mathsf{T}\mu)(x') = \sum_{x,a} P(x'|x,a)\mu(x,a)$ for all x'. Similarly, we define the operator E^T acting on μ through the assignment $(E^\mathsf{T}\mu)(x) = \sum_a \mu(x,a)$ for all x, so $E^\mathsf{T}\mu^\pi = \nu^\pi$.

Proposition 3.2.1. *(Puterman, 1994, Theorem 8.8.6) A probability distribution $\mu \in \Delta_{\mathcal{X}\times\mathcal{A}}$ is a valid stationary distribution if and only if it satisfies the following system of equations known as flow equations:*

$$E^\mathsf{T}\mu = P^\mathsf{T}\mu.$$

The system of equations of the above lemma shows that for a stationary distribution, the probability of being in a given state is the same after applying the transition function. Also, we can see that the average reward of a policy π can be written in terms of the stationary distribution as

$$\rho^\pi = \lim\inf_{t\to\infty} \mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} r_t(x_t,a_t)\right] = \langle \mu^\pi, r\rangle. \tag{3.1}$$

This, together with Proposition 3.2.1 justifies the following proposition that suggest an LP that can be used to find the optimal policy:

Proposition 3.2.2. *Let μ^* be the optimal solution of the following LP:*

$$\begin{aligned} \textit{maximize}_{\mu\in\mathbb{R}_+^{\mathcal{X}\times\mathcal{A}}} \quad & \langle \mu, r\rangle \\ \textit{s.t.} \quad & E^\mathsf{T}\mu = P^\mathsf{T}\mu \\ & \langle \boldsymbol{1}, \mu\rangle = 1 \end{aligned} \tag{3.2}$$

Then, $\mu^* = \mu^{\pi^*}$ is the optimal state-action distribution induced by the optimal policy π^*.

If we let x be a state such that $\sum_{a'} \mu^\pi(x,a') \neq 0$, then for any policy π we can easily see that

$$\pi(a|x) = \frac{\mu^\pi(x,a)}{\sum_{a'} \mu^\pi(x,a')}.$$

Note that this can not be used for states where the visitation frequency is 0, since then the denominator of the above expression is 0. This result can be used to extract the optimal policy π^* from the optimal solution of the LP (3.2), μ^*. Nevertheless, the requirement of the visitation frequency being larger than 0 to be able to extract the policy is clearly a limitation of this method. Despite this, in later chapters we will present algorithms based on the LP (3.2) that do not suffer from this problem due to the usage of regularization, that will prevent the output policy from having zero visitation frequency in any state.

Let's now present another concept of great importance in the RL literature: the *value function*. Every policy π induces a value function $V^\pi : \mathcal{X} \to \mathbb{R}$ defined in the average reward setting as

$$V^\pi(x) = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} r(x_t, a_t) - \rho^\pi | x_0 = x\big],$$

which is the expected total difference between the stationary reward and the reward starting in state x. In the literature, the value function for the average reward setting that we just defined is often referred as the bias function. Value functions have a key role in MDP optimization and reinforcement learning, since they give a measure of how "good" a given state is in terms of the expected reward collected in the future, which can be used for evaluating and improving policies. Nevertheless, the exact definition of value function is different in different settings. The following proposition presents a very well known result about the system of equations known as *Bellman equations* (also known as Poisson equations in the average reward setting):

Proposition 3.2.3. *(Puterman, 1994, Corollary 8.2.7) Let ρ and V be a solution of the Bellman equations defined as*

$$V(x) = \sum_a \left[\pi(a|x)\left(r(x,a) + \sum_{x'} P(x'|x,a)V(x') - \rho\right)\right] \qquad \forall x \in \mathcal{X}. \tag{3.3}$$

(Bellman equations)

Then, $\rho = \rho^\pi$ and $V = V^\pi + k\mathbf{1}$, being k an arbitrary scalar.

Closely related to the value function, we can define the *Q-function* associated to a given policy π, $Q^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, as

$$Q^\pi(x,a) = \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} r(x_t, a_t) - \rho^* | x_0 = x, a_0 = a\big],$$

that can be understood as a value function where the first action is not drawn from the policy π but is set as $a_0 = a$. Similarly as with the value function, we can write the Bellman equations for Q-functions:

$$Q(x,a) = \sum_{x'} \left(r(x,a) + P(x'|x,a) \sum_{a'} \pi(a'|x') Q(x',a') - \rho \right) \tag{3.4}$$
$$\forall (x,a) \in \mathcal{X} \times \mathcal{A}.$$

Again, if ρ and Q are a solution of the above system of equations, then $\rho = \rho^\pi$ and $Q = Q^\pi + k\mathbf{1}$, with k an arbitrary scalar.

Equations (3.3) and (3.4) give the value function and Q-function respectively of a given policy π, but often we are interested in computing those values for the optimal policy π^* without knowing it. In this case, one can use the *Bellman optimality equations* presented in the following proposition:

Proposition 3.2.4. *(Puterman, 1994, Theorem 8.4.3.) Let ρ^* and V^* be a solution of the Bellman optimality equations defined as follows:*

$$V(x) = \max_a \left(r(x,a) + \sum_{x'} P(x'|x,a) V(x') - \rho \right) \qquad \forall x \in \mathcal{X}. \tag{3.5}$$

(Bellman optimality equations)

Then, $\rho^* = \rho^{\pi^*}$ and $V^* = V^{\pi^*} + k\mathbf{1}$, where k is an arbitrary scalar.

The optimal policy can then be easily extracted by picking the greedy action with respect to V^*, i.e. picking the actions that maximize the Bellman optimality equations. As before, we can easily derive the Bellman optimality equations for Q-function. Once the optimal Q-function Q^* has been computed, the optimal policy can be extracted by picking at each state x the greedy action that maximizes $Q^*(x,a)$.

We will now present another approach for finding the optimal policy π^* that like the LP (3.2) is based on linear programming but from a quite different perspective. Instead of computing the optimal state-action stationary distribution, the LP that we will present computes optimal value functions. Before going to this LP, let's first present the following result:

Proposition 3.2.5. *(Puterman, 1994, Theorem 8.4.1) Let (ρ, V) be such that*

$$V(x) \geq \max_a \left(r(x,a) + \sum_{x'} P(x'|x,a) V(x') - \rho \right) \qquad \forall x \in \mathcal{X}. \tag{3.6}$$

Then, $\rho \geq \rho^*$.

This result implies that ρ^* and V^* can be found by finding the minimum ρ such that for some V, the system of inequalities 3.6 is satisfied. Furthermore, we can see that this latter condition is equivalent to the following set of constrains

$$V - r(\cdot, a) - (PV)(\cdot, a) + \rho \mathbf{1} \geq 0 \qquad \forall a.$$

This justifies the following proposition that presents the LP for value functions:

Proposition 3.2.6. *Let (ρ^*, V^*) be a solution of the following LP:*

$$\begin{aligned} &\textit{minimize}_{\rho \in \mathbb{R}, V \in \mathbb{R}^{\mathcal{X}}} \quad \rho \\ &\qquad\qquad \textit{s.t.} \quad EV \geq r + PV - \rho. \end{aligned} \tag{3.7}$$

Then, $\rho^* = \rho^{\pi^*}$ and $V^* = V^{\pi^*} + k\mathbf{1}$, where k is an arbitrary scalar.

So far we have seen two LPs that can be used to find the optimal policy of a given MDP. The following proposition gives a very insightful connection between these two LPs that will be exploited over and over during the rest of this work:

Proposition 3.2.7. *The LP* (3.2) *is the dual of the LP* (3.7).

The proof of the above proposition is a direct application of Lagrangian duality for LPs. See for example Puterman, 1994, APPENDIX D.

Finally, we present the following proposition that gives an upper bound on the value function of any policy π, since it is an important result for analyzing the algorithms in later sections:

Proposition 3.2.8. *Assume that $r(x, a) \leq 1$ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$. Then, for any policy π, we have $\|V^\pi\|_\infty \leq \tau_{mix}$, where $\tau_{mix} = 2C(1 + \tau)$. Similarly, it also holds that $\|Q^\pi\|_\infty \leq \tau_{mix}$.*

Proof. We denote as μ_x the state-action distribution such that $\mu_x(x', a) = \pi(x'|a)$ if $x' = x$ and 0 otherwise. Similarly, we denote as ν_x the state-action distribution such that $\nu_x(x') = 1$ if $x = x'$ and 0 otherwise. Then, using the definition of value function we can write

$$\begin{aligned} V^\pi(x) &= \sum_{t=0}^{\infty} \left\langle P^t \mu_x - \mu^\pi, r \right\rangle \leq \sum_{t=0}^{\infty} \left\| P^t(\mu_x - \mu^\pi) \right\|_1 = \sum_{t=0}^{\infty} \left\| P^t_\pi(\nu_x - \nu^\pi) \right\|_1 \\ &\leq \sum_{t=0}^{\infty} C e^{-t/\tau} \left\| \nu_x - \nu^\pi \right\|_1 \leq \sum_{t=0}^{\infty} 2C e^{-t/\tau} \leq 2C(1 + \tau) \end{aligned}$$

Where we have used Assumption 1 in the second inequality and $\sum_{t=0}^{\infty} e^{-\frac{t}{\tau}} \leq \frac{1}{1 - e^{-\frac{1}{\tau}}} \leq (1 + \tau)$ in the last one. The same arguments can be used for bounding Q^π. □

In what follows, we refer to the quantity τ_{mix} as the *mixing time* of the MDP. Note that this is just one of many possible definitions of a mixing time, see, e.g., Seneta [2006]; Levin and Peres [2017].

3.3 Normalized discounted reward setting

In the normalized discounted reward setting, the objective is to maximize the discounted reward defined as

$$(1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,a_t)\right],$$

where $\gamma \in (0,1)$ is the discount factor and the state x_0 is drawn from a fixed initial-state distribution ν_0. Since the ideas presented in this chapter are very similar to the ones in the previous one, we will go through them in a more informal and superficial way, to avoid a repetitive lecture.

As in the average reward setting, in the discounted setting it is also useful to work with some notion of occupancy measure. In this case though, instead of the state-action probability distribution we will work with the *normalized discounted state-action occupancy measure* (in short, occupancy measure), that is defined for a given policy π as

$$\mu^\pi(x,a) = (1-\gamma)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{I}_{\{(x_t,a_t)=(x,a)\}}\right]$$

with $x_0 \propto p_0$. Note that we use the same notation as with stationary state-action probability distributions, assuming that their nature should be clear by the context. The following proposition presents the flow equations for the discounted setting:

Proposition 3.3.1. *(Puterman, 1994, Proposition 6.9.1) The vector μ is a valid occupancy measure if and only if it satisfies the following system of equations:*

$$E^\intercal\mu = \gamma P^\intercal\mu + (1-\gamma)p_0$$

Furthermore, the discounted reward associated to a policy π can be written as

$$\rho^\pi = (1-\gamma)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,a_t)\right] = \langle\mu^\pi, r\rangle,$$

As in the previous section, this justifies an LP for finding the optimal occupancy measure. Before presenting it, let's first show the definition of value function V^π and Q-function Q^π associated to a given policy π in the discounted setting:

$$V^\pi(x) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,a_t)|x_0 = x\right], \tag{3.8}$$

$$Q^\pi(x,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) | x_0 = x, a_0 = a \right] \tag{3.9}$$

We can now present the following proposition containing two LPs that can be used to find optimal policies in the discounted setting:

Proposition 3.3.2. *(Puterman, 1994, Section 6.9) The vector μ^* is the solution of the LP*

$$\begin{aligned} \textit{maximize}_{\mu \in \mathbb{R}_+^{\mathcal{X}\times\mathcal{A}}} \quad & \langle \mu, r \rangle \\ \textit{s.t.} \quad & E^\mathsf{T}\mu = \gamma P^\mathsf{T}\mu + (1-\gamma)p_0. \end{aligned} \tag{3.10}$$

if and only if $\mu^* = \mu^{\pi^*}$ is the occupancy measure of the optimal policy π^*. Similarly, V^* is the solution of the LP

$$\begin{aligned} \textit{minimize}_{V \in \mathbb{R}^{\mathcal{X}}} \quad & (1-\gamma)\langle p_0, V \rangle \\ \textit{s.t.} \quad & EV \geq r + \gamma PV, \end{aligned} \tag{3.11}$$

if and only if $V^* = V^{\pi^*}$ is the value function of the optimal policy π^*. Furthermore, the LP (3.10) *is the dual of the LP* (3.11).

We also show the following proposition that presents the Bellman equations and Bellman optimality equations for the discounted setting:

Proposition 3.3.3. *(Puterman, 1994, Propositions 6.1.1 and 6.2.2) Let V be the solution of the Bellman equations for the discounted setting defined as*

$$V(x) = \sum_a \pi(a|x) \left(r(x,a) + \sum_{x'} \gamma P(x'|x,a) V(x') \right) \qquad \forall x \in \mathcal{X}. \tag{3.12}$$

(Bellman equations)

Then, $V = V^\pi$ is the value function of the policy π. Let now V^* be the solution of the Bellman optimality equations for the discounted setting defined as

$$V(x) = \max_a \left(r(x,a) + \sum_{x'} \gamma P(x'|x,a) V(x') \right) \qquad \forall x \in \mathcal{X}. \tag{3.13}$$

(Bellman optimality equations)

Then, $V^* = V^{\pi^*}$ is the value function of the optimal policy π^*.

The following proposition presents the upper bound for the value function and Q-function in the discounted setting:

Proposition 3.3.4. *Assume that $r(x,a) \leq 1$ for all $(x,a) \in \mathcal{X} \times \mathcal{A}$. Then, for any policy π, we have $\|V^\pi\|_\infty \leq \frac{1}{1-\gamma}$. Similarly, it also holds that $\|Q^\pi\|_\infty \leq \frac{1}{1-\gamma}$.*

Proof. The proof is a direct consequence of plugging the assumption $r(x,a) \leq 1$ in the definition of the value function (3.8) and using the inequality $\sum_{t=0}^{\infty} \gamma^t \leq \frac{1}{1-\gamma}$ for all $\gamma \in (0,1)$:

$$V^\pi(x) \leq \sum_{t=0}^{\infty} \gamma^t \leq \frac{1}{1-\gamma}$$

The same argument can be used for bounding Q^π. □

3.4 Approximate dynamic programming

The tools shown in the previous section give the foundations of methods and algorithms that allow us to find optimal policies in real world problems. In these problems, we generally do not know the transition probabilities (i.e., we do not know the model), so we need methods that work with sample trajectories. Furthermore, in real-world problems, the state-action space can be huge, so working with value functions, Q-functions, policies or distributions for every state or state-action pair is infeasible. For this reason, we need to parameterize the quantity of interest and work with methods that can learn the parameters of the model with samples.

In this section we assume that we are in the discounted reward setting. We chose this setting because it is where we can find most of the related literature. Nevertheless, similar algorithms can be found for the average reward setting. Below we present two different approaches for learning optimal policies: policy iteration and Q-learning.

3.4.1 Policy iteration

One of the most well known schemes to find the optimal policy is policy iteration. This scheme works by iteratively evaluating and improving policies with the following two steps:

- *Policy evaluation*: an estimate of the value function $\widehat{V}^\pi$ or the Q-function $\widehat{Q}^\pi$ of the current policy π is computed.
- *Policy improvement*: A new policy is computed based on the value function or Q-function found in the policy evaluation step. For Q-functions we can use the greedy policy that at any state x takes the action

$$\arg\max_a \widehat{Q}^\pi(x,a).$$

> Similarly, for value functions we can use the greedy policy that at any state x takes the action
>
> $$\arg\max_a \mathbb{E}_{x' \sim P(x'|x,a)} \left[r(x,a) + \gamma \widehat{V}^\pi(x') \right].$$
>
> Realize that in this latter case with value functions, it is needed some knowledge about the dynamics of the MDP in order to compute the expectation over next states. Other kind of non-greedy updates are also possible, as we will see in later chapters.

In this section, we will focus on Q-functions, but most results extend trivially to value functions. We will denote as Q_θ the parameterized Q-functions.

Ideally, to obtain a good approximation of Q^π in the policy evaluation step, we would like to minimize the following loss function

$$\begin{aligned}\mathbb{L}(Q_\theta) &= \\ &\mathbb{E}_{(x,a)\sim\mu}\left[\left(r(x,a) + \gamma \sum_{x'} P(x'|x,a) \sum_{a'} \pi(a'|x') Q_\theta(x',a') - Q_\theta(x,a)\right)^2\right],\end{aligned} \tag{3.14}$$

where the quantity inside the parenthesis is called the *Bellman error*. For simplicity, we will assume that $\mu = \mu^\pi$ is the stationary state-action distribution of the policy π. Since we do not have access to the model, we need a sample-based estimate of the above loss. The estimate can be written as follows:

$$\widehat{\mathbb{L}}(Q_\theta) = \mathbb{E}_{(x,a)\sim\widehat{\mu}, x' \sim P(\cdot|x,a)} \left[\left(r(x,a) + \gamma \sum_{a'} \pi(a'|x') Q_\theta(x',a') - Q_\theta(x,a)\right)^2\right] \tag{3.15}$$

where the Bellman error has been changed by the *empirical Bellman error* (also known as temporal difference (TD) error) and $\widehat{\mu}$ is an empirical distribution coming from μ. The problem of this estimator is that it is a biased estimate, which can be easily seen if we compute its expectation:

$$\mathbb{E}\left[\widehat{\mathbb{L}}(Q_\theta)\right] = \mathbb{L}(Q_\theta) + \mathbb{E}_{(x,a)\sim\mu}\left[\mathrm{Var}\left(\gamma \sum_{a'} \pi(a'|x') Q_\theta(x',a')\right)\right].$$

This problem is known as the double sampling problem. Its name comes from the fact that this bias can be eliminated as follows: For each state-action pair (x,a)

in expression (3.15), draw two independent next-state samples x' and x'' from $P(\cdot|x,a)$. Then, replacing the term

$$\left(\gamma \sum_a \pi(a|x')Q_\theta(x',a')\right)^2$$

that appears when expanding expression (3.15) for

$$\left(\gamma \sum_{a'} \pi(a'|x')Q_\theta(x',a')\right)\left(\gamma \sum_{a'} \pi(a'|x'')Q_\theta(x'',a')\right)$$

makes the estimate unbiased and solves the problem. Of course this is problematic in practice since drawing an extra next state requires a simulator that can sample next states from any state-action pair.

We will now present two *projected policy evaluation* methods that are another option to perform the policy evaluation step without having this double sampling problem.

Let's start by rewriting the system of equations (3.12) as

$$Q = B^\pi Q \tag{3.16}$$

where the operator B^π is defined as

$$(B^\pi Q)(x,a) = r(x,a) + \gamma \sum_{x'} P(x'|x,a) \sum_{a'} \pi(a'|x)Q(x',a').$$

We also define the weighted quadratic norm $\|Q\|_\mu^2 = \sum_{x,a} \mu(x,a)(Q(x,a))^2$, and the projection operator

$$\Pi(Q) \in \arg\min_{Q_\theta \in \mathbb{R}^m} \|Q_\theta - V\|_\mu^2 .$$

Projected policy evaluation methods aim to find the value function that is approximately equal to the projection of $B^\pi(Q_\theta)$ on the space of linearly parameterized value functions:

$$Q_\theta = \Pi B^\pi(Q_\theta). \tag{3.17}$$

We can see that this problem is equivalent to solving the following minimization problem:

$$\min_\theta \|Q_\theta - \Pi B^\pi(Q_\theta)\|_\mu^2 . \tag{3.18}$$

Until the end of this section, we consider linearly parameterized Q-functions. We introduce a state-action feature map $\varphi : \mathbb{R}^{\mathcal{X}\times\mathcal{A}} \to \mathbb{R}^m$ and consider a parameterization of the Q-function of the form $Q_\theta(x,a) = \langle \varphi(x,a), \theta \rangle$ where $\theta \in \mathbb{R}^m$ is the new optimization variable, and the corresponding linear operator Φ with rows $\varphi(x, a)$ such that $Q_\theta = \Phi\theta$.

We now present the sample-based version of two projected policy evaluation methods. Let's first consider N sample transitions $\{x_i, a_i, x'_i, a'_i, r_i\}_{n=1}^N$ where x'_i is drawn from the distribution $P(\cdot|x, a)$, a' from $\pi(\cdot|x')$, and the state-action pairs are drawn from μ. We now define the matrices A and B, and the vector b as

$$\begin{aligned} A_i &= A_{i-1} + \varphi(x_i, a_i)\varphi^\intercal(x_i, a_i) \\ B_i &= B_{i-1} + \varphi(x_i, a_i)\varphi^\intercal(x'_i, a'_i) \\ b_i &= b_{i-1} + \varphi(x_i, a_i)r_i \end{aligned} \tag{3.19}$$

with $A_0 = 0$, $B_0 = 0$ and $b_0 = 0$.

The first method, called *least squares temporal difference* (LSTD), directly aims to solve the optimization problem (3.18) in one shot. To do so, in its basic sample-based version LSTD takes the N samples and solves the equation

$$\left(\frac{1}{N}A_N - \gamma\frac{1}{N}B_N\right)\theta = \frac{1}{N}b_N$$

to find the parameter θ in a single shot.

The second method, called *least squares policy evaluation* (LSPE) aims to solve the optimization problem (3.18) iteratively. To do so, in its basic sample-based version LSPE starts with an arbitrary initial parameter vector θ_0 and updates it after every sample using:

$$\theta_i = \theta_{i-1} + \alpha(\theta'_i - \theta_{i-1})$$

where

$$\frac{1}{i}A_i\theta'_i = \gamma\frac{1}{i}B_i\theta_{i-1} + \frac{1}{i}b_i$$

and α is a learning parameter. Under the condition that $\mu = \mu^\pi$, the solution of both LSTD and LSPE converge to the solution of problem (3.14) as N goes to infinity, without suffering from the double sampling problem.

3.4.2 Q-learning

Another option for learning the optimal policy is the family of methods under the name of *Q-learning*. Instead of approximating the value functions of a given policy to improve it, Q-learning methods directly aim to approximate the Q-function of the optimal policy, Q^*.

The main idea behind Q-learning is very similar to what we have just seen for policy iteration. Here we would like to minimize the following loss to find a good approximator of Q^*:

$$\mathbb{L}(Q_\theta) = \mathbb{E}_{(x,a)\sim\mu}\left[\Big(r(x,a) + \gamma \max_{a'} Q_\theta(x',a') - Q_\theta(x,a)\Big)^2\right], \tag{3.20}$$

that is, the expected squared Bellman error of the Bellman optimality equations (or just Bellman error if it is clear by the context). As before, we need a sample-based approach. Thus, Q-learning algorithms are based on sequentially computing the approximations of Q^* by minimizing an empirical loss function based on the empirical squared Bellman error:

$$\widehat{\mathbb{L}}(Q_\theta) = \mathbb{E}_{(x,a)\sim\widehat{\mu},x'\sim P(\cdot|x,a)}\left[\Big(r(x,a) + \gamma \max_{a'} Q_\theta(x',a') - Q_\theta(x,a)\Big)^2\right] \tag{3.21}$$

Where like before, $\widehat{\mu}$ is an empirical distribution. As we saw for the loss function (3.15), the minimization of the above loss suffers from double sampling problem. The popular work of Mnih et al. [2015] presents a useful technique for alleviating this problem. There, a deep neural network is used to approximate the optimal Q-values, and the parameters of the network are trained through stochastic gradient descent trying to minimize the following loss:

$$\widehat{\mathbb{L}}(Q_\theta) = \mathbb{E}_{(x,a)\sim\widehat{\mu},x'\sim P(\cdot|x,a)}\left[\Big(r(x,a) + \gamma \max_{a'} Q_{\bar{\theta}}(x',a') - Q_\theta(x,a)\Big)^2\right], \tag{3.22}$$

where the $\bar{\theta}$ that parameterizes the Q-function inside the $\max$ (called the *target network*) is held fixed and updated only every some iterations. This technique is what mitigates the double sampling problem, which is key in the great performance of the algorithm.

Chapter 4

SADDLE-POINT OPTIMIZATION FOR SOLVING LARGE-SCALE MARKOV DECISION PROCESSES

4.1 Introduction

In this chapter we present an approach based on a *bilinear saddle-point* formulation of the linear program 3.2, building on the well-known general Lagrangian formulation of a constrained optimization problem (see Section 2.3). One particular advantage of this formulation is that it enables a straightforward form of dimensionality reduction of the original problem through a linear parameterization of the optimization variables, which provides a natural framework for studying effects of “function approximation” in the underlying policy optimization problem. Our main contribution regarding this setting lies in characterizing a set of assumptions that allow a reduced-order saddle-point representation of the optimal policy. These include a realizability assumption and a newly identified *coherence assumption* about the subspaces used for approximation. Our main positive result is showing that these conditions are sufficient for constructing an algorithm that outputs an ε-optimal policy with runtime guarantees of $\widetilde{\mathcal{O}}\left(\tau_{mix}^2 m^2/\varepsilon\right)$, where m is the number of variables in the relaxed optimization problem, and τ_{mix} is the mixing time defined in Proposition 3.2.8. Our approach is based on the celebrated mirror prox algorithm of Nemirovski [2004]. We complement our positive results by showing that our newly defined coherence assumption is necessary for the relaxed saddle-point approach to be viable: we construct a simple example violating the assumption, where achieving full optimality on the relaxed problem leads to a suboptimal policy.

We are not the first ones to consider saddle-point methods for optimization in Markov decision processes. Wang [2017] proposed variants of mirror descent to solve the original saddle-point problem without relaxations and provide runtime guarantees of $\widetilde{\mathcal{O}}\big((\alpha\tau_{mix})^2\,|\mathcal{X}||\mathcal{A}|/\varepsilon^2\big)$, where $\mathcal{X}$ and $\mathcal{A}$ are the finite state and action spaces, and α is a parameter that characterizes the uniformity of the stationary distributions of every policy. Specifically, their assumption implies[1] that for the stationary distribution ν^π of any policy π, one has $\frac{\max_x \nu^\pi(x)}{\min_{x'} \nu^\pi(x')} \leq \alpha$. In most cases of practical interest, this ratio is at least as large as $|\mathcal{X}|$ (e.g., when there are states that some policies visit with constant probability), and can easily be exponentially large in $|\mathcal{X}|$, or even infinite if the underlying MDP has transient states. When specialized to this setting, our bounds replace α^2 by the much more manageable $|\mathcal{X}|$ and also improve the dependence on ε from $1/\varepsilon^2$ to $1/\varepsilon$. One downside of our method is that we need full access to the transition probabilities of the MDP, whereas the algorithm of Wang [2017] only requires a generative model.

The linearly relaxed saddle-point problem we consider was first studied by Lakshminarayanan et al. [2017] and Chen et al. [2018]. Our runtime guarantees improve over the ones claimed by Chen et al. [2018] in a similar way as our first set of results improve over those of Wang [2017]. Notably, their results still feature a factor of α^2, which generally depends on the size of the original state space rather than the number of features, rendering these guarantees void of meaning in very large state spaces. In contrast, our bounds replace this factor by the number of features N. Notably, the results of Chen et al. [2018] does not require the coherence assumption to hold, which raises some interesting questions regarding the generality of both our results and theirs. One particular conclusion that one can draw from the tension between these results is that in order to derive performance bounds from the relaxed linear program formulation, one either needs to assume that the coherence condition holds, or that the value of α is bounded by some constant. Indeed, in our counterexample showing the necessity of the coherence condition, the value of α is infinite, and thus the upper bounds of Chen et al. [2018] do not apply.

4.2 The linearly relaxed saddle-point problem

We start this section by recalling the dual optimization problem 3.2, that we can rewrite as

$$\begin{aligned} \text{maximize}_{\mu\in\mathcal{P}} \quad & \langle \mu, r\rangle \\ \text{s.t.} \quad & E^\intercal \mu = P^\intercal \mu \end{aligned} \tag{4.1}$$

[1] The actual assumption made by Wang [2017] is even more restrictive.

where $\mathcal{P} = \mathcal{P}_{\mathcal{X}\times\mathcal{A}}$ is the set of probability distributions over the state-action space, so we have moved the constraint regarding μ being in the simplex to a restriction in the optimization domain. Then, as shown in Section 2.3, we compute the Lagrangian of the problem above with $V \in \mathbb{R}^{\mathcal{X}}$ as the Lagrange multiplier

$$\mathcal{L}(V, \mu) = \langle \mu, r \rangle + \langle \mu, (P - E)V \rangle \,. \tag{4.2}$$

We can now propose the following saddle-point problem, that is equivalent to the LP (4.1):

$$\min_{V\in\mathbb{R}^{\mathcal{X}}} \max_{\mu\in\mathcal{P}} \mathcal{L}(V, \mu) = \langle \mu, r \rangle + \langle \mu, (P - E)V \rangle \,, \tag{4.3}$$

While one can directly derive optimization algorithms to solve the saddle-point problem (4.3), such a direct approach would suffer from serious scalability issues due to the sheer number of variables involved in the problem: the size of the objects of interest μ and V are linear in the size of the state space, which results in prohibitive memory and computation costs for most algorithms. To address this issue, we study a *linearly relaxed* version of the full saddle-point problem that reduces the order of the original optimization problem by linearly parametrizing the variables V and μ through two sets of *feature maps*. Formally, we consider the matrices Ψ of size $\mathcal{X} \times m$ and W of size $n \times \mathcal{X} \times \mathcal{A}$, and introduce the new optimization variables $y \in \mathbb{R}^n$ and $\theta \in \mathbb{R}^m$, and use these to (hopefully) approximate the solutions to (4.3) as $\mu^* \approx W^\mathsf{T} y$ and $V^* \approx \Psi\theta$. For a tractable problem formulation, we will assume that the rows of W are non-negative and sum to one: $W_{i,x} \geq 0$ for all x, and $\sum_x W_{i,x} = 1$ for all n. We will also assume that all entries of Ψ are bounded by 1 in absolute value. These conditions enable us to optimize y over the probability simplex $\widetilde{\mathcal{P}} = \mathcal{P}_{[n]}$ and to formulate our relaxed saddle-point problem as

$$\min_{\theta\in\mathbb{R}^m} \max_{y\in\widetilde{\mathcal{P}}} \widetilde{\mathcal{L}}(\theta, y) = \min_{\theta\in\mathbb{R}^m} \max_{y\in\widetilde{\mathcal{P}}} \langle W^\mathsf{T} y, (P - E)\Psi\theta \rangle + \langle W^\mathsf{T} y, r \rangle \,. \tag{4.4}$$

The relaxed optimization problem above has been studied before by Lakshminarayanan and Bhatnagar [2015]; Lakshminarayanan et al. [2017], and Chen et al. [2018]. Lakshminarayanan and Bhatnagar [2015]; Lakshminarayanan et al. [2017] studied the relaxed linear program underlying (4.4) as a natural extension of the classic relaxed LP analyzed by de Farias and Van Roy [2003], and have focused on understanding the discrepancies between the optimal value function and the relaxed value function attaining the minimum in the above expression. On the other hand, Chen et al. [2018] focused on proposing stochastic optimization algorithms and analyzing the rate of convergence to the optimum, but provide little insight about the quality of the optimal solution of the relaxed problem.

4.2.1 Effect of the relaxation

The goal of this section is to obtain a better understanding of the effects of approximation on the policies that can be obtained through approximately solving the relaxed saddle-point problem (4.4). One peculiar challenge associated with our setting is that it is not enough to ensure that the values of $\widetilde{\mathcal{L}}$ and $\mathcal{L}$ are close at their respective saddle points, but we rather need to understand the performance of the policy extracted from the optimal solution y^*. Precisely, defining the policy extracted from y as

$$\pi^y(a|x) = \frac{(W^\mathsf{T} y)(x,a)}{\sum_{a'}(W^\mathsf{T} y)(x,a')}$$

for all x, a, and the corresponding stationary distribution induced in the original MDP as μ^y, we are interested in the suboptimality gap

$$\left\langle \mu^* - \mu^{y^*}, r \right\rangle .$$

We focus on identifying assumptions on the feature maps that allow the computation of true optimal policies with (almost) zero suboptimality gap. Specifically, we will show that the following two assumptions have a decisive role in making this gap small:

Assumption 2 (Realizability)**.** *The optimal solution is realizable by the feature maps: there exists (θ^*, y^*) such that $V^* = \Psi\theta^*$ and $\mu^* = W^\mathsf{T} y^*$. Additionally, $\|\theta^*\|_\infty \leq U\tau_{mix}$ holds for some $U > 0$.*

Assumption 3 (Coherence)**.** *The image of the set $\widetilde{\mathcal{P}}$ under the map $(P-E)^\mathsf{T} W^\mathsf{T}$ is included in the column space of Ψ: for all $y \in \widetilde{\mathcal{P}}$ such that $(P-E)^\mathsf{T} W^\mathsf{T} y \neq 0$, there exists a $\theta \in \mathbb{R}^m$ such that $\langle (P-E)^\mathsf{T} W^\mathsf{T} y, \Psi\theta\rangle \neq 0$. Additionally, for all $V \in \mathbb{R}^{\mathcal{X}}$ with $\|V\|_\infty \leq 1$, there exists a $\theta \in \mathbb{R}^m$ with $\|\theta\|_\infty \leq U$ such that $\langle (P-E)^\mathsf{T} W^\mathsf{T} y, \Psi\theta\rangle = \langle (P-E)^\mathsf{T} W^\mathsf{T} y, V\rangle$.*

The second condition of each assumption is to ensure that the columns of Ψ are well-conditioned and are satisfied if the columns form an orthonormal basis. Assumption 2 is trivially necessary, and despite it may already seem sufficient for the relaxed problem to be a good enough approximation of the original one, we argue that Assumption 3 is also necessary for the relaxation scheme to be reliable. Specifically, the following theorem shows that in the absence of the coherence assumption, near-optimal solutions to the relaxed saddle-point problem (4.4) can still lead to suboptimal policies in the original MDP.

Theorem 4.2.1. *For any $\varepsilon > 0$, there exists an MDP with relaxations W, Ψ satisfying Assumption 2 and violating Assumption 3, and a solution $(\widehat{\theta}, \widehat{y}_\varepsilon)$ simultaneously satisfying*

$$\mathcal{L}(\Psi\widehat{\theta}, \mu^*) - \mathcal{L}(V^*, W^\mathsf{T}\widehat{y}_\varepsilon) = \varepsilon$$

and

$$\left\langle \mu^* - \mu^{\widehat{y}_\varepsilon}, r \right\rangle = 2/3.$$

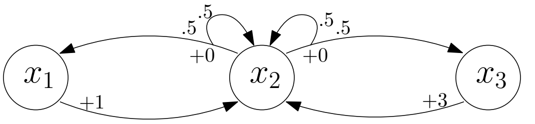


Figure 4.1: Three-state MDP for illustrating the necessity of the coherence assumption. The two actions from x_2 have stochastic transitions with probability $1/2$ of staying in x_2 and $1/2$ of moving to x_1 or x_3 depending on the action. All other transitions are deterministic. Rewards are given as a function of the state as $r(x_1) = 1$, $r(x_2) = 0$ and $r(x_3) = 3$.

Proof. The proof is based on constructing an MDP with three states x_1 (left), x_2 (middle) and x_3 (right) and two actions a_l and a_r corresponding to moving "left" or "right", respectively. The transition probabilities and rewards are as shown on Figure 4.1. It is easy to see that the optimal policy is to take action a_r in state x_2, which yields the optimal stationary state-action distribution

$$\mu^* = (\mu(x_1, a_r), \mu(x_2, a_l), \mu(x_2, a_r), \mu(x_3, a_l))^\intercal = \left(0, 0, \frac{1}{3}, \frac{2}{3}\right)^\intercal$$

and the optimal average reward $\langle \mu^*, r \rangle = 1$. The optimal value function can be shown to be $V^* = (-1, -1, 1)^\intercal$. For the relaxation, define $\Psi = V^*$ and W as the identity map so that the realizability assumption is clearly fulfilled with $y^* = \mu^*$ and $\theta^* = 1$. Now, choosing $\widehat{y} = (1, 0, 0, 0)^\intercal$ results in

$$\begin{aligned}\langle W^\intercal \widehat{y}, (P - E)\Psi\theta \rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 \\ 1/2 & -1/2 & 0 \\ 0 & -1/2 & 1/2 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \theta \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 2 \end{pmatrix} \theta = 0 \cdot \theta\end{aligned}$$

for any θ. Observing that taking $V = (-1, 1, 0)^\intercal$ gives $\langle W^\intercal \widehat{y}, (P - E)V \rangle = 2$, we see that the coherence assumption is violated since there exists no θ such that the condition $\langle W^\intercal \widehat{y}, (P - E)V \rangle = \langle W^\intercal \widehat{y}, (P - E)\Psi\theta \rangle$ is satisfied. Furthermore, it is easy to see that for any θ, $(\widehat{y}, \theta)$ is an optimal solution to the relaxed saddle-point problem (4.4) with $\langle W^\intercal \widehat{y}, r\rangle = 1$ since

$$\widetilde{\mathcal{L}}(\theta, \widehat{y}) = \widehat{y}^\intercal W(P-E)\Psi\theta + \widehat{y}^\intercal W r = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\0\\0\\3\end{pmatrix} = 1.$$

The resulting optimal state-action distribution $\widehat{\mu} = W^\intercal \widehat{y} = \widehat{y}$ is clearly not a stationary distribution.

To conclude the proof, fix any ε and consider $\widehat{y}_\varepsilon = (1-\varepsilon, \varepsilon, 0, 0)^\intercal$ and any $\widehat{\theta}$. Noticing that $\langle W^\intercal \widehat{y}_\varepsilon, (P-E)\Psi\theta\rangle = 0$ holds for all θ, the duality gap associated with $(\widehat{\theta}, \widehat{y}_\varepsilon)$ can be seen to be

$$\begin{aligned}\mathcal{L}(\Psi\widehat{\theta}, \mu^*) - \mathcal{L}(V^*, W^\intercal \widehat{y}_\varepsilon) &= \begin{pmatrix}0 & 0 & 2/3 & 1/3\end{pmatrix}\begin{pmatrix}1\\0\\0\\3\end{pmatrix} - \begin{pmatrix}1-\varepsilon & \varepsilon & 0 & 0\end{pmatrix}\begin{pmatrix}1\\0\\0\\3\end{pmatrix}\\ &= 1 - (1-\varepsilon) = \varepsilon.\end{aligned}$$

The policy $\pi^{\widehat{y}_\varepsilon}$ extracted from the state-action distribution $\widehat{y}_\varepsilon$ takes action a_l in state x_2, which results in an average reward of $2/3$. These two statements together prove the theorem. □

4.3 Mirror prox for policy optimization

In this section, we provide our main positive results: deriving strong performance guarantees for policies derived from approximate solutions of (4.4) under Assumptions 2 and 3. Our algorithm attaining these guarantees is based on the mirror prox algorithmic scheme presented in Section 2.4.2 and adapted to saddle-point optimization in Section 2.4.3.

We instantiate the mirror prox method to address the relaxed saddle-point problem as follows. Our optimization variables will be $z = (\theta, y)$ and the monotone operator g will be chosen as

$$g(z) = \begin{pmatrix} \nabla_\theta \widetilde{\mathcal{L}} \\ -\nabla_y \widetilde{\mathcal{L}} \end{pmatrix} = \begin{pmatrix} \Psi^\intercal (P-E)^\intercal W^\intercal y \\ -Wr - W(P-E)\Psi\theta \end{pmatrix}. \tag{4.5}$$

As a mirror map, we will use the function

$$\Phi(z) = \frac{1}{2}\left\|\theta\right\|_2^2 + \sum_{j=1}^{M} y(j) \log y(j),$$

Algorithm 1: MPPO

```
Compute $A = W(P - E)\Psi$
for $k = 0, 1, 2, \dots, K - 1$ do
    Extrapolation step:
        $\widehat{\theta}_{k+1} = \theta_k - \eta A^\intercal y_k \qquad \widehat{y}_{k+1}(i) \propto y_k(i) e^{\eta((Wr)(i)+(A\theta_k)(i))}$
    Gradient step:
        $\theta_{k+1} = \theta_k - \eta A^\intercal \widehat{y}_{k+1} \qquad y_{k+1}(i) \propto y_k(i) e^{\eta\big((Wr)(i)+\big(A\widehat{\theta}_{k+1}\big)(i)\big)}$
end
Compute $\overline{\theta}_K = \frac{1}{K}\sum_{k=1}^{K} \widehat{\theta}_k$ and $\overline{y}_K = \frac{1}{K}\sum_{k=1}^{K} \widehat{y}_k$ :
Result: $\pi_K = \pi^{\overline{y}_K}$
```

that is, a linear combination of the squared 2-norm of the value-function parameters θ and the Shannon entropy of the distribution y. Since $\frac{1}{2}\|\theta\|_2^2$ and $\sum_{j=1}^{M} y(j)\log y(j)$ are 1-strongly convex w.r.t. the l_2 and l_1 norms respectively, Φ is 1-strongly convex on $\mathcal{Z}$ w.r.t. the norm $\|z\|^2 = \|\theta\|_2^2 + \|y\|_1^2$. Notice that the Bregman divergence associated to Φ between $z = (\theta, y)$ and $z = (\theta', y')$ is

$$D(z\|z') = \frac{1}{2}\|\theta - \theta'\|_2^2 + \sum_i y(i)\log\frac{y(i)}{y'(i)}.$$

Given the above specifications, the updates of our algorithm can be written as

$$\begin{aligned} &\widehat{\theta}_{k+1} = \theta_k - \eta\Psi^\intercal(P - E)^\intercal W^\intercal y_k, \\ &\widehat{y}_{k+1}(i) \propto y_k(i) e^{\eta((Wr)(i)+(W(P-E)\Psi\theta_k)(i))} \end{aligned} \tag{4.6}$$

$$\begin{aligned} &\theta_{k+1} = \theta_k - \eta\Psi^\intercal(P - E)^\intercal W^\intercal \widehat{y}_{k+1}, \\ &y_{k+1}(i) \propto y_k(i) e^{\eta\big((Wr)(i)+\big(W(P-E)\Psi\widehat{\theta}_{k+1}\big)(i)\big)}, \end{aligned} \tag{4.7}$$

where we used the notation "$\propto$" to signify that $\widehat{y}_{k+1}$ and y_{k+1} are normalized multiplicatively after each update so that $\sum_i y_{k+1}(i) = 1$ is satisfied. Also introducing the notations $\overline{y}_K = \frac{1}{K}\sum_{k=1}^{K} y_k$ and $\overline{\theta}_K = \frac{1}{K}\sum_{k=1}^{K} \widehat{\theta}_k$, the algorithm outputs the policy extracted from the distribution $\overline{y}_K$: $\pi_K = \pi^{\overline{y}_K}$. Algorithm 1 contains the pseudocode for the algorithm that we have just described, that we call MPPO (mirror prox policy optimization).

By recalling the average reward of the optimal state-action stationary distribution ρ^* and defining ρ_K as the average reward of the policy output by our algorithm $\rho_K = \langle \mu^{y_K}, r \rangle$, we can present the following theorem states one of our main results regarding the suboptimality of π_K:

Theorem 4.3.1. *Suppose that Assumptions 1, 2 and 3 hold and $\eta \leq \frac{1}{2m}$. Then, the average reward ρ_K output by the algorithm satisfies*

$$\rho^* - \rho_K \leq \frac{\frac{1}{2}\tau_{mix}^2 U^2 m + \log n}{\eta K}.$$

In particular, setting $\eta = \frac{1}{2m}$, the bound becomes $\rho^* - \rho_K = \mathcal{O}\left(\frac{\tau_{mix}^2 U^2 m^2}{K}\right)$.

We leave the proof of the theorem for the next chapter, where the algorithm is analyzed.

This result can be tightened by a factor of m if we further assume that the rows of Ψ are chosen as probability distributions. This can be seen in the proof of the theorem realizing that in this case $D(z^* \| z_0) \leq \frac{1}{2} U^2 \tau_{mix}^2 + \log(n)$.

In the special case where Ψ and W are the identity maps, the relaxed saddle-point problem becomes the original problem (4.3), our Assumptions 2 and 3 are clearly satisfied with $U = 1$, and η can be set as $\eta \leq \frac{1}{2}$ (see Lemma 4.4.2 and the paragraph below it). In this case, our algorithm satisfies the following bound:

Corollary 4.3.1. *Suppose that Assumption 1 holds, W and Ψ are the identity maps, and $\eta \leq 1/2$. Then, the average reward ρ_K of the policy output by our algorithm satisfies*

$$\rho^* - \rho_K \leq \frac{\tau_{mix}^2 |\mathcal{X}| + \log(|\mathcal{X}||\mathcal{A}|)}{\eta K}.$$

In particular, setting $\eta = 1/2$, the bound becomes $\rho^* - \rho_K = \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}^2 |\mathcal{X}|}{K}\right)$.

A brief inspection of Equations (4.6)-(4.7) suggests that each update of our algorithm can be computed in $\mathcal{O}(mn)$ time, the most expensive operation being computing the matrix-vector products $W(P-E)\Psi\theta$ and $y^\intercal W(P-E)\Psi$. While this suggests that the algorithm may have runtime and memory complexity independent of the size of the state space, we note that exact computation of the matrix $W(P-E)\Psi$ can still take $\mathcal{O}(|\mathcal{X}|^2|\mathcal{A}|)$ time in the worst case. This can be improved to $\mathcal{O}(K)$ when assuming that only K entries of the transition matrix P are nonzero, which can be of order $|\mathcal{X}||\mathcal{A}|$ in many interesting problems where the support of $P(\cdot|x,a)$ is of size $\mathcal{O}(1)$ for all x, a. We stress however that the matrix $W(P-E)\Psi$ only needs to be computed *once* as an initialization step of our algorithm. In contrast, a general algorithm like value iteration needs at least $\Theta(K) = \Theta(|\mathcal{X}||\mathcal{A}|)$ for computing *each update*, showing a clear computational advantage of our method. Further discussion of computational issues is deferred to Section 4.6.

4.4 The proof of Theorem 4.3.1

This section provides an outline of the analysis of our algorithm that will culminate with the proof of Theorem 4.3.1. At a high level, the analysis builds on some well-known results regarding the performance of mirror prox, including a classical bound on the *duality gap* of the obtained solutions. The crucial challenge posed by our setting is connecting the duality gap on the saddle-point problem to a suboptimality gap of the extracted policies. To face this problem, we will show two alternative approaches.

In what follows, we first provide some general tools regarding mirror prox that will be helpful throughout the proofs, and then provide the two alternative methods to connect the duality gap with the suboptimality gap of the extracted policies. Missing proofs are provided in Section 4.7.

A central piece of our analysis are the results of Section 2.4.2 regarding the iterates of mirror prox. We state here their analogous versions for our saddle-point problem for clarity. The following lemma presents the result analogous to Theorem 2.4.3 regarding our problem:

Lemma 4.4.1. *Let Φ be σ-strongly convex and g be L-Lipschitz. Then, for all k, mirror prox guarantees*

$$\eta \left\langle \widehat{z}_{k+1} - z, g(\widehat{z}_{k+1}) \right\rangle \leq D(z\|z_k) - D(z\|z_{k+1}) - \frac{\sigma - \eta L}{4} \left\| z_{k+1} - z_k \right\|^2 .$$

holds for every $z \in \mathcal{Z}$ and $t > 0$.

The proof is the same as for Theorem 2.4.3 since the only needed condition is the monotonicity of $g(x)$ that is clearly satisfied to use the first order optimality equation. We can now state the corollaries analogous to 2.4.1 and 2.4.2. The first one shows that the iterates remain bounded during the optimization procedure:

Corollary 4.4.1. *Let $z^* = (\theta^*, y^*)$ be any solution to $\max_y \min_\theta \widetilde{\mathcal{L}}(\theta, y)$ and suppose that the conditions of Lemma 4.4.1 hold and that $\eta \leq \frac{\sigma}{L}$. Then, for all k, mirror prox guarantees*

$$D(z^*\|z_k) \leq D(z^*\|z_0).$$

The proof of this corollary is exactly the same as the one of 2.4.1. The second corollary establishes a bound on the *duality gap* evaluated at $(\overline{\theta}_K, \overline{y}_K)$:

Corollary 4.4.2. *Let $z = (\theta, y) \in \mathcal{Z}$ be arbitrary and assume that $\eta \leq \frac{\sigma}{L}$. Then, mirror prox guarantees the following bound on the duality gap:*

$$\widetilde{\mathcal{L}}\left(\overline{\theta}_K, y\right) - \widetilde{\mathcal{L}}\left(\theta, \overline{y}_K\right) \leq \frac{D(z\|z_0)}{\eta K}.$$

Proof. The proof of this corollary is slightly different since we have to take into account that we are in a saddle-point problem. Since $\widetilde{\mathcal{L}}(u, y)$ is bilinear,

$$\langle \widehat{z}_{k+1} - z, g(\widehat{z}_{k+1}) \rangle = \widetilde{\mathcal{L}}\left(\widehat{u}_{k+1}, y\right) - \widetilde{\mathcal{L}}\left(\theta, \widehat{y}_{k+1}\right).$$

Then,

$$\begin{aligned}
&\widetilde{\mathcal{L}}\left(\overline{\theta}_K, y\right) - \widetilde{\mathcal{L}}\left(\theta, \overline{y}_K\right) \\
&\qquad = \sum_{k=0}^{K} \frac{1}{K} \left(\widetilde{\mathcal{L}}\left(\widehat{u}_{k+1}, y\right) - \widetilde{\mathcal{L}}\left(\theta, \widehat{y}_{k+1}\right) \right) \\
&\qquad \leq \frac{1}{\eta K} \sum_{k=0}^{K} \left(D(z\|z_k) - D(z\|z_{k+1}) - \frac{\sigma - \eta L}{4} \|z_{k+1} - z_k\|^2 \right) \\
&\qquad \leq \frac{1}{\eta K} D(z\|z_0),
\end{aligned}$$

where in the first inequality we used the bound in Lemma 4.4.1. □

In order to apply these tools to our problem, we first need to confirm that our objective is smooth (remember that we already saw that Φ is 1-strongly convex on $\mathcal{Z}$ w.r.t. $\|z\|^2$), i.e., that g is Lipschitz, with respect to the norm $\|z\|^2 = \|\theta\|_2^2 + \|y\|_1^2$. The following lemma establishes this property.

Lemma 4.4.2. *Let $C = \max_x \|\Psi_{x,\cdot}\|_1$. Then, the function $\widetilde{\mathcal{L}}$ is $2C$-smooth (and g is $2C$-Lipschitz) with respect to $\|\cdot\|$.*

The proof is provided in Section 4.7.1. Notably, this lemma implies that $\widetilde{\mathcal{L}}$ is 2-smooth when the rows of Ψ form probability distributions. In the worst case, however, when we only assume that the entries of Ψ are bounded in absolute value by 1, the smoothness constant can be as large as $2m$. In order to ensure that $\eta \leq \frac{\sigma}{L}$, in what follows, we will assume that $\eta \leq 1/(2C)$.

At this point of the analysis, we present two alternative paths to show similar results on the convergence rate of our algorithm. The first approach is based on a smart choice for the comparator point involved in the definition of the duality gap, which gives a direct connection with the suboptimality gap of the extracted policies. This technique is inspired by the work of Cheng et al. [2020], that to the best of our knowledge, were the first ones to use this technique (along with Jin and Sidford [2020] who rediscovered the same trick independently a few months later). This approach gives the actual rate seen in Theorem 4.3.1. After this, we will present the other approach that is based on exploiting further properties of mirror prox. The second approach is the one used in the paper of Bas-Serrano and Neu [2020] where most of the results of this chapter are presented. As we will see, the first approach is more elegant, direct and gives a tighter bound, but we think that the second one is still relevant since it presents useful tools for analyzing similar algorithms.

4.4.1 Method 1: exploiting the duality gap

This approach is based on taking a carefully chosen comparator point for evaluating the duality gap, and then using the bound on this quantity achieved by mirror prox to bound the policy suboptimality. In particular, we let V^{π_K} be the value function of policy π_K and θ^{π_K} such that

$$\left\langle (P-E)^\mathsf{T} W^\mathsf{T} \bar{y}_K, \Psi\theta^{\pi_K} \right\rangle = \left\langle (P-E)^\mathsf{T} W^\mathsf{T} \bar{y}_K, V^{\pi_K} \right\rangle,$$

where the existence of θ^{π_K} is guaranteed by Assumption 3. Notably, the proof does not need realizability of the value function V^{π_K} in a sense that is stricter than the above condition, and in particular not even realizability of V^* is required. However, realizability of μ^* is still needed. Under these conditions, the following lemma connects the duality gap at the point (θ^{π_K}, y^*), with the suboptimality gap of the policy π_K:

Lemma 4.4.3. $\widetilde{\mathcal{L}}(\bar{\theta}_K, y^*) - \widetilde{\mathcal{L}}(\theta^{\pi_K}, \bar{y}_K) = \rho^* - \rho$

Proof.

$$\begin{aligned}
&\widetilde{\mathcal{L}}(\bar{\theta}_K, y^*) - \widetilde{\mathcal{L}}(\theta^{\pi_K}, \bar{y}_K) \\
&\quad = \left\langle W^\mathsf{T} y^*, r + (P-E)\Psi\bar{\theta}_K \right\rangle - \left\langle W^\mathsf{T}\bar{y}_K, r + (P-E)\Psi\theta^{\pi_K} \right\rangle \\
&\quad = \langle \mu^*, r\rangle + \left\langle \mu^*, (P-E)\Psi\bar{\theta}_K \right\rangle - \left\langle W^\mathsf{T}\bar{y}_K, r + (P-E)V^{\pi_K} \right\rangle \\
&\quad = \rho^* - \sum_{x,a} \nu_K(x)\pi_K(a|x) \left(r(x,a) + \sum_{x'} P(x'|x,a)V^{\pi_K}(x') - V^{\pi_K}(x) \right) \\
&\quad = \rho^* - \rho^{\pi_K}
\end{aligned}$$

where ν_K is defined as $\nu_K(x) = \sum_a (W^\mathsf{T} y_K)(x,a)$ and in the last step we have used the Bellman equations for policy π_K:

$$V^{\pi_K}(x) + \rho^{\pi_K} = \sum_a \pi_K(a|x) \left(r(x,a) + \sum_{x'} P(x'|x,a)V^{\pi_K}(x') \right).$$

□

Then, putting together the results of Lemma 4.4.3 and Corollary 4.4.2, and using that $D(z^*\|z_0) \leq \frac{1}{2}U^2\tau_{mix}^2 m + \log(n)$ we conclude the proof.

4.4.2 Method 2: exploiting mirror prox properties

We start appealing to the realizability assumption to choose $z = (\theta^*, y^*)$ such that $(V^*, \mu^*) = (\Psi\theta^*, W^\intercal y^*)$, and observe that

$$\begin{aligned}\widetilde{\mathcal{L}}\left(\overline{\theta}_K, y^*\right) - \widetilde{\mathcal{L}}\left(\theta^*, \overline{y}_K\right) &= \left\langle \mu^*, (P-E)\Psi\overline{\theta}_K + r\right\rangle - \left\langle W^\intercal \overline{y}_K, (P-E)V^* + r\right\rangle \\ &\leq \frac{D(z^*\|z_0)}{\eta K}\end{aligned}$$

holds by virtue of Corollary 4.4.2 and the choice of $\eta \leq 1/(4C)$. Observing that $(P-E)^\intercal \mu^* = 0$ holds due to the stationarity of μ^* and reordering gives

$$\left\langle \mu^* - W^\intercal \overline{y}_K, r\right\rangle \leq \frac{D(z^*\|z_0)}{\eta K} + \left\langle (P-E)^\intercal W^\intercal \overline{y}_K, V^*\right\rangle. \tag{4.8}$$

The remaining key question is how to relate $\langle W^\intercal \overline{y}_K, r\rangle$ to the true average reward ρ_K associated with the extracted policy. This is done with the help of the following lemma, one of the key results of this second method:

Lemma 4.4.4. *Suppose that Assumption 1 holds. Let μ be an arbitrary distribution over $\mathcal{X} \times \mathcal{A}$ and let π^μ be the policy extracted from μ. Then, the average reward ρ^μ induced by π^μ satisfies $\langle \mu, r\rangle - \rho^\mu \leq \tau_{mix} \left\|(P-E)^\intercal \mu\right\|_1$.*

The proof is provided in Section 4.7.2. It is interesting to see that the above lemma says that for a given probability distribution over states and actions, the difference between $\langle \mu, r\rangle$ and the actual average reward of the policy extracted from μ is related to "how much stationary" the distribution μ is. Using this result with the distribution $W^\intercal \overline{y}_K$ gives

$$\langle W^\intercal \overline{y}_K, r\rangle - \rho_K \leq \tau_{mix} \left\|(P-E)^\intercal W^\intercal \overline{y}_K\right\|_1$$

Combining this result with the bound of Equation 4.8 and using that $\|V^*\|_\infty \leq \tau_{mix}$, we obtain

$$\rho^* - \rho_K \leq \frac{D(z^*\|z_0)}{\eta K} + 2\tau_{mix} \left\|(P-E)^\intercal W^\intercal \overline{y}_K\right\|_1. \tag{4.9}$$

Thus, it only remains to bound $\left\|(P-E)^\intercal W^\intercal \overline{y}_K\right\|_1$. In order to do this, we crucially use Assumption 3 that guarantees the coherence of the feature maps to prove the following result:

Lemma 4.4.5. *Suppose that Assumptions 2 and 3 hold. Then,*

$$\tau_{mix} \left\|(P-E)^{\intercal} W^{\intercal} \bar{y}_K\right\|_1 \leq \frac{5\tau_{mix}^2 U^2 m + 3\log n}{\eta K}$$

The proof of this lemma is provided in Section 4.7.3. Combining the bound of this lemma with Equation (4.9) and using $D(z^*\|z_0) \leq U^2\tau_{mix}^2 m + \log{(n)}$ we get

$$\rho^* - \rho_K \leq \frac{11\tau_{mix}^2 U^2 m + 7\log n}{\eta K}.$$

In particular, setting η to the maximum allowed in statement of Theorem 4.3.1, $\eta = \frac{1}{4m}$, the bound becomes $\rho^* - \rho_K = \mathcal{O}\left(\frac{\tau_{mix}^2 m^2 U^2}{K}\right)$.

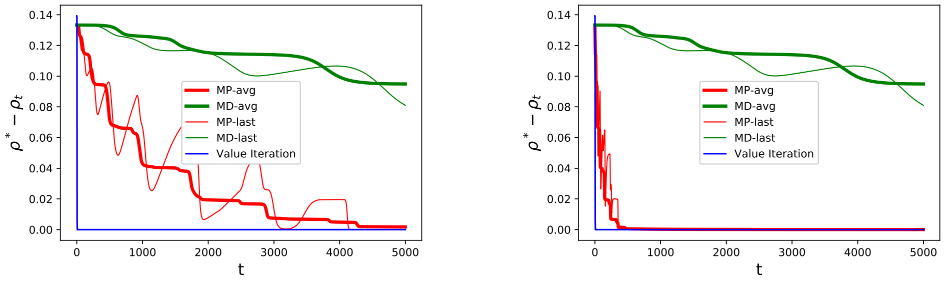
4.5 Numerical illustration

In this section, we provide empirical results on two simple environment in order to illustrate our theoretical results, and specifically compare the performance of our algorithm with that of mirror descent and the classic value iteration algorithm.

In the first example, we consider a rectangular $s \times s$ grid with one nonzero reward placed in state x_r, so that $r(x,a) = \mathcal{I}_{x=x_r}$. Once the agent arrives to x_r, it is randomly teleported to any of the other states with equal probability. In any other state, the agent can decide to move to a neighboring cell in any direction. The attempt to move to the desired direction is successful with probability p, and the agent moves to the opposite direction with probability $1-p$. If the agent is in an edge of the grid and makes an step in the direction of the edge, it appears in the opposite edge.

Figure 4.2 shows some results on a grid of side $s = 10$, in the case when no features are used, so we optimize over the whole state-action space. We observe that the convergence of mirror prox is much faster than that of mirror descent, and that the last iterate of mirror prox converges very quickly to the optimum, achieving it after *finitely many iterations*. We also note that for higher values of η than the ones found to be safe in our bounds (at most 1/4), the algorithm is still stable and can lead to faster convergence to the optimum.

In our second example, we show how the usage of good features can make mirror prox converge faster than value iteration. We consider a sequence of states of length L (see Figure 4.3) with one nonzero reward placed in the first state so that $r_{(x,a)} = I_{x=x_1}L$. In states x_2 to x_{N-1} the available actions are to go left and right, in state x_1 the only available action is to go to the last state (x_L), and in state x_L the only available action is to go left. Each action has a probability p of success and $1-p$ of remaining in the same state.



(a) $p = 0.9$, and $\eta = \frac{1}{4}$.

(b) $p = 0.9$, and $\eta = 3$.

Figure 4.2: Regret as a function of the number of iterations of mirror prox (MP), mirror descent (MD), and value iteration in a grid world example with side $s = 10$. The suffix "-avg" refers to the average over iterations (policy π_K) while "-last" refers to last iteration (policy π^{y_K}).

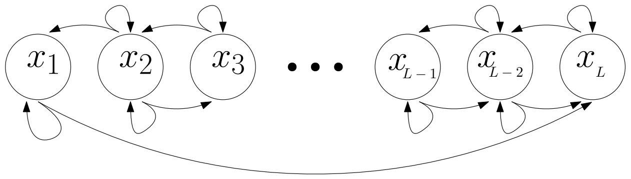


Figure 4.3: Example of MDP.

Let's first realize that the optimal policy is to take always the left action, so the optimal distribution is homogeneous and the optimal value function decreases linearly with the distance to the leftmost state.

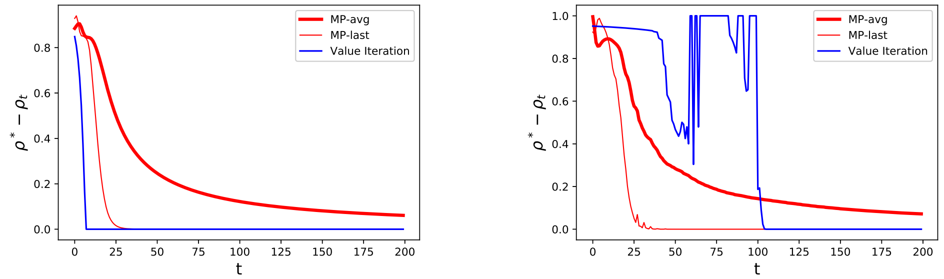
To test our algorithm in this environment, we built W and Ψ taking advantage of the structure of the problem as follows:

Let's start with W, that will be a matrix of 8 rows. We first randomly generate a vector c of length L with entries being 1, 2 or 3. We make $W^\top_{(x=i,a=\text{left}),j}$=1 if $c(j) = i$ and 0 otherwise for $i = 1, 2, 3$. After that we normalize the three rows, getting three homogeneous non-overlapping distributions. Doing this, we ensure that the realizability assumption is fulfilled for the probability distribution μ. We do the same for the "*right*" action to fill the next three rows $i = 4, 5, 6$, and we fill the last two rows with random probability distributions over the whole set of state-action pairs. This makes a total of 8 rows in W.

To build Ψ, we also randomly generate a vector c of length L with entries being 1, 2 or 3 and we make $\Psi_{j,i} = j/L$ if $c(j) = j$ and 0 otherwise for $i = 1, 2, 3$. With this we guarantee that the realizability assumption is fulfilled for the value functions. We also add three random columns with random numbers between 0 and 1, in order to fulfill coherence with high probability. This results in a total of

5 columns for Ψ.

In Figure 4.4 we show the results obtained with value iteration and the linearly relaxed mirror prox, with $p = 0.7$ and different lengths (10 and 100). While for value iteration the number of iterations needed to converge is of the order of the number of states, it is independent of the size of the state space for our algorithm, and rather scales with the number of columns of the matrices W and Ψ. This simple example shows that with proper features, our algorithm can actually beat value iteration, which by itself is not able to deal with features.



(a) $p = 0.7$, $\eta = \frac{1}{4}$ and $L = 10$.

(b) $p = 0.7$, $\eta = \frac{1}{4}$ and $L = 100$.

Figure 4.4: Suboptimality gap as a function of the number of iterations of mirror prox and value iteration for $p = 0.7$ and $\eta = 0.25$

4.6 Conclusions

Our most important contributions concern the relaxed saddle-point problem (4.4), most notably including our discussion on the necessity and sufficience of the coherence assumption (Assumption 3). As we have mentioned earlier, several relaxation schemes similar to ours have been studied in the literature. In fact, relaxing the linear program underlying (4.3) through the introduction of the feature map Ψ for approximating the value function V^* is one of the oldest ideas in approximate dynamic programming, originally introduced by Schweitzer and Seidmann [1985]. The effects of this approximation were studied by de Farias and Van Roy [2003] in the context of discounted Markov decision processes. A relaxation scheme involving both the feature maps Ψ and W was considered by Lakshminarayanan and Bhatnagar [2015]; Lakshminarayanan et al. [2017]. Both sets of authors carefully observed that introducing relaxations may make the linear program unbounded, and proposed algorithmic steps and structural assumptions of Ψ and W to fight this issue. The results of these works are incomparable to ours since they focus on controlling the errors in approximating the optimal value function

V^* rather than controlling the suboptimality of the policies output by the algorithm. Interestingly, the widely popular REPS algorithm of Peters et al. [2010] is also originally derived from the relaxed linear program analyzed by de Farias and Van Roy [2003], even if this connection has not been pointed out by the authors.

The work of Chen et al. [2018] is very close to ours in spirit. Chen et al. consider a variation of the relaxed saddle-point problem (4.4) with W being block-diagonal with Ψ^T in each of its blocks, and claim convergence results for their algorithm to the optimal policy under only a realizability assumption. Unfortunately, their choice of W does not necessarily ensure that the coherence assumption holds, which raises concerns regarding the generality of their guarantees. Indeed, the results of Chen et al. require an additional assumption that implies that $\frac{\max_x \nu_\pi(x)}{\min_{x'} \nu_\pi(x')}$ remains bounded by a constant for any policy π, which is extremely difficult to ensure in problems of practical interest. In fact, this ratio is already exponentially large in $|\mathcal{X}|$ in very simple problems like the one we consider in our experiments. Additionally, there is a subtle issue with the analysis of Chen et al.: it is based on the claim that under the realizability assumption, the representation (θ^*, y^*) of the original optimal solution $(V^*, \mu^*) = (\Psi\theta^*, W^\mathsf{T} y^*)$ always remains an optimal solution to the relaxed saddle-point problem. We could not confirm that this claim is indeed true, or to what extent their condition regarding the boundedness of stationary distribution can be relaxed.

In any case, we believe that our coherence assumption is more fundamental than the previously considered conditions, and it enables a much more transparent analysis of optimization algorithms addressing the relaxed saddle-point problem (4.4). Beyond this particular positive result, our work also cleans the slate for further theoretical work on approximate optimization in Markov decision processes. Indeed, the form of our coherence assumption naturally invites the question: can we compute good approximate solutions to the original problem when our assumptions are only satisified approximately? Similar questions are not without precedent in the reinforcement-learning literature. Translated to our notation, classical results concerning the performance of (least-squares) temporal difference learning algorithms imply that the approximation errors are controlled by the projection error of $(P - E)\Psi\theta^* + r$ to the column space of Ψ [Tsitsiklis and Van Roy, 1997; Bradtke and Barto, 1996; Lazaric et al., 2010]. When using more general function classes to approximate V^*, Munos and Szepesvári [2008] show that the approximation errors are controlled by the *inherent Bellman error* of the function class, which captures an approximation property related to our coherence condition. Whether or not we can generalize our techniques to construct provably efficient algorithms under such milder assumptions remains an exciting open problem that we leave open for future research.

4.7 Omitted proofs

This section contains the proofs of those lemmas used in Section 4.4 to prove Theorem 4.3.1.

4.7.1 The proof of Lemma 4.4.2

We start by noticing that the dual norm of $\|z\|^2 = \|\theta\|_2^2 + \|y\|_1^2$ evaluated at $x = (w, q)$ is $\|x\|_*^2 = \|w\|_2^2 + \|q\|_\infty^2$. Recalling the definition of smoothness, and the statement of the lemma that we aim to proof, we see that we need to show the following:

$$\|g(z) - g(z')\|_* \leq 2C\,\|z - z'\| = \sqrt{4C^2\,\|y - y'\|_1^2 + 4C^2\,\|\theta - \theta'\|_2^2}.$$

Using the definition of $g(z)$ and the shorthand notation $A = W(P - E)\Psi$, for any $z = (\theta, y)$ and $z' = (\theta', y')$ we have

$$\|g(z) - g(z')\|_*^2 = \|A^\mathsf{T}(y - y')\|_2^2 + \|A(\theta - \theta')\|_\infty^2\,.$$

We now have to bound the two terms of the right hand side of the above equation. Let's first see that the sum of any column j of $\Psi^\mathsf{T} W^\mathsf{T}$ is bounded by C:

$$\sum_i |(\Psi^\mathsf{T} W^\mathsf{T})_{i,j}| = \sum_{i,x} |\Psi^\mathsf{T}_{i,x} W^\mathsf{T}_{x,j}| = \sum_x |W^\mathsf{T}_{x,j}| \left(\sum_i |\Psi^\mathsf{T}_{i,x}|\right) \leq C. \tag{4.10}$$

The same can be easily proven for the matrix $\Psi^\mathsf{T} P^\mathsf{T} W^\mathsf{T}$. Now, the first term can be bounded as follows

$$\begin{aligned}
\|A^\mathsf{T}(y - y')\|_2 \leq \|A^\mathsf{T}(y - y')\|_1 & \\
=& \sum_i |\sum_j A^\mathsf{T}_{i,j}\,(y(j) - y'(j))\,| \\
\leq& \sum_{i,j} |A^\mathsf{T}_{i,j}||y(j) - y'(j)| \\
\leq& \sum_j \left(\sum_i |A^\mathsf{T}_{i,j}|\right) |y(j) - y'(j)| \\
\leq& \sum_j \left(\sum_i |(\Psi^\mathsf{T} P^\mathsf{T} W^\mathsf{T})_{i,j}|\right) |y(j) - y'(j)| \\
&\quad + \sum_j \left(\sum_i |(\Psi^\mathsf{T} W^\mathsf{T})_{i,j}|\right) |y(j) - y'(j)| \\
\leq& 2C\,\|y - y'\|_1\,.
\end{aligned}$$

To bound the last term, we observe that

$$\begin{aligned}
\|W(P-E)\Psi(\theta-\theta')\|_\infty^2 &= \max_j \left| \sum_i \left(W\Psi - WP\Psi\right)_{j,i} (\theta(i)-\theta'(i)) \right|^2 \\
&\le \max_j \left| \sum_i \left(\left|(W\Psi)_{j,i}\right| + \left|(WP\Psi)_{j,i}\right| \right) (\theta(i)-\theta'(i)) \right|^2 \\
&\le \max_j \left| \sum_i \left(\left|(W\Psi)_{j,i}\right| + \left|(WP\Psi)_{j,i}\right| \right) \|\theta-\theta'\|_\infty \right|^2 \\
&\le \max_j \left| \sum_i \left(\left|(\Psi^\intercal W^\intercal)_{i,j}\right| + \left|(\Psi^\intercal P^\intercal W^\intercal)_{i,j}\right| \right) \|\theta-\theta'\|_\infty \right|^2 \\
&\le \left|2C \|\theta-\theta'\|_\infty\right|^2 \le 4C^2 \|\theta-\theta'\|_\infty^2 \le 4C^2 \|\theta-\theta'\|_2^2 ,
\end{aligned}$$

where in the fourth inequality we have used the expression (4.10). Putting everything together concludes the proof.

4.7.2 The proof of Lemma 4.4.4

Let ν be such that $\nu(x) = \sum_a \mu(x,a)$ and ν^μ the stationary distribution induced by π^μ.

$$\langle \mu, r \rangle - \rho^\mu = \sum_{x,a} \left(\nu(x) - \nu^\mu(x)\right) \pi(a|x) r(x,a) \le \|\nu - \nu^\mu\|_1 ,$$

so all we are left with is bounding the total variation distance between ν^μ and ν. To do this, we start by fixing an arbitrary $k > 0$ and observing that

$$\begin{aligned}
\left\|\left(\nu - \nu^\mu\right) P_\pi^k\right\|_1 &\le Ce^{-k/\tau} \|\nu - \nu^\mu\|_1 \\
&\le Ce^{-k/\tau} \left(\left\|\nu - \nu P_\pi^k\right\|_1 + \left\|\nu P_\pi^k - \nu^\mu\right\|_1 \right) ,
\end{aligned} \tag{4.11}$$

where we used Assumption 1 in the first step and the triangle inequality in the second one. Regarding the first term in the parentheses, we repeatedly use the triangle inequality to obtain

$$\begin{aligned}
\left\|\nu - \nu P_\pi^k\right\|_1 &\le \|\nu - \nu P_\pi\|_1 + \left\|\nu P_\pi - \nu P_\pi^2\right\|_1 + \dots + \left\|\nu P_\pi^{k-1} - \nu P_\pi^k\right\|_1 \\
&= \|\nu - \nu P_\pi\|_1 + \|(\nu - \nu P_\pi) P_\pi\|_1 + \dots + \left\|(\nu - \nu P_\pi) P_\pi^{k-1}\right\|_1 \\
&\le \|\nu - \nu P_\pi\|_1 + Ce^{-1/\tau} \|\nu - \nu P_\pi\|_1 + \dots + Ce^{-(k-1)/\tau} \|\nu - \nu P_\pi\|_1 \\
&\le C \|\nu - \nu P_\pi\|_1 \sum_{i=0}^{k-1} e^{-i/\tau} \le \frac{C}{1 - e^{-1/\tau}} \|\nu - \nu P_\pi\|_1 .
\end{aligned}$$

Plugging this bound into Equation 4.11 and observing that $\nu P_\pi^k - \nu^\mu = (\nu - \nu^\mu) P_\pi^k$ due to stationarity of ν^μ, we get

$$\left\|(\nu - \nu^\mu) P_\pi^k\right\|_1 \le Ce^{-k/\tau} \left(\frac{C}{1 - e^{-1/\tau}} \left\|\nu - \nu P_\pi\right\|_1 + \left\|(\nu - \nu^\mu) P_\pi^k\right\|_1 \right).$$

Reordering gives

$$\left\|(\nu - \nu^\mu) P_\pi^k\right\|_1 \le \frac{Ce^{-k/\tau}}{1 - Ce^{-k/\tau}} \cdot \frac{C}{1 - e^{-1/\tau}} \left\|\nu - \nu P_\pi\right\|_1.$$

Thus, using the triangle inequality again yields

$$\begin{aligned}\left\|\nu - \nu^\mu\right\|_1 &\le \left\|\nu - \nu P_\pi^k\right\|_1 + \left\|\nu P_\pi^k - \nu^\mu\right\|_1 \\ &\le \left(1 + \frac{Ce^{-k/\tau}}{1 - Ce^{-k/\tau}}\right) \frac{C}{1 - e^{-1/\tau}} \left\|\nu - \nu P_\pi\right\|_1.\end{aligned}$$

Now, choosing any $k \ge \tau \log(2C)$, using the elementary inequality $1/(1 - e^{-1/\tau}) \le \tau + 1$ and recalling the definition of $\tau_{mix} = 2C(1+\tau)$ concludes the proof. □

4.7.3 The proof of Lemma 4.4.5

The statement is obvious when $(P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K = 0$, so we will assume that the contrary holds below. Let us define

$$w = \tau_{mix} \cdot \underset{V : \|V\|_\infty = 1}{\arg\max} \left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, V \right\rangle,$$

noting that $\left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, w \right\rangle = \tau_{mix} \left\|(P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K\right\|_1 > 0$. By using this fact and Assumption 3, we crucially observe that there exists a $\widetilde{\theta}$ such that $\left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, w \right\rangle = \left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, \Psi \widetilde{\theta} \right\rangle$ and $\left\|\widetilde{\theta}\right\|_\infty \le \tau_{mix} U$. This implies that we can apply Corollary 4.4.2 with $z = (\overline{\theta}_K - \widetilde{\theta}, \overline{y}_K)$ to obtain the bound

$$\begin{aligned}\left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, w \right\rangle =& \left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, \Psi \overline{\theta}_K \right\rangle + \left\langle W^\mathsf{T} \overline{y}_K, r \right\rangle \\ & - \left\langle (P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K, \Psi \left(\overline{\theta}_K - \widetilde{\theta}\right) \right\rangle - \left\langle W^\mathsf{T} \overline{y}_K, r \right\rangle \\ \le& \frac{D(z \| z_0)}{\eta K}.\end{aligned}$$

Plugging in the definition of w and the Bregman divergence D_Φ, we obtain

$$\left\|(P - E)^\mathsf{T} W^\mathsf{T} \overline{y}_K\right\|_1 \le \frac{\frac{1}{2}\left\|\widetilde{\theta} - \overline{\theta}_K\right\|_2^2 + \log n}{\eta \tau_{mix} K}.$$

Due to Assumption 2 and our assumption on Ψ stated before Theorem 4.3.1, we can choose an optimal solution θ^* satisfying $\Psi\theta^* = V^*$ and $\left\|\theta^*\right\|_\infty \leq \tau_{mix}U$ and write

$$\begin{aligned}\left\|\widetilde{\theta} - \overline{\theta}_K\right\|_2^2 \leq 2\left\|\widetilde{\theta} - \theta^*\right\|_2^2 + 2\left\|\overline{\theta}_K - \theta^*\right\|_2^2 \leq 4\left\|\widetilde{\theta}\right\|_2^2 + 4\left\|\theta^*\right\|_2^2 + 4D(z^*\|\overline{z}_K)\\ \leq 4m\left\|\widetilde{\theta}\right\|_\infty^2 + 4m\left\|\theta^*\right\|_\infty^2 + 4D(z^*\|z_0)\\ \leq 10\tau_{mix}^2U^2m + 4\log n,\end{aligned}$$

where in the second line we have used Corollary 4.4.1 that implies $D(z^*\|\overline{z}_K) \leq D(z^*\|z_0)$. Putting everything together concludes the proof. □

Chapter 5

LOGISTIC Q-LEARNING

5.1 Introduction

Despite the enormous empirical successes of deep reinforcement learning, we understand little about the convergence of the algorithms that are commonly used. The use of the empirical squared Bellman error (squared Bellman error for short) for deep reinforcement learning has been popularized in the breakthrough paper of Mnih et al. [2015] (see Section 3.4.2), and has been *exclusively* used for policy evaluation ever since. Despite its broad usage, it has a number of undesirable properties: it is not directly motivated by standard Markov Decision Processes (MDP) theory, not convex in the action-value function parameters, and RL algorithms based on its recursive optimization are known to be unstable [Geist et al., 2017; Mehta and Meyn, 2020]. While several algorithmic improvements have been proposed for improving policy updates over the past few years, the squared Bellman error remained a staple: among others, it is used for policy evaluation in TRPO [Schulman et al., 2015], SAC [Haarnoja et al., 2018], A3C [Mnih et al., 2016], TD3 [Fujimoto et al., 2018], MPO [Abdolmaleki et al., 2018] and POLITEX [Abbasi-Yadkori et al., 2019]. Despite its extremely broad use, the squared Bellman error suffers from a range of well-known issues pointed out by several authors including Sutton and Barto [2018, Chapter 11.5], Geist et al. [2017], and Mehta and Meyn [2020]. While some of these have been recently addressed by Dai et al. [2018] and Feng et al. [2019], several concerns remain.

On the other hand, the RL community has been very productive in developing novel policy-improvement rules: since the seminal work of Kakade and Langford [2002] established the importance of soft policy updates for dealing with policy-evaluation errors, several practical update rules have been proposed and applied successfully in the context of deep RL—see the list we provided in the previous paragraph. Many of these soft policy updates are based on the idea of *entropy reg-*

ularization, first explored by Kakade [2001] and Ziebart et al. [2008] and inspiring an impressive number of followup works eventually unified by Neu et al. [2017] and Geist et al. [2019]. A particularly attractive feature of entropy-regularized methods is that they often come with a closed-form "softmax" policy update rule that is easily expressed in terms of an action-value function. A limitation of these methods is that they typically do not come with a theoretically well-motivated loss function for estimating the value functions and end up relying on the squared Bellman error.

One notable exception is the Relative Entropy Policy Search (`REPS`) algorithm of Peters et al. [2010] that comes with a natural loss function for policy evaluation, but no tractable policy-update rule. `REPS` is elegantly derived from the LP formulation of optimal control in MDPs, but it has the serious shortcoming that its faithful implementation requires access to the true MDP for both the policy evaluation and improvement steps, even at deployment time. The usual way to address this limitation is to use an empirical approximation to the policy evaluation step and to project the policy from the improvement step into a parametric space [Deisenroth et al., 2013], losing all the theoretical guarantees of `REPS` in the process.

In this chapter, we present a new algorithm called `Q-REPS` that eliminates this limitation of `REPS` by introducing a simple softmax policy improvement step expressed in terms of an action-value function that naturally arises from a regularized LP formulation. The action-value functions are obtained by minimizing a convex loss function that we call the *logistic Bellman error* (LBE) due to its analogy with the classic notion of Bellman error and the logistic loss for logistic regression. The LBE has numerous advantages over the most commonly used notions of Bellman error: unlike the squared Bellman error, the logistic Bellman error is convex in the action-value function parameters, smooth, and has bounded gradients (see Figure 5.1). This latter property obviates the need for the heuristic technique of gradient clipping (or using the Huber loss in place of the square loss), a commonly used optimization trick to improve stability of training of deep RL algorithms [Mnih et al., 2015].

We also present an empirical version of the LBE and provide a bound on its bias in terms of the regularization parameters used in `Q-REPS`. Furthermore, we propose a semi-empirical version of the LBE (using a simulator) that is an unbiased estimate of the true LBE.

Our main theoretical contribution is an error-propagation analysis that relates the quality of the optimization subroutine to the quality of the policy output by the algorithm. For a version of the algorithm minimizing the semi-empirical LBE we also provide rigorous theoretical guarantees that establish its convergence to the optimal policy under appropriate conditions.

Our error propagation analysis is close in spirit to that of Scherrer et al. [2015], recently extended to entropy-regularized approximate dynamic programming algorithms by Geist et al. [2019], Vieillard et al. [2020a], and Vieillard et al. [2020b]. One major difference between our approaches is that their guarantees depend on the ℓ_p norms of the policy evaluation errors, but still optimize squared-Bellman-error-like quantities that only serve as proxy for these errors. In contrast, our analysis studies the propagation of the optimization errors on the objective function that is *actually optimized* by the algorithm.

Our main algorithmic contribution is a saddle-point optimization framework for optimizing the empirical and semi-empirical versions of the LBE. It formulates the minimization problem as a two-player game between a *learner* and a *sampler*. The learner plays stochastic gradient descent (SGD) on the samples proposed by the sampler, and the sampler updates its distribution over the sample transitions in response to the observed Bellman errors. We evaluate the resulting algorithm experimentally on a range of standard benchmarks, showing excellent empirical performance of Q-REPS with minimization of the empirical LBE.

Furthermore, since our Q-REPS comes with both a natural loss function and an explicit and tractable policy update rule, it is possible to implement Q-REPS *entirely faithfully* to its theoretical specification in a deep reinforcement learning context, modulo the step of using a neural network for parameterizing the Q-function. This implementation is justified by our error propagation analysis accounting for the optimization and representation errors.

Some of the results of this chapter regarding the analysis of Q-REPS are different from the results of the original paper [Bas-Serrano et al., 2021]. This is because several details in the original proofs were incorrect and the final result does not hold in the form claimed in the paper. The new analysis presented here amends the mistakes in the original work, albeit at the price of some more restrictive assumptions. It remains an open problem to relax these assumptions and in particular prove a performance guarantee for the case of linear function approximation.

5.2 REPS

The results shown in this chapter are directly inspired by the seminal *relative entropy policy search* (REPS) algorithm proposed by Peters et al. [2010]. The core ideas underlying REPS are adding a strongly convex regularization function to the objective of the LP (3.7) and relaxing the primal constraints through the use of a feature map $\psi : \mathcal{X} \to \mathbb{R}^m$. Introducing the operator $\Psi^\intercal$ acting on $q \in \mathbb{R}^{\mathcal{X}}$ as $\Psi^\intercal q = \sum_x q(x)\psi(x)$, and letting μ_0 be an arbitrary state-action distribution, REPS is defined as an iterative optimization scheme that produces a sequence of occupancy measures as follows:

$$\begin{aligned} \mu_{k+1} = \max_{\mu \in \mathbb{R}_+^{\mathcal{X}\times\mathcal{A}}} \quad & \langle \mu, r \rangle - \frac{1}{\eta} D\left(\mu \| \mu_k\right) \\ \text{s.t.} \quad & \Psi^\mathsf{T} E^\mathsf{T} \mu = \Psi^\mathsf{T} \left(\gamma P^\mathsf{T} \mu + (1-\gamma) p_0\right). \end{aligned} \tag{5.1}$$

Here, $D\left(\mu \| \mu'\right)$ is the *unnormalized relative entropy* (or Kullback–Leibler divergence) between the distributions μ and μ' defined as

$$D\left(\mu \| \mu'\right) = \sum_{x,a} \big(\mu(x,a)\big(\log \frac{\mu(x,a)}{\mu'(x,a)} - 1\big) + \mu'(x,a)\big).$$

Introducing the notation $V_\theta = \Psi\theta$, the unique optimal solution to this optimization problem can be written as

$$\mu_{k+1}(x,a) = \mu_k(x,a) e^{\eta(r(x,a) + \gamma(PV_\theta)(x,a) - V_\theta(x) - \rho_k)}, \tag{5.2}$$

where ρ_k is a normalization constant and θ_k is given as the minimizer of the *dual function* given as

$$\mathcal{G}_k(\theta) = \sum_{x,a} \mu_{k-1}(x,a) e^{\eta(r(x,a) + \gamma(PV_\theta)(x,a) - V_\theta(x))} + (1-\gamma)\left\langle p_0, V_\theta \right\rangle, \tag{5.3}$$

that is obtained from the primal optimization problem (5.1) through Lagrangian duality.

As highlighted by Zimin and Neu [2013] and Neu et al. [2017], REPS can be seen as a *mirror descent* algorithm [Martinet, 1970; Rockafellar, 1976; Beck and Teboulle, 2003], and thus its iterates μ_k are guaranteed to converge to an optimal occupancy measure μ^*.

Despite its exceptional elegance, the formulation above has a number of features that limit its practical applicability. One very serious limitation of REPS is that its output policy π_K involves an expectation with respect to the transition function, thus requiring knowledge of P to run the policy. Another issue is that optimizing an empirical version of the loss (5.3) as originally proposed by Peters et al. [2010] may be problematic due to the empirical loss being a biased estimator of the true objective (5.3) caused by the conditional expectation appearing in the exponent.

5.3 Q-REPS optimization problem

Inspired by REPS, in this section we derive the Q-REPS optimization problem. In the first part of this section, we derive the primal problem: a constrained convex optimization problem inspired in (5.1) but with some new ideas. After this, Lagrangian duality is used to derive the dual problem, an unconstrained minimization problem whose objective function is the novel logistic Bellman error.

5.3.1 The primal problem

The primal optimization problem derived below build on the following three main ideas:

(a) Lagrangian decomposition of constraints,

(b) linear relaxation of the resulting decomposed constraints, and

(c) proximal entropy regularization.

In what follows, we are going to use these ideas to build our final optimization problem.

Lagrangian decomposition of constraints

As we have seen, one of the main limitations of `REPS` is that its output policy π_K involves an expectation with respect to the transition function P. This is directly related to the fact that the algorithm works with value functions instead of Q functions. This observation motivates a reformulation of the dual LP (3.10) that can be seen to be equivalent to having a primal with Q-functions. To our best knowledge, this LP has been first proposed by Mehta and Meyn [2009] and has been recently rediscovered by Lee and He [2019] and Neu and Pike-Burke [2020] and revisited by Mehta and Meyn [2020]. Specifically, this approach is based on introducing an additional set of primal variables $d \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$ and split the constraints of the LP as follows:

$$\begin{aligned} \text{maximize}_{\mu\in\mathbb{R}^{\mathcal{X}\times\mathcal{A}},d\in\mathbb{R}_+^{\mathcal{X}\times\mathcal{A}}} \quad & \langle \mu, r\rangle \\ \text{s.t.} \quad & E^\mathsf{T} d = \gamma P^\mathsf{T}\mu + (1-\gamma)p_0 \\ & \quad d = \mu. \end{aligned} \tag{5.4}$$

In the problem above, d can be thought of as a "mirror image" of μ. Furthermore, we can see that by making the second constraint implicit (i.e., substituting d by μ), the problem is left only with the variable μ and becomes the same problem as LP(3.10). By straightforward calculations, the dual of this LP can be shown to be

$$\begin{aligned} \text{minimize}_{V\in\mathbb{R}^{\mathcal{X}},Q\in\mathbb{R}^{\mathcal{X}\times\mathcal{A}}} \quad & (1-\gamma)\langle p_0, V\rangle \\ \text{s.t.} \quad & Q = r + \gamma PV \\ & EV \geq Q. \end{aligned} \tag{5.5}$$

The optimal solution of the above LP correspond to the optimal value function and Q-function, V^* and Q^*.

Linear constraint decomposition

With a similar purpose as in REPS, we relax some of the constraints of our optimization problem. Unlike in the original REPS formulation, we relax the set of constraints $d = \mu$ instead of the flow constraints. The motivation of this choice will become clear soon. We introduce a state-action feature map $\varphi : \mathbb{R}^{\mathcal{X}\times\mathcal{A}} \to \mathbb{R}^m$ and the corresponding linear operator Φ^T acting on μ as $\Phi^\mathsf{T}\mu = \sum_{x,a}\mu(x,a)\varphi(x,a)$. We then propose the following relaxed optimization problem:

$$\begin{aligned} \text{maximize}_{\mu\in\mathbb{R}^{\mathcal{X}\times\mathcal{A}},d\in\mathbb{R}_+^{\mathcal{X}\times\mathcal{A}}} \quad & \langle\mu, r\rangle \\ \text{s.t.} \quad & E^\mathsf{T}d = \gamma P^\mathsf{T}\mu + (1-\gamma)p_0 \\ & \Phi^\mathsf{T}d = \Phi^\mathsf{T}\mu. \end{aligned} \tag{5.6}$$

By computing the dual of the above optimization problem, we get

$$\begin{aligned} \text{minimize}_{V\in\mathbb{R}^{\mathcal{X}},\theta\in\mathbb{R}^m} \quad & (1-\gamma)\langle p_0, V\rangle \\ \text{s.t.} \quad & \Phi\theta = r + \gamma PV \\ & EV \geq \Phi\theta. \end{aligned}$$

We can see that the constraint relaxation applied in LP (5.6) through the feature map Φ gives parametrized Q-functions of the form $Q_\theta = \Phi\theta$ in the dual. It is interesting to notice that a relaxation in the flow constraints would turn into parametrized value functions, as it happens in REPS. Nevertheless, as we will see soon, the value functions V will end up being a function of θ too, thanks to the usage of a carefully selected regularization.

Let's now denote as $\mathcal{M}_\Phi$ the set of (μ, d) pairs that satisfy the constraints of the problem (5.7), so that the optimization problem can be rewritten as

$$\text{maximize}_{(\mu,d)\in\mathcal{M}_\Phi} \quad \langle\mu, r\rangle\,.$$

A reasonable question is whether the set $\mathcal{M}_\Phi$ matches the set of valid discounted occupancy measures $\mathcal{M}^* = \{\mu : E^\mathsf{T}\mu = \gamma P^\mathsf{T}\mu + (1-\gamma)p_0\}$ in an appropriate sense. We introduce the following assumption under which we will show an interesting relation between the two sets $\mathcal{M}_\Phi$ and $\mathcal{M}^*$:

Assumption 4 (Factored linear MDP)**.** *There exists a function $\omega : \mathcal{X} \to \mathbb{R}^m$ and a vector $\vartheta \in \mathbb{R}^m$ such that for any x, a, x', the transition function factorizes as $P(x'|x,a) = \langle\omega(x'), \varphi(x,a)\rangle$ and the reward function can be expressed as $r(x,a) = \langle\vartheta, \varphi(x,a)\rangle$.*

The class of factored linear MDPs ensure that the feature space is expressive enough to allow the representation of the optimal action-value function and thus the optimal policy (a property sometimes called *realizability*). This condition has been first proposed by Yang and Wang [2019] and has quickly become a standard model for studying reinforcement learning algorithms under linear function approximation [Jin et al., 2020; Cai et al., 2020; Wang et al., 2020; Neu and Pike-Burke, 2020; Agarwal et al., 2020a]. For this class of MDPs, the following proposition establishes an interesting relation between the sets $\{d : (\mu, d) \in \mathcal{M}_\Phi\}$ and $\mathcal{M}^*$:

Proposition 5.3.1. *Let $\mathcal{M}'_\Phi = \{d : (\mu, d) \in \mathcal{M}_\Phi\}$. Then, under Assumption 4, $\mathcal{M}^* = \mathcal{M}'_\Phi$ holds. Furthermore, letting $(\mu^*, d^*) = \arg\max_{(\mu,d)\in\mathcal{M}_\Phi} \langle \mu, r\rangle$, we have $\langle d^*, r\rangle = \max_{\mu\in\mathcal{M}^*} \langle \mu, r\rangle$.*

Proof. It is easy to see that $\mathcal{M}^* \subseteq \mathcal{M}'_\Phi$: for any $\mu \in \mathcal{M}^*$, we can choose $d = \mu$ and directly verify that all constraints of (5.7) are satisfied. For proving the other direction, it is helpful to define the operator M through its action $Mv = \sum_x \omega(x)v(x)$ for any $v \in \mathbb{R}^{\mathcal{X}}$, so that the condition of Assumption 4 can be expressed as $P = \Phi M$ and $r = \Phi\vartheta$. Then, for any $(\mu, d) \in \mathcal{M}_\Phi$, we write

$$\begin{aligned} E^\mathsf{T} d &= \gamma P^\mathsf{T}\mu + (1-\gamma)p_0 = \gamma M^\mathsf{T}\Phi^\mathsf{T}\mu + (1-\gamma)p_0 \\ &= \gamma M^\mathsf{T}\Phi^\mathsf{T} d + (1-\gamma)p_0 = \gamma P^\mathsf{T} d + (1-\gamma)p_0. \end{aligned}$$

Combined with the fact that d is non-negative, this implies that $d \in \mathcal{M}^*$ and thus that $\mathcal{M}'_\Phi \subseteq \mathcal{M}^*$. Together with the previous argument, this shows that $\mathcal{M}^* = \mathcal{M}'_\Phi$ indeed holds. For proving the second statement, we use the assumption on r to write $\langle \mu, r\rangle = \langle \Phi^\mathsf{T}\mu, \Phi r\rangle = \langle \Phi^\mathsf{T} d, \Phi r\rangle = \langle d, r\rangle$ for any feasible (μ, d). Using this fact for the maximizer d^* implies $\langle d^*, r\rangle = \max_{d\in\mathcal{M}'_\Phi} \langle d, r\rangle = \max_{\mu\in\mathcal{M}^*} \langle \mu, r\rangle$, which concludes the proof. □

This result is of particular interest since, as we will see in the following section, our derivations provide an explicit expression for the policy associated with d^*.

Entropy regularization

Finally, and once again inspired by `REPS`, we introduce a convex regularization term in the objective. We augment the relative-entropy regularization used in `REPS` by a *conditional relative entropy* term defined between two state-action distributions d and d' as

$$H(d\|d') = \sum_{x,a} d(x,a) \log \frac{\pi_d(a|x)}{\pi_{d'}(a|x)}.$$

One minor change is that we will restrict d and μ to belong to the set of probability distributions over $\mathcal{X} \times \mathcal{A}$, denoted as $\mathcal{P}$.

Letting μ_0 and d_0 be two arbitrary reference distributions, denoting the corresponding policy as $\pi_0 = \pi_{d_0}$, and letting α and η be two positive parameters, we define the primal Q-REPS optimization problem as follows:

$$\begin{aligned} \text{maximize}_{\mu,d\in\mathcal{P}} \quad & \langle \mu, r\rangle - \frac{1}{\eta}D(\mu\|\mu_0) - \frac{1}{\alpha}H(d\|d_0) \\ \text{s.t.} \quad & E^{\mathsf{T}}d = \gamma P^{\mathsf{T}}\mu + (1-\gamma)p_0 \\ & \Phi^{\mathsf{T}}d = \Phi^{\mathsf{T}}\mu. \end{aligned} \tag{5.7}$$

Or, equivalently:

$$\text{maximize}_{(\mu,d)\in\mathcal{M}_\Phi} \quad \langle \mu, r\rangle - \frac{1}{\eta}D(\mu\|\mu_0) - \frac{1}{\alpha}H(d\|d_0).$$

In the next section we will see how the incorporation of the conditional relative entropy term $H(d\|d')$ makes it possible to find a closed form solution for the value function V as a function of Q. In consequence, both Q and V will be parameterized by θ.

5.3.2 The dual problem

This section presents the dual problem of the optimization problem that we just proposed. We will denote the optimal solution of problem (5.7) as (μ^+, d^+) and save the notation μ^* and d^* to express other quantities that will appear later. Similarly as in REPS, the following proposition based on Lagrangian duality shows an equivalent problem to (5.7):

Proposition 5.3.2. *Define the Q-function $Q_\theta = \Phi\theta$ taking values $Q_\theta(x,a) = \langle \theta, \varphi(x,a)\rangle$, the value function*

$$V_\theta(x) = \frac{1}{\alpha}\log\left(\sum_a \pi_0(a|x)e^{\alpha Q_\theta(x,a)}\right) \tag{5.8}$$

and the Bellman error function $\Delta_\theta = r + \gamma P V_\theta - Q_\theta$. Then, the optimal solution of the optimization problem (5.7) *is given as*

$$\begin{aligned} \mu^+(x,a) &\propto \mu_0(x,a)e^{\eta\Delta_{\theta^*}(x,a)} \\ \pi_{d^+}(a|x) &= \pi_0(a|x)e^{\alpha\left(Q_{\theta^*}(x,a) - V_{\theta^*}(x)\right)}, \end{aligned}$$

where θ^* is the minimizer of the convex function

$$\mathcal{G}(\theta) = \frac{1}{\eta}\log\left(\sum_{x,a}\mu_0(x,a)e^{\eta\Delta_\theta(x,a)}\right) + (1-\gamma)\langle p_0, V_\theta\rangle. \tag{5.9}$$

Proof. We start writing the Lagrangian of problem (5.7) with $V \in \mathbb{R}^X$, $\theta \in \mathbb{R}^m$ and $\rho \in \mathbb{R}$ as the Lagrange multipliers for the two sets of flow constraints and the normalization constraint of μ respectively

$$\begin{aligned}\mathcal{L}(\mu, d; V, \theta, \rho) &= \langle \mu, r \rangle + \langle V, \gamma P^{\mathsf{T}} \mu + (1-\gamma) p_0 - E^{\mathsf{T}} d \rangle + \langle \theta, \Phi^{\mathsf{T}} d - \Phi^{\mathsf{T}} \mu \rangle \\ &\quad + \rho \left(1 - \langle \mu, \mathbf{1} \rangle\right) - \frac{1}{\eta} D(\mu \| \mu_0) - \frac{1}{\alpha} H(d \| d_0) \\ &= \langle \mu, r + \gamma P V - \Phi\theta - \rho \mathbf{1} \rangle + \langle d, \Phi\theta - EV \rangle + (1-\gamma) \langle p_0, V \rangle \\ &\quad + \rho - \frac{1}{\eta} D(\mu \| \mu_0) - \frac{1}{\alpha} H(d \| d_0) \\ &= \langle \mu, \Delta_{\theta,V} - \rho \mathbf{1} \rangle + \langle d, Q_\theta - EV \rangle + (1-\gamma) \langle p_0, V \rangle + \rho \\ &\quad - \frac{1}{\eta} D(\mu \| \mu_0) - \frac{1}{\alpha} H(d \| d_0), \end{aligned} \tag{5.10}$$

where we used the notation $Q_\theta = \Phi\theta$ taking values $Q_\theta(x,a) = \langle \theta, \varphi(x,a) \rangle$ and $\Delta_{\theta,V} = r + \gamma PV - Q_\theta$ in the last line. Notice that the above is a concave function of d and μ, so its maximum can be found by setting the derivatives with respect to these parameters to zero. In order to do this, we note that

$$\frac{\partial D(\mu \| \mu_0)}{\partial \mu(x,a)} = \log \frac{\mu(x,a)}{\mu_0(x,a)} \quad \text{and} \quad \frac{\partial H(d \| d_0)}{\partial d(x,a)} = \log \frac{\pi_d(a|x)}{\pi_{d_0}(a|x)},$$

where $\pi_d(a|x) = d(x,a) / \sum_{a'} d(x,a')$ and the last expression can be derived by straightforward calculations (see, e.g., Appendix A.4 in Neu et al., 2017). This gives the following expressions for the optimal choices of μ and d:

$$\mu^+(x,a) = \mu_0(x,a) e^{\eta(\Delta_{\theta,V}(x,a) - \rho)} \quad \text{and} \quad \pi_{d^+}(x,a) = \pi_0(x,a) e^{\alpha(Q_\theta(x,a) - V(x))}.$$

From the constraint $\sum_{x,a} \mu^+(x,a) = 1$, we can express the optimal choice of ρ as

$$\rho^* = \log \left(\sum_{x,a} \mu_0(x,a) e^{\eta \Delta_{\theta,V}(x,a)} \right).$$

Similarly, from the constraint $\sum_a \pi_{d^+}(a|x) = 1$, we can express V as a function of θ for all x:

$$V_\theta(x) = \frac{1}{\alpha} \log \left(\sum_a \pi_0(x,a) e^{\alpha Q_\theta(x,a)} \right) \tag{5.11}$$

This implies that d^+ has the form $d^+(x,a) = \nu^+(x) \pi_d^+(a|x)$, where ν^+ is some non-negative function on the state space. Recalling the definition of $\Delta_\theta = r +$

$\gamma PV_\theta - Q_\theta$ and plugging the above parameters $(\mu^+, d^+, \rho^*, V_\theta)$ back into the Lagrangian (5.10) gives the Lagrange dual function

$$\begin{aligned}\mathcal{G}(\theta) =& \mathcal{L}(\mu^+, d^+; V_\theta, \theta, \rho^*)\\ =& \sum_{x,a} \mu_0(x,a) e^{\eta(\Delta_\theta(x,a)-\rho^*)}(\Delta_\theta(x,a) - \rho^*)\\ &+ \sum_{x,a} \nu^+(x)\pi_0(x,a)e^{\alpha(Q_\theta(x,a)-V_\theta(x))}\left(Q_\theta(x,a) - V_\theta(x)\right)\\ &- \sum_{x,a} \frac{1}{\eta}\mu_0(x,a)e^{\eta(\Delta_\theta(x,a)-\rho^*)} \log \frac{\mu_0(x,a)e^{\eta(\Delta_\theta(x,a)-\rho^*)}}{\mu_0(x,a)}\\ &- \sum_{x,a} \frac{1}{\eta}\left(\mu_0(x,a) - \mu_0(x,a)e^{\eta(\Delta_\theta(x,a)-\rho^*)}\right)\\ &- \sum_{x,a} \frac{1}{\alpha}\nu^+(x)\pi_0(x,a)e^{\alpha(Q_\theta(x,a)-V_\theta(x))} \log \frac{\pi_0(x,a)e^{\alpha(Q_\theta(x,a)-V_\theta(x))}}{\pi_0(x,a)}\\ &+ (1-\gamma)\left\langle p_0, V\right\rangle + \rho^*\\ =& (1-\gamma)\left\langle p_0, V\right\rangle + \rho^*\\ =& (1-\gamma)\left\langle p_0, V\right\rangle + \frac{1}{\eta}\log\left(\sum_{x,a}\mu_0(x,a)e^{\eta\Delta_\theta(x,a)}\right).\end{aligned}$$

Furthermore, observe that since the parameters were chosen so that all constraints are satisfied, we also have

$$\mathcal{G}(\theta) = \mathcal{L}(\mu^+, d^+; V_\theta, \theta, \rho^*) = \left\langle \mu^+, r\right\rangle - \frac{1}{\eta}D(\mu^+\|\mu_0) - \frac{1}{\alpha}H(d^+\|d_0)$$

due to strong duality. Thus, the solution of the optimization problem (5.7) can indeed be written as

$$\begin{aligned}\max_{\mu,d\geq 0}\min_{\theta,V,\rho}\mathcal{L}(\mu,d;V,\theta,\rho) &= \min_{\theta,V,\rho}\max_{\mu,d\geq 0}\mathcal{L}(\mu,d;V,\theta,\rho)\\ &= \min_\theta \mathcal{L}(\mu^+, d^+; V_\theta, \theta, \rho^*)\\ &= \min_\theta \mathcal{G}(\theta).\end{aligned}$$

Thus, the constrained optimization problem (5.7) is equivalent to $\min_\theta \mathcal{G}(\theta)$, which concludes the proof. □

This result has several important implications. First, it shows that the constrained optimization problem (5.7) can be reduced to *uncostrained* minimization of the convex loss function $\mathcal{G}$. By analogy with the classic logistic loss, we will call this loss function the *logistic Bellman error*, and its solutions Q_θ and V_θ the *logistic value functions*. It is important to notice that, due to the use of regularization, the logistic value functions have lost the original meaning of value functions explained in Section 3.3. Despite this, they still conserve some of the flavour of their original counterparts as an heuristic of the quality of a given state or state-action pair, as seen in the closed form expression of μ^+ and π_{d^+}.

Another major implication of the above results is that it provides a simple explicit expression for the policy associated with d^+ as a function of the logistic action-value function Q_{θ^*}. This is remarkable since no such policy parameterization is directly imposed in the primal optimization problem (5.7) as a constraint, but it rather emerges naturally from the overall structure we propose.

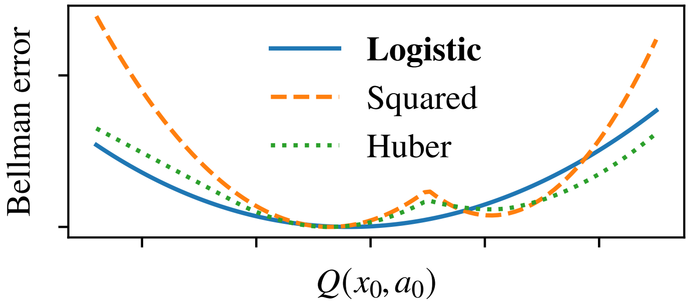


Figure 5.1: Squared Bellman error considered harmful: Loss functions plotted as a function of the Q-value at a fixed state-action pair while keeping other values fixed.

Besides convexity, the LBE has other favorable properties: when regarded as a function of Q, its gradient satisfies $\|\nabla_Q \mathcal{G}(Q)\|_1 \leq 2$ and is thus 2-Lipschitz with respect to the ℓ_∞ norm, and it is smooth with parameter $\alpha+\eta$ (due to being a composition of an α-smooth and an η-smooth function). These additional properties make the LBE a desirable alternative to the squared Bellman error, which is non-convex, non-smooth, and has unbounded gradients. Indeed, the Lipschitzness of the LBE implies that optimizing the loss via stochastic gradient descent does not require any gradient clipping tricks since the derivatives are bounded by default. In this sense, the LBE can be seen as a theoretically well-motivated alternative to the Huber loss commonly used instead of the squared loss for policy evaluation.

The Effect of α on the Action Gap

One interesting feature of the `Q-REPS` optimization problem (5.7) is that it becomes essentially identical to the `REPS` problem (5.1) when setting $\alpha = +\infty$.

To see this, let Ψ and Φ be the identity maps. Then, the primal form of Q-REPS becomes

$$\begin{aligned}\text{maximize}_{\mu,d\in\mathcal{P}} \quad & \langle \mu, r\rangle - \frac{1}{\eta} D(\mu\|\mu_0) \\ \text{s.t.} \quad & E^{\mathsf{T}} d = \gamma P^{\mathsf{T}}\mu + (1-\gamma)p_0 \\ & d = \mu,\end{aligned}$$

which is clearly seen to be a simple reparameterization of the convex program (5.1). Furthermore, when $\alpha = +\infty$, the closed-form expression for V (5.11) is replaced with the inequality constraint $V(x) \geq Q(x,a)$ required to hold for all x, a and the dual function $\mathcal{G}'(Q,V)$ becomes

$$\frac{1}{\eta}\log\left(\sum_{x,a}\mu_0(x,a)e^{\eta\left(r(x,a)+\gamma\sum_{x'}P(x'|x,a)V(x')-Q(x,a)\right)}\right) + (1-\gamma)\langle p_0, V\rangle .$$

Since this function needs to be minimized in terms of Q and V and it is monotone decreasing in Q, its minimum is achieved when the constraints are tight and thus when $Q(x,a) = V(x)$ for all x, a. Thus, in this case Q loses its intuitive interpretation as an action-value function, highlighting the importance of the conditional-entropy regularization in making Q-REPS practical.

From a practical perspective, this suggests that the choice of α impacts the gap between the values of Q: as α goes to infinity, the gap between the values vanish and they become harder to distinguish based on noisy observations. Figure 5.2 shows that the action gap indeed decreases as α is increased, roughly at an asymptotic rate of $1/\alpha$, and that learning indeed becomes harder as the gaps decrease.

5.4 Approximate policy iteration: Q-REPS

In this section we present the algorithmic framework Q-REPS that is based on the minimization of the logistic Bellman error. We will consider a mirror-descent algorithm that calculates a sequence of distributions iteratively as

$$(\mu_k, d_k) = \underset{(\mu,d)\in\mathcal{M}_\Phi}{\arg\max}\, \langle \mu, r\rangle - \frac{1}{\eta}D(\mu\|d_{k-1}) - \frac{1}{\alpha}H(d\|d_{k-1}).$$

for $k = 1, 2, ..., K$. We can realize that we have chosen the reference distribution in the two regularization terms to be d_{k-1}. We have made this choice because of practical implementability, as we will see soon. By the results established in the

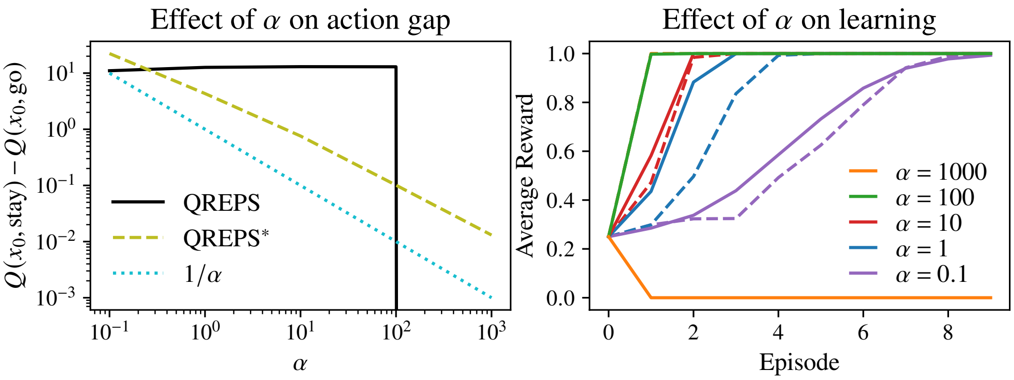


Figure 5.2: Effect of conditional-entropy regularization parameter α on the performance of `Q-REPS`. On this figure, `Q-REPS`* (dashed line) refers to the ideal version of the algorithm that minimizes the exact LBE, whereas `Q-REPS` (solid line) is the sample-based implementation minimizing the empirical LBE. On the left plot, we see the effect of α on the action gap. For `Q-REPS`*, the action gap decreases at a rate slightly slower than $1/\alpha$. On the other hand, for `Q-REPS`, the estimation noise dominates the action gap for smaller values of α. For larger values of α, `Q-REPS` fails to identify the optimal action which results in a negative action gap. On the right plot, we show the performance of the iterative procedure presented in Algorithm 2 for different values of α. For `Q-REPS`*, α plays the role of a learning rate: as α increases so does the learning speed For `Q-REPS`, this effect is only preserved for moderate values of α, as the small action gap in the ideal Q-values makes identifying the optimal action harder. For $\alpha = 100$ (green solid line), the sign is identified correctly and it performs almost as if no regularization was present. For $\alpha = 1000$ (orange solid line), the sign is misidentified and the wrong action is preferred, leading to poor performance.

previous section, implementing these updates requires finding the minimum θ_k^* of the logistic Bellman error function

$$\mathcal{G}_k(\theta) = \frac{1}{\eta} \log \left(\sum_{x,a} d_{k-1}(x,a) e^{\eta \Delta_\theta(x,a)} \right) + (1-\gamma) \langle p_0, V_\theta \rangle \,. \tag{5.12}$$

We will denote the logistic value functions corresponding to θ_k^* as Q_k^* and V_k^*, and the induced policy as $\pi_k^*(a|x)$. In practice, exact minimization can be often infeasible due to the lack of knowledge of the transition function P and limited access to computation and data. Thus, practical implementations of `Q-REPS` will inevitably have to work with approximate minimizers θ_k of the logistic Bellman error $\mathcal{G}_k$. We will denote the corresponding logistic value functions as Q_k and V_k and the policy as π_k, and the distribution d_k will be chosen as the occupancy measure induced by π_k. By analogy with classical approximate policy iteration schemes, we will refer to the minimization of the LBE $\mathcal{G}_k$ as a *policy evaluation* step that is carried out by the subroutine Q-REPS-Eval. Using this language, we present a pseudocode for Q-REPS as Algorithm 2.

Algorithm 2: Q-REPS

```
Initialize $\pi_0$ arbitrarily;
for $k = 1, 2, \ldots, K$ do
    Policy evaluation: $\theta_k = \texttt{Q-REPS-Eval}(\pi_{k-1})$;
    Policy update: $\pi_k(a|x) \propto \pi_{k-1}(a|x) e^{\alpha Q_k(x,a)}$;
end
Result: $\pi_K$
```

Boundedness of the logistic value functions

In the following sections, boundedness of the logistic value functions is required in order to provide an analysis of the performance of the proposed algorithms. Since logistic value functions are not the same as the real value funcitons, we can not use the bounds from Proposition 3.3.4. We do not have any theoretical result proving an upper bound on those quantities, but in the performed experiments (see Section 5.6) we have observed that those values are bounded and behave well. As an example, in Figure 5.3 we show the optimal logistic Q-function as a function of the regularization parameter η (and α is set equal to η) for the different state-action pairs in the two-states environment presented in Figure 5.4. There we can see how the logistic Q-function is bounded for all the range of considered regularization, and it converges to a fixed value for large enough regularization (small values of η). In what follows, we will often use the following assumption regarding the boundedness of the logistic Q-functions:

Assumption 5. *Let $\mathcal{Q} = \{Q_\theta : \|Q_\theta\|_\infty \leq B'\}$ for some $B' > 0$ and Θ be the corresponding set of parameter vectors. Furthermore, define $B = 1 + (1+\gamma)B'$. Then, for all $k = 1, 2..., K$ it holds that $Q_k^* \in \mathcal{Q}$ and $\theta_k^* \in \Theta$.*

5.4.1 Error propagation analysis

In this section we present an analysis of the propagation of optimization errors in the Q-REPS scheme. Specifically, we will study how the suboptimality of each policy evaluation step impacts the performance of the sequence of policies. We first present two general results regarding the performance of the original Q-REPS

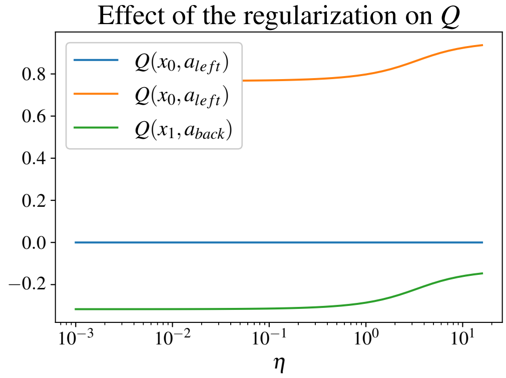


Figure 5.3: Effect of the amount of regularization in the logistic Q-function in the MDP described in Figure 5.4. We set $\alpha = \eta$.

algorithm under almost no assumptions. Both results bound the same quantity but are based on different analysis techniques and use different quantities in the bound. The presented bounds are insightful but not very practical, since there are some residual terms for which we have not been able to show sublinear growth with the number of epochs K. After this, we will make some assumptions and consider a slightly modified algorithm (σ-`Q-REPS`) in order to be able to show a more practical result.

We let $\theta_k^* = \arg\min_\theta \mathcal{G}_k(\theta)$ and define the suboptimality gap associated with the parameter vector θ_k computed by `Q-REPS-Eval` as $\varepsilon_k = \mathcal{G}_k(\theta_k) - \mathcal{G}_k(\theta_k^*)$. We also let $d^* = \arg\max_{d \in \mathcal{M}^*} \langle d, r \rangle$, μ^* be any state-action distribution satisfying $(\mu^*, d^*) \in \mathcal{M}_\Phi$, and $\widetilde{\mu}_k(x,a) = d_{k-1}(x,a) e^{\eta\left(\Delta_{\theta_k}(x,a) - \rho_k\right)}$ for appropriately defined normalization constant ρ_k. Furthermore, we denote the normalized discounted return associated with policy π_k as $R_k = \langle d_k, r \rangle$ and the optimal return as $R^* = \langle d^*, r \rangle$. With these definitions, we can show the following theorem regarding the performance of the sequence of policies output by `Q-REPS`:

Theorem 5.4.1. *Suppose that Assumption 4 hold. Then, the policy sequence output by `Q-REPS` satisfies*

$$\sum_{k=1}^{K} (R^* - R_k) \leq \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(d^* \| d_0)}{\alpha} + \sum_{k=1}^{K} \left(\frac{D(\mu^* \| d_k) - D(\mu^* \| \widetilde{\mu}_k)}{\eta} + \sqrt{\frac{2\alpha\varepsilon_k}{1-\gamma}} + \varepsilon_k \right)$$

As we have seen in the proof of Theorem 2.4.2, the analysis of algorithms that follow a mirror-descent scheme is usually based on a telescopic sum of terms over epochs. In the proof of the above theorem, we can see that the terms do not telescope nicely and there are some divergences left that do not vanish and accumulate over epochs. These terms come from using d_k instead of $\widetilde{\mu}_k$ as the baseline term for the policy evaluation steps, which is the choice that makes practical implementation of the algorithm possible.

The following theorem shows another bound on $\sum_{k=1}^{K}(R^* - R_k)$ based on a different analysis technique. This alternative technique will be useful in the second part of this section for deriving a bound for a modified version of Q-REPS.

Theorem 5.4.2. *Suppose that Assumption 4 hold and without loss of generality, pick μ^* to match d^*: $\mu^* = d^* = \arg\max_{d\in\mathcal{M}^*}\langle d, r\rangle$. Then, the policy sequence output by Q-REPS satisfies:*

$$\begin{aligned}\sum_{k=0}^{K}(R^* - R_k) \leq & \frac{D(\mu^*\|d_0) + H(\mu^*\|d_0)}{\eta} - \sum_{k=0}^{K}\left(\frac{D(\mu_k^*\|d_{k-1})}{\eta} - \frac{H(d_k^*\|d_{k-1})}{\alpha}\right) \\ & + \sum_{k=0}^{K}\left(\frac{D(\mu^*\|d_k) - D(\mu^*\|d_k^*)}{\eta} + \frac{H(\mu^*\|d_k) - H(\mu^*\|d_k^*)}{\alpha}\right) \\ & + \sum_{k=0}^{K}\left(\frac{D(\mu^*\|d_k^*) - D(\mu^*\|\mu_k^*)}{\eta} + \sqrt{\frac{2\alpha\varepsilon_k}{1-\gamma}}\right)\end{aligned}$$

As in Theorem 5.4.1, this bound again features a sum of terms that may not telescope well enough in general. If we look at the terms from the second line of the bound, we can see that those terms become 0 when $d_k = d_k^*$, which happens when $\varepsilon_k = 0$. This suggests that it should be possible to bound those terms in term of ε_k but so far we have not been able to do so without further assumptions. In what follows, we present a modified version of the Q-REPS that together with Assumption 6 (concentrability) makes it possible to bound these terms. Regarding the first term of the last line, it is not clear how to bound it given the relaxation that only ensures $\Phi^\mathsf{T}\mu_k^* = \Phi^\mathsf{T}d_k^*$. Nevertheless, for the tabular case this term vanishes since then there is no constraint relaxation so $\mu_k^* = d_k^*$.

Analysis of σ-**Q-REPS**

In order to derive more meaningful bounds, we now present the analysis of a modified version of the Q-REPS algorithmic template, where some extra exploration is imposed by mixing the policy π_k at each iteration k with the policy π_0 that we set as the uniform policy. In concrete, at the end of each iteration we define

$$\bar{\pi}_{k+1}(a|x) = (1-\sigma)\pi_{k+1}(a|x) + \sigma\pi_0(a|x)$$

with $\sigma \in (0, 1)$, and use this policy in the policy evaluation step of the next iteration. The pseudocode for this new algorithmic scheme that we call σ-`Q-REPS` can be found in Algorithm 3.

Algorithm 3: σ-`Q-REPS`

```
Initialize $\pi_0$;
for $k = 1, 2, \ldots, K$ do
    Policy evaluation:
        $\theta_k = \texttt{Q-REPS-Eval}(\bar{\pi}_{k-1})$;
    Policy update:
        $\pi_k(a|x) \propto \pi_{k-1}(a|x) e^{\alpha Q_k(x,a)}$;
        $\bar{\pi}_k(a|x) = (1 - \sigma)\pi_k(a|x) + \sigma \pi_0(a|x)$;
end
Result: $\pi_K$
```

In addition to the extra exploration, in the analysis we will consider the tabular case, that is, the case where indicator features are used, and we make the following assumption that ensures that every policy explores the state space sufficiently well:

Assumption 6 (Concentrability)**.** *The likelihood ratio for any two valid occupancy measures μ and μ' is upper-bounded by some C_γ called the* concentrability coefficient*:* $\sup_x \frac{\sum_a \mu(x,a)}{\sum_a \mu'(x,a)} \leq C_\gamma$.

Although the above assumption is a rather strong condition that is rarely verified in problems of practical interest, it is commonly assumed to ease theoretical analysis of batch RL algorithms. For instance, similar conditions are required in the classic works of Kakade and Langford [2002], Antos et al. [2006], and more recently by Geist et al. [2017], Agarwal et al. [2020b] and Xie and Jiang [2020].

Before showing the main result for this setting, we want to remark that the modification of the original algorithm to enforce exploration, the restriction to the tabular case, and the concentrability condition have been introduced to make the analysis manageable, but we strongly believe that similar results can be derived for the original algorithm just under Assumption 4 (factored linear MDPs). We leave as future work to derive similar results for `Q-REPS` in a more general setting.

With this in mind, we can present the following theorem regarding the error propagation of σ-`Q-REPS`:

Theorem 5.4.3. *Suppose that Assumptions 5 and 6 hold and that we are in the* *tabular setting. Then, the policy sequence output by σ-*`Q-REPS` *satisfies*

$$
\begin{aligned}
\sum_{k=1}^{K}(R^* - R_k) \leq & \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(\mu^* \| d_0)}{\alpha} \\
& + \sum_{k=1}^{K}\left(1 + \frac{\alpha}{\eta}\right)\left(e^{3\eta B}\sqrt{\frac{2C_\gamma|\mathcal{A}|}{\sigma\eta}} + \sqrt{2\alpha}B\right)\sqrt{\varepsilon_k} \\
& + \sum_{k=1}^{K}\left(1 + \frac{C_\gamma}{\eta}\right)\sqrt{\frac{2\alpha}{1-\gamma}}\sqrt{\varepsilon_k} \\
& + \sum_{k=1}^{K}\left(\left(1 + \frac{C_\gamma}{\eta}\right)\sqrt{\frac{2\sigma}{1-\gamma}\log|\mathcal{A}|} + \frac{\sigma}{\alpha}\log|\mathcal{A}|\right)
\end{aligned}
$$

The proof can be found in Section 5.8. Looking at the terms of the right hand side of the above bound, we can appreciate three different sources of regret. The terms from the first line are the usual terms in mirror-descent algorithms optimizing a fixed linear loss. The terms from the second line correspond to the extra loss coming from the errors in the evaluation steps, ε_k. The terms of the last line are the price to pay to force exploration, since the extra exploration keep us away from the optimal policy. In the next section we present two practical algorithms and use the latter bound to derive more specific performance guarantees.

5.5 Policy evaluation via saddle-point optimization

In this section we provide two versions of an efficient batch reinforcement learning algorithm that implements the `Q-REPS` policy updates through saddle-point optimization. The two versions minimize an empirical and a semi-empirical version of the LBE respectively. For the second one, we show convergence under some conditions by combining the results from the previous section with a concentration bounds regarding the estimate of the LBE.

In this section we will show the derivations for the version of the algorithm with σ-exploration, but the equivalent algorithm without exploration can be trivially built in the same way.

5.5.1 The empirical LBE

In order to use the ideas from the previous section in a reinforcement learning setting, we need to design a policy-evaluation subroutine that is able to directly work with sample transitions obtained through interaction with the environment. We will specifically consider a scheme where at each epoch k, we execute policy $\bar{\pi}_{k-1}$ and obtain a batch of N sample transitions $\{\xi_{k,n}\}_{n=1}^{N}$, with $\xi_{k,n} = (X_{k,n}, A_{k,n}, X'_{k,n})$, drawn from the occupancy measure $\bar{d}_{k-1}$ induced by $\bar{\pi}_{k-1}$. Furthermore, defining the *empirical Bellman error* for any (x, a, x') as

$$\widehat{\Delta}_\theta(x, a, x') = r(x, a) + \gamma V_\theta(x') - Q_\theta(x, a),$$

we define the *empirical logistic Bellman error* (ELBE):

$$\widehat{\mathcal{G}}_k(\theta) = \frac{1}{\eta} \log \left(\frac{1}{N} \sum_{n=1}^{N} e^{\eta \widehat{\Delta}_\theta(\xi_{k,n})} \right) + (1 - \gamma) \langle p_0, V_\theta \rangle . \tag{5.13}$$

As in the case of the REPS objective function (5.3) and the squared Bellman error (3.15), the empirical counterpart of the LBE is a biased estimator of the true loss due to the conditional expectation taken over X' within the exponent. As we show below, this bias can be directly controlled by the magnitude of the regularization parameter η.

Concentration of the empirical LBE

We will now present some important properties of the empirical logistic Bellman error (5.13). For simplicity, we will assume that the sample transitions are generated in an i.i.d. fashion: each $(X_{k,n}, A_{k,n})$ is drawn independently from $\bar{d}_{k-1}$ and $X'_{k,n}$ is drawn independently from $P(\cdot|X_{k,n}, A_{k,n})$. Under this condition, the following theorem establishes the connection between the ELBE and the true LBE:

Theorem 5.5.1. *Suppose that Assumption 5 hold and assume that $\eta B \leq 1$ holds. Then, with probability at least $1 - \delta$, the following holds:*

$$\sup_{\theta \in \Theta} \left| \widehat{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) \right| \leq 8\eta B^2 + 56 \sqrt{\frac{m \log\big((1 + 4BN)/\delta\big)}{N}}.$$

In Section 5.8.5, we provide a more detailed statement of the theorem that holds for general Q-function classes, as well as the proof. The main feature of Theorem 5.5.1 is quantifying the bias of the empirical LBE, showing that it is proportional to the regularization parameter η, making it possible to tune the parameters of `Q-REPS` to reduce it. Regarding the variance of the estimate, it can be controlled with the number of sample transitions N as expected.

The effect of η on the bias of the ELBE

We know from Jensen's inequality that for a given random variable X,

$$e^{\mathbb{E}[X]} \leq \mathbb{E}\left[e^{X}\right],$$

and in general, strict inequality holds. It is our case, where we find a "risk-seeking" effect of the bias in estimating the LBE that favors policies that promise higher extreme values of the return.

We illustrate the effect of this bias in a simple environment below. While Theorem 5.5.1 establishes that the bias is of order η, one may naturally wonder if larger values of η truly result in larger bias, and if the bias impacts the learning procedure negatively. In this section, we show that there indeed exist MDPs where this issue is real.

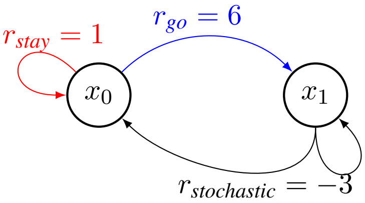


Figure 5.4: Two-state MDP for illustrating the effect of biased estimation of the logistic Bellman error through the empirical LBE. From x_0 there are two actions with deterministic effects: *stay* and *go*. The *stay* action stays in x_0 and results in a reward of $r_{\textit{stay}} = 1$, while the *go* action moves to x_1 and results in a reward of $r_{\textit{go}} = 6$. From x_1 there is one single stochastic action *stochastic* that goes to x_0 or remains in x_1 with equal probability and has reward $r_{\textit{stochastic}} = -3$.

The MDP we consider has two states x_0 and x_1, with two actions available at x_0: *stay* and *go*, with the corresponding rewards being $r_{\textit{stay}}$ and $r_{\textit{go}}$, and the rest of the dynamics are as explained in Figure 5.4. To simplify the reasoning, we set $\gamma = 1$ and consider the case $r_{\textit{stay}} = 0$ first. In this case, the two policies that systematically pick *stay* and *go* respectively would both have zero average reward. Despite this, it can be shown that minimizing the empirical LBE in `Q-REPS` converges to a policy that consistently picks the *go* action for any choice of η. This is due to the risk-seeking effect of the bias explained before, that favors policies that promise higher extreme values of the return. This risk-seeking effect continues to impact the behavior of `Q-REPS` even when $r_{\textit{stay}} = 1$ and η is chosen to be large enough—see the learning curves corresponding to various choices of η in Figure 5.5. This suggests that the bias of the LBE can indeed be a concern in practical implementations of `Q-REPS`, and that the guidance provided by Theorem 5.5.1 is essential for tuning this hyperparameter.

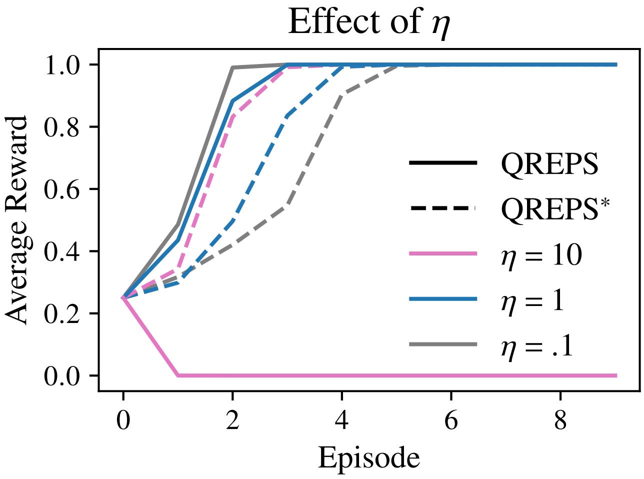


Figure 5.5: Effect of relative entropy regularization parameter η on the performance of Q-REPS. On this figure, Q-REPS* (dashed line) refers to the ideal version of the algorithm that minimizes the exact LBE, whereas Q-REPS (solid line) is the sample-based implementation minimizing the empirical LBE. For large η, Q-REPS suffers from bias and only converges to the optimal policy for smaller values of η. This effect is independent of the sample size N used for the updates. On the other hand, the ideal updates performed by Q-REPS* do not suffer from such bias.

5.5.2 MinMax-Q-REPS

We now provide a practical algorithmic framework for optimizing the empirical LBE (5.13) based on the following reparameterization of the loss function:

Proposition 5.5.1. *Let $\mathcal{D}_N$ be the set of all probability distributions over $[N]$ and define*

$$\widehat{\mathcal{S}}_k(\theta, z) = \sum_n z(n) \left(\widehat{\Delta}_\theta(\xi_{k,n}) - \frac{1}{\eta} \log(N z(n)) \right) + (1-\gamma) \langle p_0, V_\theta \rangle$$

for each $z \in \mathcal{D}_N$. Then, the problem of minimizing the ELBE can be rewritten as $\min_\theta \hat{\mathcal{G}}_k(\theta) = \min_\theta \max_{z \in \mathcal{D}_N} \widehat{\mathcal{S}}_k(\theta, z)$.

The proof is a straightforward aplication of the classic duality formula of the Shannong entropy (see, e.g., Boucheron et al., 2013, Corollary 4.14). In particular, this result establishes that if Z is a real-valued integrable random variable, and p

and q two probability distributions with q absolutely continuous with respect to p, then for every $\eta \in \mathbb{R}$ we have

$$\log \mathbb{E}_p \left[e^{\eta(Z - \mathbb{E}_p[Z])} \right] = \sup_q \left[\eta(\mathbb{E}_q \left[Z\right] - \mathbb{E}_p \left[Z\right]) - D(q \| p) \right].$$

Motivated by the characterization above, we propose to formulate the optimization of the ELBE as a two-player game between a *sampler* and a *learner*: in each round $\tau = 1, 2, \dots, T$, the sampler proposes a distribution $z_{k,\tau} \in \mathcal{D}_N$ over sample transitions and the learner updates the parameters $\theta_{k,\tau}$, together attempting to approximate the saddle point of $\widehat{\mathcal{S}}_k$. In particular, the learner will update the parameters θ through online stochastic gradient descent on the sequence of loss functions $\ell_\tau = \widehat{\mathcal{S}}_k(\cdot, z_{k,\tau})$. In order to estimate the gradients, we define the policy $\pi_{k,\theta}(a|x) = \bar{\pi}_{k-1}(a|x) e^{\alpha(Q_\theta(x,a) - V_\theta(x))}$ and propose the following procedure:

- Sample an index I from the distribution over sample transitions $z_{k,\tau}$ and let $(X, A, X') = (X_{k,I}, A_{k,I}, X'_{k,I})$. Sample an action $A' \sim \pi_{k,\theta}(\cdot|X')$.
- Sample a state $\overline{X} \sim p_0$ and an action $\overline{A} \sim \pi_{k,\theta}(\cdot|\overline{X})$.
- Define the gradient estimate as

$$\widehat{g}_{k,\tau}(\theta) = \gamma \varphi(X', A') - \varphi(X, A) + (1 - \gamma) \varphi(\overline{X}, \overline{A}). \tag{5.14}$$

The following proposition justifies the election of this gradient estimate:

Proposition 5.5.2. *The vector $\widehat{g}_{k,\tau}(\theta)$ is an unbiased estimate of the gradient $\nabla_\theta \widehat{\mathcal{S}}_k(\theta_{k,\tau}, z_{k,\tau})$.*

Proof. For each i, the partial derivatives of $S(\theta, z)$ with respect to θ_i can be written as

$$\begin{aligned} \frac{\partial \widehat{\mathcal{S}}(\theta, z)}{\partial \theta_i} = &\sum_n z(n) \frac{\partial \widehat{\Delta}(X_{k,n}, A_{k,n}, X'_{k,n})}{\partial \theta_i} \\ &+ \sum_{x,y,a} (1 - \gamma) p_0(x) \frac{\partial V_\theta(x)}{\partial Q_\theta(y, a)} \frac{\partial Q_\theta(y, a)}{\partial \theta_i}. \end{aligned} \tag{5.15}$$

Computing the derivatives

$$\frac{\partial V_\theta(x)}{\partial Q_\theta(y, a)} = \mathbb{I}_{\{x=y\}} \frac{\bar{\pi}_{k-1}(a|x) e^{\alpha Q_\theta(x,a)}}{\sum_{a'} \bar{\pi}_{k-1}(a'|x) e^{\alpha Q_\theta(x,a')}} = \mathbb{I}_{\{x=y\}} \pi_{k,\theta}(a|x)$$

and

$$\begin{aligned}\frac{\partial\widehat{\Delta}(X_{k,n},A_{k,n},X'_{k,n})}{\partial\theta_i} =&\gamma\sum_{x,a}\frac{\partial V_\theta(X'_{k,n})}{\partial Q_\theta(x,a)}\frac{\partial Q_\theta(x,a)}{\partial\theta_i}-\frac{\partial Q_\theta(X_{k,n},A_{k,n})}{\partial\theta_i}\\ =&\gamma\sum_a\pi_{k,\theta}(a|X'_{k,n})\varphi_i(X'_{k,n},a)-\varphi_i(X_{k,n},A_{k,n})\end{aligned}$$

and plugging them back in Equation (5.15), we get

$$\begin{aligned}\nabla_\theta\widehat{\mathcal{S}}(\theta,z)=&\sum_{n=1}^N z(n)\left(\gamma\sum_a\pi_{k,\theta}(a|X'_{k,n})\varphi(X'_{k,n},a)-\varphi(X_{k,n},A_{k,n})\right)\\ &+\sum_{x,a}(1-\gamma)p_0(x)\pi_{k,\theta}(a|x)\varphi(x,a).\end{aligned}$$

The statement of the proposition can now be directly verified using the definitions of X, A, X' and $\overline{X}, \overline{A}$. □

Using the gradient estimator $\widehat{g}_{k,\tau}$, the learner updates $\theta_{k,\tau}$ as

$$\theta_{k,\tau+1}=\theta_{k,\tau}-l\widehat{g}_{k,\tau}(\theta_{k,\tau}),$$

where $l>0$ is a stepsize parameter.

As for the sampler, one can consider several different algorithms for updating the distributions $z_{k,\tau}$. A straightforward choice is simply using the best-response strategy playing

$$z_{k,\tau+1}(n)\propto\exp\left(\eta\widehat{\Delta}_{\theta_{k,\tau}}(\xi_{k,n})\right),$$

whence the overall algorithm becomes equivalent to optimizing the empirical LBE via stochastic gradient descent. A slightly more sophisticated (and sometimes empirically more stable) approach is updating the parameters incrementally by first computing the gradient $h_{k,\tau}=\nabla_z\widehat{\mathcal{S}}_k(\theta_{k,\tau},z_{k,\tau})$ with components

$$h_{k,\tau}(n)=\widehat{\Delta}_\theta(\xi_{k,n})-\frac{1}{\eta}\log\left(Nz_{k,\tau}(n)\right),$$

and then updating $z_{k,\tau}$ through an exponentiated gradient step with a stepsize l':

$$z_{k,\tau+1}(n)\propto z_{k,\tau}(n)e^{l'h_{k,\tau}(n)}.$$

We refer to the above procedure as `MinMax-Q-REPS` and provide pseudocode as Algorithm 4. We note that for the case without σ-exploration, the policy update step can be computed as $\pi_k(a|x)\propto\pi_0(a|x)e^{\alpha\sum_{i=0}^k Q_{\theta_i}(x,a)}$, that is very convenient in practical implementations when the Q-function is linearly

Algorithm 4: `MinMax-Q-REPS`

```
Result: $\pi_I$ with $I \sim \text{Unif}(K)$
Initialize $\pi_0$;
for $k = 0, 1, 2, \ldots, K-1$ do
    Run $\bar{\pi}_{k-1}$ and collect sample transitions $\{\xi_{k,n}\}_{n=1}^{N}$;
    Saddle-point optimization for Q-REPS-Eval:
    for $\tau = 1, 2, \ldots, T$ do
        $\theta_{k,\tau} \leftarrow \theta_{k,\tau-1} - l\widehat{g}_{k,\tau-1}(\theta)$;
        $z_{k,\tau}(n) \leftarrow \frac{z_{k,\tau-1}(n)\exp\big(l' h_{k,\tau-1}(n)\big)}{\sum_m z_{k,\tau-1}(m)\exp\big(l' h_{k,\tau-1}(m)\big)}$;
    end
    $\theta_k = \frac{1}{T}\sum_{\tau=0}^{T}\theta_{k,\tau}$;
    Policy update:
        $\pi_k(a|x) \propto \bar{\pi}_{k-1}(a|x)e^{\alpha Q_{\theta_k}(x,a)}$;
        $\bar{\pi}_k(a|x) = (1-\sigma)\pi_k(a|x) + \sigma\pi_0(a|x)$;
end
```

parameterized since then it is only required to store the sum of the parameters from previous epochs.

Interestingly, this saddle-point optimization scheme can be seen as a principled form of *prioritized experience replay* where the samples used for value-function updates are drawn according to some priority criteria [Schaul et al., 2015]. Indeed, our method maintains a probability distribution over sample transitions that governs the value updates, with the distribution being adjusted after each update according to a rule that is determined by the TD error. Different rules for the priority updates result in different learning dynamics with the best choice potentially depending on the problem instance. In our own experiments (see Section 4.5), we have observed that best-response updates tend to be overly aggressive, and the incremental exponentiated gradient updates we describe above lead to more stable behavior. We leave a formal study of these questions as an exciting direction for future work.

Performance analysis

By observing Theorems 5.4.3 and 5.5.1 we can see the following: since the bias of the error is controlled by η, in the term

$$\frac{e^{2\eta B}}{\sqrt{\sigma\beta\eta}}\sqrt{\varepsilon_k}$$

from Theorem 5.4.3, the η from ε_k cancels with the one from the denominator. This makes this term to grow linearly with K no matter how we tune the parameters, which makes impossible to show any convergence guarantee. Despite this, in Section 5.6 we test the algorithm in different scenarios without enforcing exploration and its performance in remarkable. That is why we belief that our negative result is an artifact of our analysis, and it should be possible to show convergence in this setting. As we have seen, this problem is caused by the bias of our estimator. In the next section we show an algorithm thet uses a simulator to draw fresh next states to get rid of the bias, as we saw that can be done for the empirical Bellman error in Section 3.4.

5.5.3 `SimMinMax-Q-REPS`

In this section we show how the bias issue of the ELBE can be eliminated if one has access to a simulator of the environment that allows drawing states from the transition distribution $P(\cdot|x,a)$ for any state-action pair in the replay buffer. Note that this condition is relatively mild since it does not require sampling follow-up states for *any* state-action pair, which may be difficult to provide in practical applications where the set of valid states may not be known a priori.

In concrete, we present a modified version of algorithm `MinMax-Q-REPS` that we call `SimMinMax-Q-REPS`, that uses a simulator to optimize an unbiased estimate of the LBE, the *semi-empirical LBE* (SELBE):

$$\begin{aligned}\widetilde{\mathcal{G}}_k(\theta) =& \frac{1}{\eta}\log\left(\frac{1}{N}\sum_{n=1}^{N} e^{\eta\Delta_\theta(X_{k,n},A_{k,n})}\right) \\ &+ (1-\gamma)\left\langle p_0, V_\theta\right\rangle .\end{aligned} \tag{5.16}$$

To motivate this choice, we can analyze the concentration of the SELBE. As in Theorem 5.5.1, we will assume that the sample transitions are generated in an i.i.d. fashion: each $(X_{k,n}, A_{k,n})$ is drawn independently from d_{k-1}. Under this condition, the following theorem establishes the connection between the SELBE and the true LBE:

Theorem 5.5.2. *Suppose that Assumption 5 hold and assume that $\eta B \le 1$ holds. Then, with probability at least $1-\delta$, the following holds:*

$$\sup_{\theta\in\Theta}\left|\widetilde{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta)\right| \le 56\sqrt{\frac{m\log\big((1+4BN)/\delta\big)}{N}}.$$

The above result shows that the SELBE is a unbiased estimate of the LBE, which justifies its election. The proof of Theorem 5.5.2 can be found in Section 5.8.6.

The following proposition reflects that, like with the ELBE, the problem of minimizing the SELBE is equivalent to solving a saddle-point problem:

Proposition 5.5.3. *Let $\mathcal{D}_N$ be the set of all probability distributions over $[N]$ and define*

$$\begin{aligned}\widetilde{\mathcal{S}}_k(\theta, z) =& \sum_n z(n) \left(\Delta_\theta(X_{k,n}, A_{k,n}) - \frac{1}{\eta} \log(N z(n)) \right) \\ &+ (1-\gamma) \langle p_0, V_\theta \rangle\end{aligned}$$

for each $z \in \mathcal{D}_N$. Then, the problem of minimizing the SELBE can be rewritten as $\min_\theta \widetilde{\mathcal{G}}_k(\theta) = \min_\theta \max_{z \in \mathcal{D}_N} \widetilde{\mathcal{S}}_k(\theta, z)$.

This result motivates a modification of the `MinMax-Q-REPS` algorithm to perform the saddle-point optimization of $\widetilde{\mathcal{S}}_k$ instead of $\widehat{\mathcal{S}}_k$, which will allow us to proof convergence to the minimizer of the semi-empirical LBE. To do so, we only need to take advantage of the simulator and change the way how the gradients and gradient estimates were computed in `MinMax-Q-REPS`.

To compute the estimate of the gradient $\nabla_\theta \widetilde{\mathcal{S}}_k(\theta_{k,\tau}, z_{k,\tau})$ we define the policy $\pi_{k,\theta}(a|x) = \bar{\pi}_{k-1}(a|x) e^{\alpha(Q_\theta(x,a) - V_\theta(x))}$ as before and we propose the following procedure:

- Sample an index I from the distribution $z_{k,\tau}$ and let $(X, A) = (X_{k,I}, A_{k,I})$.
- Sample $X' \sim P(\cdot|X, A)$ (with the simulator) and an action $A' \sim \pi_{k,\theta}(\cdot|X')$.
- Sample a state $\overline{X} \sim p_0$ and an action $\overline{A} \sim \pi_{k,\theta}(\cdot|\overline{X})$.

Then, the gradient estimate w.r.t. θ can be computed as follows:

$$\widetilde{g}_{k,\tau}(\theta) = \gamma \varphi(X', A') - \varphi(X, A) + (1-\gamma) \varphi(\overline{X}, \overline{A}). \tag{5.17}$$

The following proposition justifies the new procedure to estimate the gradient:

Proposition 5.5.4. *The vector $\widetilde{g}_{k,\tau}(\theta)$ is an unbiased estimate of the gradient $\nabla_\theta \widetilde{\mathcal{S}}_k(\theta_{k,\tau}, z_{k,\tau})$.*

Proof. For each i, the partial derivatives of $\widetilde{\mathcal{S}}(\theta, z)$ with respect to θ_i can be written as

$$\begin{aligned}\frac{\partial \widetilde{\mathcal{S}}(\theta, z)}{\partial \theta_i} =& \sum_n z(n) \frac{\partial \Delta(X_{k,n}, A_{k,n})}{\partial \theta_i} \\ &+ \sum_{x,y,a} (1-\gamma) p_0(x) \frac{\partial V_\theta(x)}{\partial Q_\theta(y, a)} \frac{\partial Q_\theta(y, a)}{\partial \theta_i}.\end{aligned} \tag{5.18}$$

Recalling that

$$\Delta(x,a) = r + \gamma P(x'|x,a)V(x') - Q(x,a),$$

we start computing the derivatives

$$\frac{\partial V_\theta(x)}{\partial Q_\theta(y,a)} = \mathbb{I}_{\{x=y\}} \frac{\bar{\pi}_{k-1}(a|x)e^{\alpha Q_\theta(x,a)}}{\sum_{a'} \bar{\pi}_{k-1}(a'|x)e^{\alpha Q_\theta(x,a')}} = \mathbb{I}_{\{x=y\}}\pi_{k,\theta}(a|x)$$

and

$$\begin{aligned}\frac{\partial \Delta(X_{k,n},A_{k,n})}{\partial \theta_i} =& \gamma \sum_{x'} P(x'|X_{k,n},A_{k,n}) \sum_{x,a'} \frac{\partial V_\theta(x')}{\partial Q_\theta(x,a')} \frac{\partial Q_\theta(x,a')}{\partial \theta_i} \\ &- \frac{\partial Q_\theta(X_{k,n},A_{k,n})}{\partial \theta_i} \\ =& \gamma \sum_{x'} P(x'|X_{k,n},A_{k,n}) \sum_{a'} \pi_{k,\theta}(a'|x')\varphi_i(x',a') \\ &- \varphi_i(X_{k,n},A_{k,n}),\end{aligned}$$

and plugging them back in Equation (5.18), we get

$$\begin{aligned}\nabla_\theta \widetilde{\mathcal{S}}(\theta,z) =& \sum_{n=1}^{N} z(n) \left(\gamma \sum_{x'} P(x'|X_{k,n},A_{k,n}) \sum_{a'} \pi_{k,\theta}(a'|x')\varphi(x',a') \right) \\ &- \sum_{n=1}^{N} z(n)\varphi(X_{k,n},A_{k,n}) \\ &+ \sum_{x,a} (1-\gamma)p_0(x)\pi_{k,\theta}(a|x)\varphi(x,a).\end{aligned}$$

The statement of the proposition can now be directly verified using the definitions of $X, A, X', \overline{X}$ and $\overline{A}$. □

As in the previous algorithm, we can consider different procedures to update the weights of the sampler. One option is to note that

$$\left(\nabla_z \widetilde{\mathcal{S}}_k(\theta_{k,\tau}, z_{k,\tau})\right)(n) = \Delta_\theta(\xi_{k,n}) - \frac{1}{\eta}\log\left(N z_{k,\tau}(n)\right),$$

so an unbiased estimate of this gradient can be easily computed by sampling an $X'_{k,n} \sim P(\cdot|X_{k,n},A_{k,n})$ for each n and defining the gradient estimate $\widetilde{h}_{k,\tau}$ with components

$$\widetilde{h}_{k,\tau}(n) = r(X_{k,n},A_{k,n}) + \gamma V_k(X'_{K,n}) - Q_k(X_{k,n},A_{k,n}) - \frac{1}{\eta}\log\left(N z_{k,\tau}(n)\right).$$

An alternative choice for the gradient estimate to minimize the number of times that we call the simulator would be to sample an index I uniformly from $[N]$ and an $X'_{k,I} \sim P(\cdot|X_{k,I}, A_{k,I})$ and let the gradient estimate $\widetilde{h}_{k,\tau}$ to have components

$$\begin{aligned}\widetilde{h}_{k,\tau}(n) =& \mathbb{I}_{\{n=I\}} \left(r(X_{k,I}, A_{k,I}) + \gamma V_k(X'_{k,I}) - Q_k(X_{k,I}, A_{k,I})\right) \\ &- \mathbb{I}_{\{n=I\}} \left(\frac{1}{\eta} \log\left(N z_{k,\tau}(I)\right)\right).\end{aligned}$$

Once we have computed the gradient estimate with one of these methods, $z_{k,\tau}$ can be updated through an exponentiated gradient step with stepsize l':

$$z_{k,\tau+1}(n) \propto z_{k,\tau}(n) e^{l' \widetilde{h}_{k,\tau}(n)}.$$

As in `MinMax-Q-REPS`, another option to update the weights of the sampler is to use best-response:

$$z_{k,\tau+1}(n) \propto \exp\left(\eta \widetilde{\Delta}_{\theta_{k,\tau}}(\xi_{k,n})\right),$$

where $\widetilde{\Delta}$ would be an unbiased estimate of Δ that can be trivially computed by using the same ideas as for $\widetilde{h}_{k,\tau}$.

Performance analysis

Let's consider any algorithm following the σ-`Q-REPS` scheme and minimizing the semi-empirical LBE (5.16) at each epoch (e.g., `SimMinMax-Q-REPS`). Then, putting together Theorems 5.4.3 and 5.5.2 we can derive the following performance guarantee :

Corollary 5.5.1. *Suppose that we are in the tabular case (so $m = |\mathcal{X} \times \mathcal{A}|$), that Assumptions 6 and 5 hold, and that at each update σ-`Q-REPS` is implemented by minimizing the semi-empirical LBE evaluated on N independent sample transitions. Furthermore, suppose that $\eta B \leq 1$ and let*

$$E = \left(56\sqrt{|\mathcal{X} \times \mathcal{A}| \log \frac{1+4BN}{\delta}}\right)^{\frac{1}{2}}, \qquad R = \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(\mu^* \| d_0)}{\alpha},$$

$$A = 2e^{3\eta B}\sqrt{\frac{2C_\gamma |\mathcal{A}|}{\eta}}, \qquad C = \left(1 + \frac{C_\gamma}{\eta}\right)\sqrt{\frac{2}{1-\gamma}\log|\mathcal{A}|}, \qquad \textit{and}$$

$$\begin{aligned}S =& \sqrt{ACE} \\ =& 2^{\frac{5}{2}} e^{\frac{3}{2}\eta B} \left(1 + \frac{C_\gamma}{\eta}\right)^{\frac{1}{2}} \left(\frac{C_\gamma |\mathcal{A}| \log|\mathcal{A}|}{(1-\gamma)\eta}\right)^{\frac{1}{4}} \left(|\mathcal{X} \times \mathcal{A}| \log \frac{(1+4BN)}{\delta}\right)^{\frac{1}{8}}.\end{aligned}$$

Then, setting

$$\sigma = \frac{AE}{C} N^{-\frac{1}{4}}, \quad \textit{and} \quad N = \left(\frac{S}{R}\right)^8 K^8,$$

and letting the value of $\eta = \alpha$ *fixed, we have that after observing* $T = KN$ *transitions our algorithm is guaranteed to output an* ϵ*-optimal policy with*

$$\epsilon = \widetilde{O}\left(C_\gamma^{\frac{20}{27}} \left(\frac{|\mathcal{A}| \log |\mathcal{A}|}{1-\gamma}\right)^{\frac{8}{27}} \left(|\mathcal{X} \times \mathcal{A}| \log \frac{1}{\delta} \frac{D(\mu^* \| d_0) + H(\mu^* \| d_0)}{\eta^8} \right)^{\frac{1}{9}} T^{-\frac{1}{9}} \right),$$

with probability at least $(1 - K\delta)$.

Recall that in the above corollary we do not have a closed form solution for N since there is a $log(N)$ term inside S. Nevertheless, the equation for N is guaranteed to have a solution.

The theorem shows that under appropriate conditions, the policies output by `SimMinMax-Q-REPS` do converge to the optimal policy π^*. In concrete, we can appreciate that after observing T transitions, `SimMinMax-Q-REPS` outputs a ϵ-optimal policy with ϵ of the order of $T^{-\frac{1}{9}}$.

5.6 Experiments

In this section we evaluate `Q-REPS` empirically. For the implementation we consider the original `Q-REPS` algorithmic scheme (Algorithm 2) where the policies π_k are *not* mixed with π_0 at each iteration to enforce exploration. This decision has been made because we consider that the extra exploration introduced in σ-`Q-REPS` (Algorithm 3) is an artifact for the analysis, and it has been shown not to be necessary in practice. For the exact implementation, we use the `MinMax-Q-REPS` scheme (version without simulator) with two different update rules (exponentiated gradient and best response) for the sampler depending on the environment. The exact details are explained below in the *hyperparameters* paragraph. We run the experiments without simulator because we have not found problems controlling the bias.

As the algorithm is essentially on-policy, we compare it with:

- DQN using Polyak averaging and getting new samples at every episode [Mnih et al., 2015],
- PPO as a surrogate of TRPO [Schulman et al., 2017],
- VMPO as the on-policy version of MPO [Song et al., 2019],
- and REPS with parametric policies [Deisenroth et al., 2013].

We evaluate these algorithms in different standard environments. The used environments and all the relevant information about the algorithm specification and training are the following:

Environment description. We use Double-chain and Single-Chain from Furmston and Barber [2010], River Swim from Strehl and Littman [2008], WideTree from Ayoub et al. [2020], CartPole from Brockman et al. [2016], Two-State Deterministic from Bagnell and Schneider [2003], windy-grid world from Sutton and Barto [2018], and a new Two-State Stochastic that we present in Figure 5.4.

Code environment. We use the open-source implementation of these algorithms from Curi [2020] which is based on PyTorch [Paszke et al., 2017].

Features For all environments we use indicator features, except for Cart-Pole. For the later we initialize a two-layer neural network with a hidden layer of 200 units and ReLU activations, and use the default initialization from PyTorch. We freeze the first layer and use the outputs of the activations as state features $\phi' : \mathcal{X} \to \mathbb{R}^{200}$. To account for early termination, we multiply each of the features with an indicator feature $\delta(x)$ that takes the value 1 if the transition is valid and 0 if the next transition terminates. The final state features are given by the product $\phi(x) = \phi'(x)\delta(x) \in \mathbb{R}^{200}_{\geq 0}$. Finally, we define state-action features $\varphi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{200\times 2}$ by letting $\varphi_{i,b}(x, a) = \phi_i(x)\mathbb{I}_{\{a=b\}}$ for all i and both actions $b \in \mathcal{A}$.

Training For all environments but CartPole we run episodes of length 200 and update the policy at the end of each episode. Due to the early-termination of CartPole, we run episodes until termination or length 200 and update the policy after 4 episodes.

Hyperparameters. In Table 5.1 we show the hyperparameters we use for each environment. We fix the regularization parameters as $\eta = \alpha$ and set them so that $1/\eta$ matches the average optimal returns in each game. As optimizers for the player controlling the θ parameters in `MinMax-Q-REPS` (the learner), we use SGD [Robbins and Monro, 1951] and in CartPole we use Adam [Kingma and Ba, 2014]. For the player controlling the distributions z (the sampler), we use the exponentiated gradient (EG) update explained in the main text as the default choice, and use the best response (BR) for CartPole:

$$z_{k,\tau+1}(n) \propto e^{\eta \widehat{\Delta}_{k,\tau}(\xi_{k,n})}.$$

The learning rates l and l' were picked as the largest values that resulted in stable optimization performance.

	η, α	l	l'	γ	T	Learner	Sampler	Features
Default	.5	.1	.1	1.0	200	SGD	EG	Tabular
Cart Pole	.01	.08	x	.99	-	Adam	BR	Linear
Double Chain	-	.01	-	-	-	-	-	-
River Swim	2.5	.01	-	-	-	-	-	-
Single Chain	5.0	.05	-	-	-	-	-	-
Two State D	-	.05	-	-	-	-	-	-
Two State S	-	-	-	-	-	-	-	-
Wide Tree	-	-	.05	-	-	-	-	-
Grid World	-	-	.03	-	-	-	-	-

Table 5.1: Experiment hyperparameters. The "-" symbol indicates that the default values were used, whereas "x" symbol indicates that the algorithm does not require such hyperparameter.

Once the experimental setup is clear, we show the obtained results. In Fig. 5.6 we plot the sample mean and one standard deviation of 50 independent runs of the algorithms (random seeds 0 to 49). In all cases, `Q-REPS` outperforms or is comparable to the competing algorithms. It is remarkable the case of Cart Pole, that is the most challenging environment and we clearly see a much faster convergence of `Q-REPS` in contrast to the other algorithms. Nevertheless, all the used environments are relatively simple and we have not done an exhaustive comparison so we can not generalize these results.

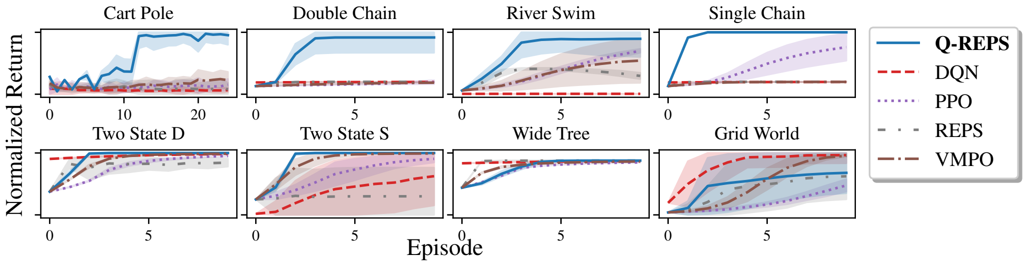


Figure 5.6: Empirical performance `Q-REPS` on different benchmarks. The returns are scaled to $[0, 1]$ by dividing by the maximum achievable return, with the mean plotted in solid lines and and the shaded area representing one standard deviation.

5.7 Conclusions

Due to its many favorable properties, we believe that `Q-REPS` has significant potential to become a state-of-the-art method for reinforcement learning. That said, there is still a lot of room for improvement on both fronts of theoretical guarantees and practical applicability. We outline some challenges for future research and discuss some implications of our results below.

Limitations of our theory. While our theoretical guarantees have several desirable properties, they also have a number of shortcomings. First, most of our analysis requires the condition that the logistic Q-functions have to be bounded. While we were not able to prove an explicit upper bound on the logistic Q-functions, our extensive supplementary experiments indicate that they are bounded by a constant independent of η (see Section 5.4), and we believe that a more sophisticated analysis could formally establish this property. Second, Theorem 5.4.3 requires enforcing extra exploration, restricting the analysis to the tabular case, and the concentrability assumption to hold. Out of this specific scenario, we have not been able to bound the divergence terms that appear in Theorem 5.4.1. Nevertheless, we believe that these shortcomings are an artifact of our analysis and expect that they can be removed by a more careful proof technique. In light of these limitations, we prefer to think of the guarantees of Theorems 5.4.3, 5.5.1 and 5.5.2 as promising initial results, and we leave the important challenge of tightening these guarantees open for future work.

Limitations of our algorithm. The most important merit of `Q-REPS` is that it can be implemented without any significant deviation from its theoretical specifications. The most serious implementation issue is that `Q-REPS` requires sampling from the discounted occupancy measure, which can only be done efficiently when having access to a reset action. This is a common issue of many reinforcement learning algorithms that is often addressed by using samples from the undiscounted state-action distribution. This heuristic often leads to well-performing practical algorithms, but has been long known to suffer from bias issues, as pointed out by Thomas [2014] and Nota and Thomas [2019]. We expect that this heuristic could help practical implementations of `Q-REPS`, although it should be applied with caution. Another practical limitation of our algorithm (without σ-exploration) is that it requires storing the cumulative sum of all past logistic Q-functions, which is not feasible without approximations in a deep RL implementation. It is straightforward to address this limitation by adjusting the regularization terms, but it is currently unclear if it is still possible to meaningfully control the error propagation of the resulting variant.

The relaxed LP formulation. Our method is based on a subtle variation on the classic LP formulation of optimal control in MDPs due to Manne [1960]. One key element in our formulation is a linear relaxation of some of the constraints in this LP, which is a technique looking back to a long history: a similar relaxation has been first proposed by Schweitzer and Seidmann [1985], whose approach was later popularized by the influential work of de Farias and Van Roy [2003]. This latter paper initiated a long line of work studying the properties of solutions to various linearly relaxed versions of the LP, mostly focusing on the quality of value functions extracted from the solutions (see, e.g., Petrik and Zilberstein, 2009; Desai et al., 2012; Lakshminarayanan et al., 2017). Another complementary line of work was initiated by Peters et al. [2010], whose main goal was deriving practical RL algorithms from a relaxed LP formulation. Our own work is heavily influenced by this latter line of research, in that our main focus is also on algorithmic aspects. That said, one important result in our paper is providing a sufficient condition for the LP relaxation to yield exact solutions to the original LP: our analysis shows that for factored linear MDPs, the relaxation we propose suffers from no approximation error (cf. Proposition 5.3.1). Understanding the approximation errors without this structural assumption is a very exciting question that we plan to address in future work, building on the approximate linear programming literature initiated by de Farias and Van Roy [2003]. Similarly, we expect that our algorithmic techniques can be combined with other, more sophisticated relaxation methods. In light of this discussion, we view our work as a promising step toward bridging the gap between LP-based approximate dynamic-programming approaches and mainstream reinforcement learning.

5.8 Omitted proofs

This section collects the proofs of Theorems seen during the chapter that due to their lengths have been allocated here.

5.8.1 Some useful tools

Here we present some results that will be used in the subsequent sections for proving Theorems 5.4.1, 5.4.2 and 5.4.3. We first introduce some useful notation and outline the main challenges faced in the analysis. We start by defining the action-value functions $Q_k = \Phi\theta_k$ and $Q_k^* = \Phi\theta_k^*$, the policies

$$\pi_k(a|x) = \pi_{k-1}(a|x)e^{\alpha\left(Q_{\theta_k}(x,a)-V_{\theta_k}(x)\right)}$$

and

$$\pi_k^*(a|x) = \pi_{k-1}(a|x)e^{\alpha\left(Q_{\theta_k^*}(x,a)-V_{\theta_k^*}(x)\right)},$$

and the state-action distributions

$$\widetilde{\mu}_k(x,a) = d_{k-1}(x,a) e^{\eta\left(\Delta_{\theta_k}(x,a) - \rho_k\right)}$$

and

$$\mu_k^*(x,a) = d_{k-1}(x,a) e^{\eta\left(\Delta_{\theta_k^*}(x,a) - \rho_k^*\right)},$$

for appropriately defined normalization constants ρ_k and ρ_k^* and where d_k is the state-action distribution induced by policy π_k.

A crucial challenge we have to address in the analysis is that, since θ_k is not the exact minimizer of $\mathcal{G}_k$, the state-action distribution $\widetilde{\mu}_k$ is not a valid occupancy measure. In order to prove meaningful guarantees about the performance of the algorithm, we need to consider the actual occupancy measure d_k induced by policy π_k. We define it for all x, a as

$$d_k(x,a) = (1-\gamma)\mathbb{E}_{\pi_k}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}_{\{(x_t,a_t)=(x,a)\}}\right],$$

where the notation emphasizes that the actions are generated by policy π_k. During the proof, we will often factorize occupancy measures as $d(x,a) = \nu(x)\pi(a|x)$, where ν is the discounted state-occupancy measure induced by π. In particular, we will use the notations

$$d_k(x,a) = \nu_k(x)\pi_k(a|x) \qquad \textit{and} \qquad d_k^*(x,a) = \nu_k^*(x)\pi_k^*(a|x),$$

to refer to the state-action occupancy measures respectively induced by π_k and π_k^*.

Our first lemma presents an important technical result that relates the suboptimality gap ε_k to the divergence between the ideal and realized updates.

Lemma 5.8.1. $\varepsilon_k = \frac{D(\mu_k^* \| \widetilde{\mu}_k)}{\eta} + \frac{H(d_k^* \| d_k)}{\alpha}$.

Notably, this result does not require any assumption, as its proof only uses the properties of the optimization problem (5.7).

Proof. The proof uses the feasibility of (μ_k^*, d_k^*) that follows from their definition.

We start by observing that

$$
\begin{aligned}
D(\mu_k^*\|\widetilde{\mu}_k) =& \sum_{x,a} \mu_k^*(x,a) \log \frac{\mu_k^*(x,a)}{\widetilde{\mu}_k(x,a)} \\
=& \eta \big\langle \mu_k^*, r + \gamma P V_k^* - Q_k^* - \rho_k^* \mathbf{1} - r - \gamma P V_k + Q_k + \rho_k \mathbf{1} \big\rangle \\
=& \eta \big\langle d_k^*, E V_k^* - E V_k \big\rangle + \eta \left\langle \Phi^\mathsf{T} \mu_k^*, \theta_k - \theta_k^* \right\rangle \\
& + \eta (\rho_k + (1-\gamma) \left\langle p_0, V_k \right\rangle - \rho_k^* - (1-\gamma) \left\langle p_0, V_k^* \right\rangle) \\
& \qquad \text{(using } d_k^* = \gamma P^\mathsf{T} \mu_k^* + (1-\gamma) p_0 \text{ and } Q_k - Q_k^* = \Phi(\theta_k - \theta_k^*)) \\
=& \eta \big\langle d_k^*, E V_k^* - E V_k \big\rangle + \eta \left\langle \Phi^\mathsf{T} d_k^*, \theta_k - \theta_k^* \right\rangle + \eta (\mathcal{G}_k(\theta_k) - \mathcal{G}_k(\theta_k^*)) \\
& \qquad \text{(using } \Phi^\mathsf{T} d_k^* = \Phi^\mathsf{T} \mu_k^* \text{ and the form of } \mathcal{G}_k) \\
=& \eta \big\langle d_k^*, E V_k^* - Q_k^* - E V_k + Q_k \big\rangle + \eta (\mathcal{G}_k(\theta_k) - \mathcal{G}_k(\theta_k^*)).
\end{aligned}
$$

On the other hand, we have

$$
H(d_k^*\|d_k) = \sum_{x,a} d_k^*(x,a) \log \frac{\pi_k^*(a|x)}{\pi_k(a|x)} = \alpha \left\langle d_k^*, Q_k^* - E V_k^* - Q_k + E V_k \right\rangle .
$$

Putting the two equalities together, we get

$$
\frac{D(\mu_k^*\|\widetilde{\mu}_k)}{\eta} + \frac{H(d_k^*\|d_k)}{\alpha} = \mathcal{G}_k(V_k) - \mathcal{G}_k(V_k^*)
$$

as required. □

The next result shows that, as a consequence of the above property, the realized occupancy measure d_k will be close to the ideal one, d_k^*. The proof only uses Assumption 4 to make sure that d_k^* is a valid occupancy measure.

Lemma 5.8.2. *For any two valid occupancy measures* d *and* d'*, we have that*

$$
D\left(d\|d'\right) \leq \frac{H\left(d\|d'\right)}{1-\gamma}.
$$

In particular, if Assumption 4 holds then $D\left(d_k^*\|d_k\right) \leq \frac{H\left(d_k^*\|d_k\right)}{1-\gamma}$.

Proof. The proof follows from direct calculations and exploiting several properties of the relative entropy. We proof it for d_k^* and d_k but the reader can see that the same proof works for any two valid occupancy measures.

$$\begin{aligned}
D\left(d_k^*\|d_k\right) &= D\left(\nu_k^*\|\nu_k\right) + H\left(d_k^*\|d_k\right) \\
&\qquad \text{(by the chain rule of the relative entropy)} \\
&= D\left((1-\gamma)p_0 + \gamma P^\mathsf{T} d_k^* \| (1-\gamma)p_0 + \gamma P^\mathsf{T}\mu_k\right) + H\left(d_k^*\|d_k\right) \\
&\qquad \text{(using that } d_k^* \text{ and } d_k \text{ are valid occupancy measures)} \\
&\leq (1-\gamma) D\left(p_0\|p_0\right) + \gamma D\left(P^\mathsf{T} d_k^* \| P^\mathsf{T} d_k\right) + H\left(d_k^*\|d_k\right) \\
&\qquad \text{(using the joint convexity of the relative entropy)} \\
&\leq \gamma D\left(d_k^*\|d_k\right) + H\left(d_k^*\|d_k\right),
\end{aligned}$$

where the final step follows from the using information-processing inequality for the relative entropy. Reordering the terms concludes the proof. □

Armed with the above definitions and lemmas we are ready to proof Theorems 5.4.2, 5.4.1 and 5.4.3.

5.8.2 The proof of Theorem 5.4.1

The proof is based on direct calculations inspired by the classical mirror descent analysis. We first express the divergence between the comparator μ^* and the unprojected iterate $\widetilde{\mu}_k$:

$$\begin{aligned}
D(\mu^*\|\widetilde{\mu}_k) =& \sum_{x,a} \mu^*(x,a) \log \frac{\mu^*(x,a)}{\widetilde{\mu}_k(x,a)} \\
=& \sum_{x,a} \mu^*(x,a) \log \frac{\mu^*(x,a)}{d_{k-1}(x,a)} - \sum_{x,a} \mu^*(x,a) \log \frac{\widetilde{\mu}_k(x,a)}{d_{k-1}(x,a)} \\
=& D(\mu^*\|d_{k-1}) - \eta \left\langle \mu^*, r + \gamma P V_k - Q_k \right\rangle + \eta \rho_k \\
=& D(\mu^*\|d_{k-1}) - \eta \left\langle \mu^*, r - \Phi\theta_k \right\rangle - \eta \left\langle d^*, E V_k \right\rangle \\
& + \eta\big(\rho_k + (1-\gamma)\left\langle p_0, V_k \right\rangle\big)
\end{aligned}$$

Where in the last equality we used that $d^* = \gamma P^\mathsf{T}\mu^* + (1-\gamma)p_0$ and $Q_k = \Phi\theta_k$). Now, by using that

$$\rho_k + (1-\gamma)\left\langle p_0, V_k \right\rangle = \mathcal{G}_k(\theta_k) \leq \mathcal{G}_k(\theta_k^*) + \varepsilon_k,$$

we can write

$$
\begin{aligned}
D(\mu^*\|\widetilde{\mu}_k) \leq & D(\mu^*\|d_{k-1}) - \eta\left\langle\mu^*, r\right\rangle + \eta\left\langle d^*, \Phi\theta_k - EV_k\right\rangle + \eta\mathcal{G}_k(\theta_k^*) + \eta\varepsilon_k \\
& \qquad \text{(using the suboptimality guarantee of } \theta_k\text{)} \\
\leq & D(\mu^*\|d_{k-1}) - \eta\left\langle\mu^*, r\right\rangle + \eta\left\langle d^*, \Phi\theta_k - EV_k\right\rangle + \eta\left\langle\mu_k^*, r\right\rangle \\
& - D(\mu_k^*\|d_{k-1}) - \frac{\eta H(d_k^*\|d_{k-1})}{\alpha} + \eta\varepsilon_k \\
& \qquad \text{(using the dual form (5.3.2) of } \mathcal{G}_k(\theta_k)\text{)} \\
\leq & D(\mu^*\|d_{k-1}) - \eta\left\langle\mu^*, r\right\rangle + \eta\left\langle d^*, \Phi\theta_k - EV_k\right\rangle + \eta\left\langle d_k, r\right\rangle \\
& + \eta\left\langle d_k^* - d_k, r\right\rangle + \eta\varepsilon_k \\
& \qquad \text{(using that } \left\langle d_k^*, r\right\rangle = \left\langle\mu_k^*, r\right\rangle \text{ by Proposition 5.3.1)} \\
\leq & D(\mu^*\|d_{k-1}) - \eta\left\langle\mu^*, r\right\rangle + \eta\left\langle d^*, \Phi\theta_k - EV_k\right\rangle + \eta\left\langle d_k, r\right\rangle \\
& + \eta\left\|d_k^* - d_k\right\|_1 + \eta\varepsilon_k,
\end{aligned}
$$

where we used $\|r\|_\infty \leq 1$ in the last step. After reordering and noticing that $\left\langle\mu^*, r\right\rangle = \left\langle d^*, r\right\rangle$, we obtain

$$
\left\langle d^* - d_k, r\right\rangle \leq \frac{D(\mu^*\|d_{k-1}) - D(\mu^*\|\widetilde{\mu}_k)}{\eta} + \left\langle d^*, Q_k - EV_k\right\rangle + \eta\left\|d_k^* - d_k\right\|_1 + \varepsilon_k.
$$

Furthermore, we have

$$
\begin{aligned}
H(d^*\|d_k) &= \sum_{x,a} d^*(x,a)\log\frac{\pi^*(a|x)}{\pi_k(a|x)} \\
&= \sum_{x,a} d^*(x,a)\log\frac{\pi^*(a|x)}{\pi_{k-1}(a|x)} - \sum_{x,a}\mu(x,a)\log\frac{\pi_k(a|x)}{\pi_{k-1}(a|x)} \\
&= H(d^*\|d_{k-1}) - \alpha\left\langle d^*, Q_k - EV_k\right\rangle.
\end{aligned}
$$

Plugging this equality back into the previous bound, we finally obtain

$$
\begin{aligned}
\left\langle d^* - d_k, r\right\rangle \leq & \frac{D(\mu^*\|d_{k-1}) - D(\mu^*\|\widetilde{\mu}_k)}{\eta} + \frac{H(d^*\|d_{k-1}) - H(d^*\|d_k)}{\alpha} \\
& + \left\|d_k^* - d_k\right\|_1 + \varepsilon_k \\
= & \frac{D(\mu^*\|d_k) - D(\mu^*\|\widetilde{\mu}_k)}{\eta} + \frac{D(\mu^*\|d_{k-1}) - D(\mu^*\|d_k)}{\eta} \\
& + \frac{H(d^*\|d_{k-1}) - H(d^*\|d_k)}{\alpha} + \left\|d_k^* - d_k\right\|_1 + \varepsilon_k.
\end{aligned}
$$

Summing up for all k and omitting some non-positive terms, we obtain

$$\begin{aligned}\sum_{k=1}^{K} \langle d^* - d_k, r \rangle \leq & \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(d^* \| d_0)}{\alpha} \\ & + \sum_{k=1}^{K} \left(\frac{D(\mu^* \| d_k) - D(\mu^* \| \widetilde{\mu}_k)}{\eta} + \left\| d_k^* - d_k \right\|_1 + \varepsilon_k \right)\end{aligned} \tag{5.19}$$

Combining Lemma 5.8.2 with Pinsker's inequality, we can bound

$$\left\| d_k^* - d_k \right\|_1 \leq \sqrt{2D(d_k^* \| d_k)} \leq \sqrt{\frac{2H(d_k^* \| d_k)}{1-\gamma}} \leq \sqrt{\frac{2\alpha\varepsilon_k}{1-\gamma}},$$

where in the last step we also used Lemma 5.8.1 that implies $H(d_k^* \| \widetilde{d}_k) \leq \alpha\varepsilon_k$. Putting this into expression (5.19) concludes the proof.

5.8.3 The proof of Theorem 5.4.2

This proof also relies on the lemmas and notation introduced in Section 5.8.1. The proof is based on using different properties of the optimization problem to express the quantity $\langle \mu^* - \mu_k^*, r \rangle$ in terms of different divergences:

$$\begin{aligned}D(\mu^* \| \mu_k^*) =& D(\mu^* \| d_{k-1}) - \sum_{x,a} \mu^*(x,a) \log \frac{\mu_k^*(x,a)}{d_{k-1}(x,a)} \\ =& D(\mu^* \| d_{k-1}) - \eta \langle \mu^*, r + \gamma P V_k^* - Q_k^* - \rho_k^* \rangle\end{aligned}$$

Now, by using that

$$\rho_k^* + (1-\gamma) \langle p_0, V_k^* \rangle = \langle \mu_k^*, r \rangle - \frac{1}{\eta} D(\mu_k^* \| d_{k-1}) - \frac{1}{\alpha} H(d_k^* \| d_{k-1})$$

and that

$$E^\mathsf{T} \mu^* = \gamma P^\mathsf{T} \mu^* + (1-\gamma) p_0,$$

we can write

$$\begin{aligned}D(\mu^* \| \mu_k^*) =& D(\mu^* \| d_{k-1}) - \eta \langle \mu^* - \mu_k^*, r \rangle + \eta \langle \mu^*, Q_k^* - E V_k^* \rangle \\ & - D(\mu_k^* \| d_{k-1}) - \frac{\eta}{\alpha} H(d_k^* \| d_{k-1}) \\ =& D(\mu^* \| d_{k-1}) - \eta \langle \mu^* - \mu_k^*, r \rangle - D(\mu_k^* \| d_{k-1}) - \frac{\eta}{\alpha} H(d_k^* \| d_{k-1}) \\ & + \eta \frac{H(\mu^* \| d_{k-1}) - H(\mu^* \| d_k^*)}{\alpha},\end{aligned}$$

where in the last equality we have used that

$$\begin{aligned}H(\mu^*\|d_k^*) &= \sum_{x,a}\mu^*(x,a)\log\frac{\pi^*(x,a)}{\pi_{k-1}} - \sum_{x,a}\mu^*(x,a)\log\frac{\pi_k^*(x,a)}{\pi_{k-1}} \\ &= H(\mu^*||d_{k-1}) - \alpha\left\langle\mu^*, Q_k^* - EV_k^*\right\rangle.\end{aligned}$$

Furthermore, we have

$$\begin{aligned}\langle d^* - d_k, r\rangle &\leq \langle d^* - d_k^*, r\rangle + \left\|d_k^* - d_k\right\|_1 \\ &\leq \langle d^* - d_k^*, r\rangle + \sqrt{2D(d_k^*\|d_k)} \\ &\leq \langle d^* - d_k^*, r\rangle + \sqrt{2\frac{1}{1-\gamma}H(d_k^*\|d_k)} \\ &\leq \langle \mu^* - \mu_k^*, r\rangle + \sqrt{\frac{2\alpha}{1-\gamma}\varepsilon_k},\end{aligned}$$

where in the second inequality we used Pinsker's inequality, in the third one we used Lemma 5.8.2, and in the last one we used Lemma 5.8.1 and that $\langle d_k^*, r\rangle = \langle \mu_k^*, r\rangle$ by Proposition 5.3.1.

Putting together the above results we get

$$\begin{aligned}\langle d^* - d_k, r\rangle \leq & \frac{D(\mu^*\|d_{k-1}) - D(\mu^*\|\mu_k^*)}{\eta} + \frac{H(\mu^*\|d_{k-1}) - H(\mu^*\|d_k^*)}{\alpha} \\ & - \frac{D(\mu_k^*\|d_{k-1})}{\eta} - \frac{H(d_k^*\|d_{k-1})}{\alpha} + \sqrt{\frac{2\alpha}{1-\gamma}\varepsilon_k}.\end{aligned}$$

Summing over epochs and rearranging terms concludes with the proof.

5.8.4 The proof of Theorem 5.4.3

The proof of this result is somewhat lengthy and is broken down into several lemmas that will be combined with the lemmas from Section 5.8.1 to proof the main result. We start recalling the definition of $\bar{\pi}_k$,

$$\bar{\pi}_k(a|x) = (1-\sigma)\pi_k(a|x) + \sigma\pi_0(a|x)$$

where π_0 is the uniform policy so $\pi_0(a|x) \geq \frac{1}{|\mathcal{A}|}$. Then, we can define the occupancy measure induced by $\bar{\pi}_k$ as

$$\bar{d}_k(x,a) = (1-\gamma)\mathbb{E}_{\bar{\pi}_k}\left[\sum_{t=0}^{\infty}\gamma^t\mathbb{I}_{\{(x_t,a_t)=(x,a)\}}\right].$$

We now overwrite the following definitions from the previous section:

$$\pi_k(a|x) = \bar{\pi}_{k-1}(a|x)e^{\alpha\left(Q_{\theta_k}(x,a)-V_{\theta_k}(x)\right)},$$

$$\pi_k^*(a|x) = \bar{\pi}_{k-1}(a|x)e^{\alpha\left(Q_{\theta_k^*}(x,a)-V_{\theta_k^*}(x)\right)},$$

$$\widetilde{\mu}_k(x,a) = \bar{d}_{k-1}(x,a)e^{\eta\left(\Delta_{\theta_k}(x,a)-\rho_k\right)},$$

and

$$\mu_k^*(x,a) = \bar{d}_{k-1}(x,a)e^{\eta\left(\Delta_{\theta_k^*}(x,a)-\rho_k^*\right)},$$

where we have changed π_{k-1} and d_{k-1} by $\bar{\pi}_{k-1}$ and $\bar{d}_{k-1}$ respectively to account for the σ-exploration. We also factorize the state-action occupancy measure induced by $\bar{\pi}_k$ as

$$\bar{d}_k(x,a) = \bar{\nu}_k(x)\bar{\pi}_k(a|x),$$

where $\bar{\nu}_k(x)$ is the state-occupancy measure induced by $\bar{\pi}_k(a|x)$. We can realize that all the lemmas stated in Section 5.8.1 still hold with these new definitions, since we have only modified the reference policy and its corresponding reference state-action occupancy measure. Also, recall that since we are considering the tabular setting, $\mu_k^* = d_k^*$ and $\mu^* = d^*$.

We define $\delta_\theta = r + PV_\theta - Q_\theta + (1-\gamma)\langle p_0, V_\theta\rangle \mathbf{1}$ and denote $\delta_k = \delta_{\theta_k}$ and $\delta_k^* = \delta_{\theta_k^*}$.With some abuse of notation we will denote

$$\mathcal{G}_k(\delta) = \frac{1}{\eta}\log\left(\sum_{x,a}\bar{d}_k(x,a)e^{\eta\delta(x,a)}\right),$$

with $\delta \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$. The difference between the function $\mathcal{G}(\delta)$ and the function $\mathcal{G}(\theta)$ with $\theta \in \mathbb{R}^m$ should be clear by the context. Realizing that $\mathcal{G}_k(\delta_\theta) = \mathcal{G}_k(\theta)$ for any θ justifies the abuse of notation. Also, realize that since the term $(1-\gamma)\langle p_0, V_k\rangle$ is a scalar, in the definitions of the state-action distributions $\widetilde{\mu}_k$ and μ_k^* , the terms Δ_{θ_k} and $\Delta_{\theta_k^*}$ can be substituted by δ_{θ_k} and $\delta_{\theta_k^*}$ with no effect.

The following lemma relates the suboptimality gap ε_k with the quantities δ_k and δ_k^*:

Lemma 5.8.3. *Define the state-action distribution*

$$\mu_k' = \frac{\bar{d}_{k-1}(x,a)e^{\eta\delta_k'(x,a)}}{\sum_{x',a'}\bar{d}_{k-1}(x',a')e^{\eta\delta_k'(x',a')}}$$

where $\delta_k' = \lambda\delta_k + (1-\lambda)\delta_k^*$ *for some* $\lambda \in (0,1)$ *to be determined. Then, defining* $X_k = \delta_k - \delta_k^*$*, there exists a* $\lambda \in (0,1)$ *such that the error* ε_k *is equal to the variance of* X_k *under the distribution* μ_k' *multiplied by* $\frac{\eta}{2}$*:*

$$\varepsilon_k = \frac{\eta}{2}\sum_{x,a}\mu_k'(x,a)\left(X_k(x,a) - \sum_{x',a'}\mu_k'(x',a')X_k(x',a')\right)^2$$

Proof. Using a second order tailor expansion of $\mathcal{G}_k(\delta_k)$, we have

$$\mathcal{G}_k(\delta_k) = \mathcal{G}_k(\delta_k^*) + \langle \delta_k - \delta_k^*, \nabla \mathcal{G}_k(\delta_k^*) \rangle + \frac{1}{2} \left\langle \delta_k - \delta_k^*, H_{\delta_k'}(\delta_k - \delta_k^*) \right\rangle$$

where $H_{\delta_k'}$ is the Hessian of $\mathcal{G}_k$ evaluated at $\delta_k' = \lambda\delta_k + (1-\lambda)\delta_k^*$ for some $\lambda \in (0,1)$. Rearranging terms and realizing that the gradient of $\mathcal{G}_k$ vanishes at δ_k^* we get

$$\varepsilon_k = \mathcal{G}_k(\delta_k) - \mathcal{G}_k(\delta_k^*) = \frac{1}{2} \left\langle \delta_k - \delta_k^*, H_{\delta_k'}(\delta_k - \delta_k^*). \right\rangle \tag{5.20}$$

Now, defining $\mu_{k,\delta}$ as

$$\mu_{k,\delta}(x,a) = \frac{\bar{d}_{k-1}(x,a)e^{\eta\delta(x,a)}}{\sum_{x',a'} \bar{d}_{k-1}(x',a')e^{\eta\delta(x',a')}},$$

the first and second derivative of $\mathcal{G}_k(\delta)$ w.r.t. δ can be written as

$$\frac{\partial \mathcal{G}_k}{\partial \delta(x,a)} = \mu_{k,\delta}(x,a)$$

and

$$\frac{\partial^2 \mathcal{G}_k}{\partial \delta(x,a)\partial\delta(x',a')} = \begin{cases} \eta(\mu_{k,\delta}(x,a) - \mu_{k,\delta}^2(x,a)) & if \quad (x,a) = (x',a') \\ -\eta\mu_{k,\delta}(x,a)\mu_{k,\delta}(x',a') & if \quad (x,a) \neq (x',a'). \end{cases}$$

By denoting $\mu_k' = \mu_{k,\delta'}$, the right hand side of equation (5.20) can now be written as

$$\begin{aligned} &\left\langle \delta_k - \delta_k^*, H_{\delta_k'}(\delta_k - \delta_k^*) \right\rangle \\ &\quad = \eta \sum_{x,a} (\delta_k(x,a) - \delta_k^*(x,a))^2 \mu_k'(x,a) \\ &\qquad - \eta \sum_{x,a} \sum_{x',a'} (\delta_k(x,a) - \delta_k^*(x,a))\mu_k'(x,a)(\delta_k(x',a') - \delta_k^*(x',a'))\mu_k'(x',a') \\ &\quad = \eta \sum_{x,a} (\delta_k(x,a) - \delta_k^*(x,a))^2 \mu_k'(x,a) - \eta \langle \delta_k - \delta_k^*, \mu_k' \rangle \langle \delta_k - \delta_k^*, \mu_k' \rangle \\ &\quad = \eta \sum_{x,a} (\delta_k(x,a) - \delta_k^*(x,a))^2 \mu_k'(x,a) \\ &\qquad - \eta \left(\sum_{x,a} (\delta_k(x,a) - \delta_k^*(x,a))\mu_k'(x,a) \right)^2 \end{aligned}$$

Which concludes the proof. □

The following Lemma presents a bound on the quantity $H(\mu^*\|d_k)-H(\mu^*\|d_k^*)$, which is one of the main parts of the proof:

Lemma 5.8.4.

$$H(\mu^*\|d_k) - H(\mu^*\|d_k^*) \leq \alpha e^{3\eta B}\sqrt{C_\gamma \frac{|\mathcal{A}|}{\sigma\eta}\varepsilon_k} + 2\alpha\sqrt{2\varepsilon_k}B + H(d_k^*||\bar{d}_{k-1}).$$

Proof.

$$\begin{aligned}
H(\mu^*\|d_k) - H(\mu^*\|d_k^*) = &\sum_{x,a}\mu^*(x,a)\left(\log\frac{\pi^*(x,a)}{\pi_k(x,a)} - \log\frac{\pi^*(x,a)}{\pi_k^*(x,a)}\right)\\
=&\sum_{x,a}\mu^*(x,a)\left(\log\frac{\pi_k^*(x,a)}{\pi_k(x,a)}\right)\\
=&\alpha\left\langle\mu^*, Q_k^* - Q_k - EV_k^* + EV_k\right\rangle\\
=&\alpha\left\langle\mu^*, Q_k^* - \gamma PV_k^* - (1-\gamma)\left\langle p_0, V_k^*\right\rangle\mathbf{1}\right\rangle\\
&-\alpha\left\langle\mu^*, Q_k - \gamma PV_k - (1-\gamma)\left\langle p_0, V_k\right\rangle\mathbf{1}\right\rangle\\
=&\alpha\left\langle\mu^*, \delta_k - \delta_k^*\right\rangle\\
=&\alpha\left\langle\mu^*, \delta_k - \delta_k^* - \left\langle\mu_k^*, \delta_k - \delta_k^*\right\rangle\mathbf{1}\right\rangle + \alpha\left\langle\mu_k^*, \delta_k - \delta_k^*\right\rangle
\end{aligned}$$

We start bounding the first term of the last line. For this, we will use the notation $a = \delta_k - \delta_k^* - \left\langle\mu_k^*, \delta_k - \delta_k^*\right\rangle\mathbf{1}$ so that the term that we want to bound becomes $\alpha\left\langle\mu^*, a\right\rangle$. Then, we have

$$\begin{aligned}
\left\langle\mu^*, a\right\rangle =&\sum_{x,a}\mu^*(x,a)a(x,a)\\
=&\sum_{x,a}\frac{\mu^*(x,a)}{(\mu_k^*(x,a))^{\frac{1}{2}}}(\mu_k^*(x,a))^{\frac{1}{2}}a(x,a)\\
\leq&\left(\sum_{x,a}\frac{(\mu^*(x,a))^2}{\mu_k^*(x,a)}\right)^{\frac{1}{2}}\left(\sum_{x,a}\mu_k^*(x,a)a^2(x,a)\right)^{\frac{1}{2}},
\end{aligned}$$

where in the last inequality we used Cauchy-Schwarz inequality. Using that $\mu_k^*(x,a) \geq \frac{\bar{d}_{k-1}(x,a)e^{-\eta B}}{\sum_{x',a'}\bar{d}_{k-1}(x',a')e^{\eta B}} \geq \bar{d}_{k-1}(x,a)e^{-2\eta B}$ and Assumption 6, we can see that

$$\begin{aligned}\left(\sum_{x,a}\frac{(\mu^*(x,a))^2}{\mu_k^*(x,a)}\right)^{\frac{1}{2}} &\leq \left(\sum_{x,a}\mu^*(x,a)\frac{\mu^*(x,a)e^{2\eta B}}{\bar{d}_{k-1}(x,a)}\right)^{\frac{1}{2}}\\ &\leq \left(\sum_{x,a}\mu^*(x,a)\frac{\nu^*(x)\pi^*(a|x)}{\bar{\nu}_{k-1}(x)\bar{\pi}_{k-1}(a|x)}e^{2\eta B}\right)^{\frac{1}{2}}\\ &\leq \left(\sum_{x,a}\mu^*(x,a)C_\gamma e^{2\eta B}\frac{|\mathcal{A}|}{\sigma}\right)^{\frac{1}{2}} = \left(C_\gamma e^{2\eta B}\frac{|\mathcal{A}|}{\sigma}\right)^{\frac{1}{2}}\end{aligned}$$

Now, recalling the definitions of μ' and X_k from the proof of Lemma 5.8.3, we can bound the term $\sum_{x,a}\mu_k^*(x,a)a^2(x,a)$ as follows:

$$\begin{aligned}\sum_{x,a}\mu_k^*(x,a)a^2(x,a) &= \sum_{x,a}\mu_k^*(x,a)\left(X(x,a)-\sum_{x',a'}\mu_k^*(x',a')X(x',a')\right)^2\\ &\leq \sum_{x,a}\mu_k^*(x,a)\left(X(x,a)-\sum_{x',a'}\mu_k'(x',a')X(x',a')\right)^2\\ &\leq \sum_{x,a}e^{4\eta B}\mu_k'(x,a)\left(X(x,a)-\sum_{x',a'}\mu_k'(x',a')X(x',a')\right)^2\\ &= \frac{2e^{4\eta B}}{\eta}\varepsilon_k\end{aligned}$$

Where in the first inequality we have used that if ν and ν' are two probability distributions, for a fixed ν, the expression $\sum_i \nu_i(X_i - \sum_j \nu_j' X_j)^2$ takes its minimum value when $\nu' = \nu$, in the second inequality we have used that

$$\begin{aligned}\mu_k^*(x,a) &= \frac{\bar{d}_{k-1}(x,a)e^{\eta\delta_k^*(x,a)}}{\sum_{x',a'}\bar{d}_{k-1}(x',a')e^{\eta\delta_k^*(x',a')}}\\ &= \frac{\mu_k'(x,a)e^{\eta(\delta_k^*(x,a)-\delta'(x,a))}}{\sum_{x',a'}\mu_k'(x',a')e^{\eta(\delta_k^*(x',a')-\delta'(x',a'))}}\\ &\leq \mu_k'(x,a)e^{4\eta B},\end{aligned}$$

and in the last equality we have used Lemma 5.8.3. Putting everything together we get

$$\alpha\left\langle \mu^*, \delta_k - \delta_k^* - \langle \mu_k^*, \delta_k - \delta_k^*\rangle \mathbf{1}\right\rangle \leq \alpha e^{3\eta B}\sqrt{\frac{2C_\gamma|\mathcal{A}|}{\sigma\eta}\varepsilon_k}.$$

Thus, the main bound becomes

$$H(\mu^*\|d_k) - H(\mu^*\|\mu_k^*) \leq \alpha e^{3\eta B}\sqrt{\frac{2C_\gamma|\mathcal{A}|}{\sigma\eta}\varepsilon_k} + \alpha\left\langle \mu_k^*, \delta_k - \delta_k^*\right\rangle$$

For the term $\langle \mu_k^*, \delta_k - \delta_k^*\rangle$, we can do the following:

$$\begin{aligned}\langle \mu_k^*, \delta_k - \delta_k^*\rangle &= \langle \mu_k^*, \gamma PV_k + (1-\gamma)\langle p_0, V_k\rangle \mathbf{1}\ - Q_k\rangle \\ &\quad - \langle \mu_k^*, \gamma PV_k^* + (1-\gamma)\langle p_0, V_k^*\rangle \mathbf{1}\ - Q_k^*\rangle \\ &= \langle \mu_k^*, EV_k - Q_k\rangle - \langle \mu_k^*, EV_k^* - Q_k^*\rangle\end{aligned}$$

Now, for the second term of the last line, we have

$$\begin{aligned}-\langle \mu_k^*, EV_k^* - Q_k^*\rangle =& -\sum_{x,a}\mu_k^*(x,a)(V_k^*(x) - Q_k^*(x,a)) \\ =& \sum_x \nu_k^*(x)\left(\sum_a \pi_k^*(a|x)Q_k^*(x,a) - V_k^*(x)\right) \\ =& \sum_x \nu_k^*(x)\left(\sum_a \pi_k^*(a|x)Q_k^*(x,a) - \sum_a \pi_k^*(a|x)Q_k^*(x,a)\right) \\ &+ \sum_x \nu_k^*(x)\left(\frac{1}{\alpha}D(\pi_k^*(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))\right) \\ =& \sum_x \nu_k^*(x)\left(\frac{1}{\alpha}D(\pi_k^*(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))\right) \\ =& \frac{1}{\alpha}H(d_k^*||\bar{d}_{k-1}),\end{aligned}$$

where we have used that $V_k(x) = \sum_a \pi_k(a|x)Q_k(x,a) - \frac{1}{\alpha}D(\pi_k(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))$. Similarly, for the first term we have

$$\begin{aligned}\langle \mu_k^*, EV_k - Q_k\rangle =& \sum_x \nu_k^*(x)\left(\sum_a \pi_k(a|x)Q_k(x,a) - \sum_a \pi_k^*(a|x)Q_k(x,a)\right) \\ &- \sum_x \nu_k^*(x)\left(\frac{1}{\alpha}D(\pi_k(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))\right) \\ \leq& \sum_{x,a}\nu_k^*(x)|\pi_k(a|x) - \pi_k^*(a|x)|B \\ &- \sum_x \nu_k^*(x)\left(\frac{1}{\alpha}D(\pi_k(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))\right) \\ \leq& \sum_{x,a}\nu_k^*(x)|\pi_k(a|x) - \pi_k^*(a|x)|B\end{aligned}$$

where we have used that $V_k(x) = \sum_a \pi_k(a|x)Q_k(x,a) - \frac{1}{\alpha}D(\pi_k(\cdot|x)||\bar{\pi}_{k-1}(\cdot|x))$ and Assumption 5. Now, by using Pinsker's inequality and Lemma 5.8.1, we get

$$\begin{aligned}
\langle \mu_k^*, EV_k - Q_k\rangle \leq & \sum_{x,a} \nu_k^*(x)|\pi_k(a|x) - \pi_k^*(a|x)|B \\
= & \sum_x \nu_k^*(x) \left\|\pi_k(\cdot|x) - \pi_k^*(\cdot|x)\right\|_1 B \\
= & \sum_x \sqrt{2(\nu_k^*(x))^2 D\left(\pi_k^*(\cdot|x)\|\pi_k(\cdot|x)\right)} B \\
\leq & \sqrt{2H\left(d_k^*\|d_k\right)} B \leq \sqrt{2\alpha\varepsilon_k} B
\end{aligned}$$

Thus, we have

$$\alpha \left\langle \mu_k^*, \delta_k - \delta_k^*\right\rangle \leq \alpha\sqrt{2\alpha\varepsilon_k}B + H(d_k^*||\bar{d}_{k-1})$$

Putting everything together gives

$$H(\mu^*\|d_k) - H(\mu^*\|d_k^*) \leq \alpha e^{3\eta B}\sqrt{\frac{2C_\gamma|\mathcal{A}|}{\sigma\eta}\varepsilon_k} + \alpha\sqrt{2\alpha\varepsilon_k}B + H(d_k^*||\bar{d}_{k-1}),$$

which concludes the proof. □

The following three lemmas will be used to take care of the effect of introducing the extra exploration, that is related to the difference between d_k and $\bar{d}_k$

Lemma 5.8.5. $H(\mu^*\|\bar{d}_k) - H(\mu^*\|d_k) \leq \sigma \log|\mathcal{A}|$

Proof.

$$\begin{aligned}
H(\mu^*\|\bar{d}_k) - H(\mu^*\|d_k) = & \sum_{x,a} \mu^*(x,a) \log\frac{\pi_k(a|x)}{\bar{\pi}_k(a|x)} = -\sum_{x,a} \mu^*(x,a) \log\frac{\bar{\pi}_k(a|x)}{\pi_k(a|x)} \\
= & -\sum_{x,a} \mu^*(x,a) \log\frac{(1-\sigma)\pi_k(a|x) + \sigma\pi_0(a|x)}{\pi_k(a|x)} \\
\leq & -(1-\sigma)\sum_{x,a} \mu^*(x,a) \log\frac{\pi_k(a|x)}{\pi_k(a|x)} \\
& - \sigma\sum_{x,a} \mu^*(x,a) \log\frac{\pi_0(a|x)}{\pi_k(a|x)} \\
= & \sigma\sum_{x,a} \mu^*(x,a) \left(\log\pi_k(a|x) - \log\pi_0(a|x)\right) \\
\leq & \sigma\log|\mathcal{A}|
\end{aligned}$$

Where in the first inequality we have used Jensen's inequality. □

Lemma 5.8.6. $D(d_k \| \bar{d}_k) \leq \frac{1}{1-\gamma} \sigma \log |\mathcal{A}|$

Proof.

$$
\begin{aligned}
D(d_k \| \bar{d}_k) \leq & \frac{1}{1-\gamma} H(d_k \| \bar{d}_k) \\
= & \frac{1}{1-\gamma} \sum_{x,a} d_k(x,a) \log \frac{\pi_k(a|x)}{\bar{\pi}_k(a|x)} \\
= & -\frac{1}{1-\gamma} \sum_{x,a} d_k(x,a) \log \frac{(1-\sigma)\pi_k(a|x) + \sigma \pi_0(a|x)}{\pi_k(a|x)} \\
\leq & \frac{1}{1-\gamma} \sigma \sum_{x,a} d_k(x,a) \log \frac{\pi_k(a|x)}{\pi_0(a|x)} \\
\leq & \frac{1}{1-\gamma} \sigma \sum_{x,a} d_k(x,a) \left(\log \pi_k(a|x) - \log \pi_0(a|x) \right) \\
\leq & \frac{1}{1-\gamma} \sigma \log |\mathcal{A}|.
\end{aligned}
$$

Where in the first inequality we used Lemma 5.8.2 and in the third one we have used Jensen's inequality similarly as in the previous proof. □

Lemma 5.8.7. $D(\mu^* \| \bar{d}_k) - D(\mu^* \| d_k) \leq C_\gamma \sqrt{\frac{2}{1-\gamma} \sigma \log |\mathcal{A}|}$

Proof.

$$
\begin{aligned}
D(\mu^* \| \bar{d}_k) - D(\mu^* \| d_k) = & \sum_{x,a} \mu^*(x,a) \log \frac{d_k(x,a)}{\bar{d}_k(x,a)} \\
\leq & \sum_{x,a} \mu^*(x,a) \left(\frac{d_k(x,a)}{\bar{d}_k(x,a)} - 1 \right) \\
= & \sum_{x} \frac{\mu^*(x,a)}{\bar{d}_k(x,a)} \big(d_k(x,a) - \bar{d}_k(x,a) \big) \leq C_\gamma \left\| d_k - \bar{d}_k \right\|_1 \\
\leq & C_\gamma \sqrt{2 D(d_k \| \bar{d}_k)} \leq C_\gamma \sqrt{\frac{2}{1-\gamma} \sigma \log |\mathcal{A}|}
\end{aligned}
$$

Where in the first inequality we have used that $\log(x) \leq x - 1$ for $x \geq -1$ and in the last inequality we have used Lemma 5.8.6. □

Lemma 5.8.8. $D(\nu^* \| \nu_k) - D(\nu^* \| \nu_k^*) \leq C_\gamma \sqrt{\frac{2\alpha}{1-\gamma} \varepsilon_k}$

Proof. This lemma can be proved by using again the inequality $\log(x) \leq x - 1$ for $x \geq -1$ and Lemma 5.8.2:

$$
\begin{aligned}
D(\nu^* \| \nu_k) - D(\nu^* \| \nu_k^*) &= \sum_x \nu^*(x) \log \frac{\nu_k^*(x)}{\nu_k(x)} \leq \sum_x \nu^*(x) \left(\frac{\nu_k^*(x)}{\nu_k(x)} - 1 \right) \\
&\leq \sum_x \frac{\nu^*(x)}{\nu_k(x)} |\nu_k^*(x) - \nu_k(x)| \leq C_\gamma \left\| \nu_k^* - \nu_k \right\|_1 \\
&\leq C_\gamma \sqrt{2D(\nu_k^* || \nu_k)} \leq C_\gamma \sqrt{2D(d_k^* || d_k)} \\
&\leq C_\gamma \sqrt{\frac{2}{1-\gamma} H(d_k^* || d_k)} \leq C_\gamma \sqrt{\frac{2\alpha}{1-\gamma} \varepsilon_k}
\end{aligned}
$$

□

Finally, we are ready to proof Theorem 5.4.3 with the help of the above lemmas.

Proof of Theorem 5.4.3.

$$
\begin{aligned}
D(\mu^* \| \mu_k^*) =& D(\mu^* \| \bar{d}_{k-1}) - \sum_{x,a} \mu^*(x,a) \log \frac{\mu_k^*(x,a)}{\bar{d}_{k-1}(x,a)} \\
=& D(\mu^* \| \bar{d}_{k-1}) - \eta \left\langle \mu^*, r + \gamma P V_k^* - Q_k^* - \rho_k^* \right\rangle \\
=& D(\mu^* \| \bar{d}_{k-1}) - \eta \left\langle \mu^* - \mu_k^*, r \right\rangle - D(\mu_k^* \| \bar{d}_{k-1}) + \eta \left\langle \mu^*, Q_k^* - E V_k^* \right\rangle \\
=& D(\mu^* \| \bar{d}_{k-1}) - \eta \left\langle \mu^* - \mu_k^*, r \right\rangle - D(\mu_k^* \| \bar{d}_{k-1}) - \frac{\eta}{\alpha} H(d_k^* \| \bar{d}_{k-1}) \\
& + \eta \frac{H(\mu^* \| \bar{d}_{k-1}) - H(\mu^* \| d_k^*)}{\alpha},
\end{aligned}
$$

where in the third equality we have used that

$$
\rho_k^* + (1-\gamma) \left\langle p_0, V_k^* \right\rangle = \left\langle \mu_k^*, r \right\rangle - \frac{1}{\eta} D(\mu_k^* \| \bar{d}_{k-1}) - \frac{1}{\alpha} H(d_k^* \| \bar{d}_{k-1})
$$

and in the last equality we have used that

$$
\begin{aligned}
H(\mu^* \| d_k^*) &= \sum_{x,a} \mu^*(x,a) \log \frac{\pi^*(x,a)}{\bar{\pi}_{k-1}} - \sum_{x,a} \mu^*(x,a) \log \frac{\pi_k^*(x,a)}{\bar{\pi}_{k-1}} \\
&= H(\mu^* || \bar{d}_{k-1}) - \alpha \left\langle \mu^*, Q_k^* - E V_k^* \right\rangle .
\end{aligned}
$$

Rearranging and summing and subtracting $\frac{D(\mu^*\|d_k^*)}{\eta}$, we have

$$\begin{aligned}\langle \mu^* - \mu_k^*, r\rangle =& \frac{D(\mu^*\|\bar{d}_{k-1}) - D(\mu^*\|d_k^*) - D(\mu_k^*\|\bar{d}_{k-1})}{\eta} \\ &+ \frac{H(\mu^*\|\bar{d}_{k-1}) - H(\mu^*\|d_k^*) - H(\mu_k^*\|\bar{d}_{k-1})}{\alpha} \\ &+ \frac{D(\mu^*\|d_k^*) - D(\mu^*\|\mu_k^*)}{\eta},\end{aligned}$$

where the terms from the last line cancel since we are in the tabular setting.

Summing and subtracting $\frac{D(\mu^*\|\bar{d}_k)+D(\mu^*\|d_k)}{\eta} + \frac{H(\mu^*\|d_k)+H(\mu^*\|\bar{d}_k)}{\alpha}$ in the right hand side of the above expression and rearranging we get

$$\begin{aligned}\langle \mu^* - \mu_k^*, r\rangle =& \frac{D(\mu^*\|\bar{d}_{k-1}) - D(\mu^*\|\bar{d}_k)}{\eta} - \frac{D(\mu_k^*\|\bar{d}_{k-1})}{\eta} - \frac{H(\mu_k^*\|\bar{d}_{k-1})}{\alpha} \\ &+ \frac{D(\nu^*\|\nu_k) - D(\nu^*\|\nu_k^*) + H(\mu^*\|d_k) - H(\mu^*\|d_k^*)}{\eta} \\ &+ \frac{H(\mu^*\|\bar{d}_{k-1}) - H(\mu^*\|\bar{d}_k) + H(\mu^*\|d_k) - H(\mu^*\|d_k^*)}{\alpha} \\ &+ \frac{D(\mu^*\|\bar{d}_k) - D(\mu^*\|d_k)}{\eta} + \frac{H(\mu^*\|\bar{d}_k) - H(\mu^*\|d_k)}{\alpha},\end{aligned}$$

where we also used the chain rule of relative entropy saying that if μ and μ' are state-action occupancy measures, and ν and ν' are their corresponding state occupancy measure, then $D(\mu\|\mu') = D(\nu\|\nu') + H(\mu\|\mu')$. Summing over epochs, we get

$$\begin{aligned}\sum_{k=1}^{K}\langle \mu^* - \mu_k^*, r\rangle \leq& \frac{D(\mu^*\|d_0)}{\eta} + \frac{H(\mu^*\|d_0)}{\alpha} \\ &- \sum_{k=1}^{K}\left(\frac{D(\mu_k^*\|\bar{d}_{k-1})}{\eta} + \frac{H(\mu_k^*\|\bar{d}_{k-1})}{\alpha}\right) \\ &+ \sum_{k=1}^{K}\frac{D(\nu^*\|\nu_k) - D(\nu^*\|\nu_k^*)}{\eta} \\ &+ \sum_{k=1}^{K}\left(\frac{1}{\eta} + \frac{1}{\alpha}\right)\left(H(\mu^*\|d_k) - H(\mu^*\|d_k^*)\right) \\ &+ \sum_{k=1}^{K}\frac{D(\mu^*\|\bar{d}_k) - D(\mu^*\|d_k)}{\eta} + \frac{H(\mu^*\|\bar{d}_k) - H(\mu^*\|d_k)}{\alpha}.\end{aligned} \tag{5.21}$$

Using this last bound and Lemmas 5.8.4, 5.8.5, 5.8.7 and 5.8.8, the above bound becomes

$$\begin{aligned}\sum_{k=1}^{K} \langle \mu^* - \mu_k^*, r \rangle \leq \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(\mu^* \| d_0)}{\alpha} \\ - \sum_{k=1}^{K} \left(\frac{D(\mu_k^* \| \bar{d}_{k-1})}{\eta} + \frac{H(\mu_k^* \| \bar{d}_{k-1})}{\alpha} \right) \\ + \sum_{k=1}^{K} \frac{C_\gamma}{\eta} \sqrt{\frac{2\alpha}{1-\gamma} \varepsilon_k} \\ + \sum_{k=1}^{K} \left(1 + \frac{\alpha}{\eta}\right) \left(e^{3\eta B} \sqrt{\frac{2C_\gamma |\mathcal{A}|}{\sigma\eta}} + B\sqrt{2\alpha} \right) \sqrt{\varepsilon_k} \\ + \sum_{k=1}^{K} \left(1 + \frac{\alpha}{\eta}\right) \left(\frac{H(\mu_k^* \| \bar{d}_{k-1})}{\alpha} \right) \\ + \sum_{k=1}^{K} \left(\frac{C_\gamma}{\eta} \sqrt{\frac{2\sigma}{1-\gamma} \log |\mathcal{A}|} + \frac{\sigma}{\alpha} \log |\mathcal{A}| \right).\end{aligned}$$

We can see that $\left(1 + \frac{\alpha}{\eta}\right) \frac{H(\mu_k^* \| \bar{d}_{k-1})}{\alpha}$ is smaller than $\left(\frac{D(\mu_k^* \| \bar{d}_{k-1})}{\eta} + \frac{H(\mu_k^* \| \bar{d}_{k-1})}{\alpha} \right)$ so we can eliminate them from the bound. Rearranging we get

$$\begin{aligned}\sum_{k=1}^{K} \langle \mu^* - \mu_k^*, r \rangle \leq & \frac{D(\mu^* \| d_0)}{\eta} + \frac{H(\mu^* \| d_0)}{\alpha} \\ & + \sum_{k=1}^{K} \frac{C_\gamma}{\eta} \sqrt{\frac{2\alpha}{1-\gamma} \varepsilon_k} \\ & + \sum_{k=1}^{K} \left(1 + \frac{\alpha}{\eta}\right) \left(e^{3\eta B} \sqrt{\frac{2C_\gamma |\mathcal{A}|}{\sigma\eta}} + B\sqrt{2\alpha} \right) \sqrt{\varepsilon_k} \\ & + \sum_{k=1}^{K} \left(\frac{C_\gamma}{\eta} \sqrt{\frac{2\sigma}{1-\gamma} \log |\mathcal{A}|} + \frac{\sigma}{\alpha} \log |\mathcal{A}| \right).\end{aligned}$$

We now need to connect $\langle \mu^* - \mu_k^*, r \rangle$ with $\left\langle d^* - \bar{d}_k, r \right\rangle$. Using Pinsker's inequality and Lemmas 5.8.1, 5.8.2 and 5.8.6 we can write the following:

$$
\begin{aligned}
\left\langle d^* - \bar{d}_k, r \right\rangle &= \langle d^* - d_k^*, r\rangle + \langle d_k^* - d_k, r\rangle + \left\langle d_k - \bar{d}_k, r \right\rangle \\
&\leq \langle d^* - d_k^*, r\rangle + \|d_k^* - d_k\|_1 + \left\| d_k - \bar{d}_k \right\|_1 \\
&\leq \langle d^* - d_k^*, r\rangle + \sqrt{2D(d_k^* \| d_k)} + \sqrt{2D(d_k \| \bar{d}_k)} \\
&\leq \langle d^* - d_k^*, r\rangle + \sqrt{2\frac{1}{1-\gamma} H(d_k^* \| d_k)} + \sqrt{2D(d_k \| \bar{d}_k)} \\
&= \langle \mu^* - \mu_k^*, r\rangle + \sqrt{\frac{2\alpha}{1-\gamma}\varepsilon_k} + \sqrt{\frac{2\sigma}{1-\gamma}\log|A|}
\end{aligned}
$$

Putting this result into equation (5.8.4) concludes the proof.

□

5.8.5 The proof of Theorem 5.5.1

We will prove the following, more general version of the theorem below:

Theorem 5.8.2. *(General statement) Let $\mathcal{Q} = \{Q_\theta : \|Q_\theta\|_\infty \leq B'\}$ for some $B' > 0$ and Θ be the corresponding set of parameter vectors, and let $\mathcal{N}_{\mathcal{Q},\epsilon}$ be the ϵ-covering number of $\mathcal{Q}$ with respect to the ℓ_∞ norm. Furthermore, define $B = 1 + (1+\gamma)B'$, and assume that $\eta B \leq 1$ holds. Then, with probability at least $1-\delta$, the following holds:*

$$
\sup_{\theta \in \Theta} \left| \widehat{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) \right| \leq 8\eta B^2 + 56\sqrt{\frac{\log(2\mathcal{N}_{\mathcal{Q},1/\sqrt{N}}/\delta)}{2N}}.
$$

The proof of the version stated in Theorem 5.5.1 follows from bounding the covering number of our linear logistic Q-function class as $\mathcal{N}_{\mathcal{Q},\epsilon} \leq (1 + 4B/\epsilon)^m$.

Proof. We first prove a concentration bound for a fixed θ and then provide a uniform guarantee through a covering argument.

For the first part, let us fix a confidence level $\delta' > 0$ and an arbitrary θ, and define the shorthand notation $\widehat{S}_n = \widehat{\Delta}_\theta(X_{k,n}, A_{k,n}, X'_{k,n})$ and $S_n = \Delta_\theta(X_{k,n}, A_{k,n})$. Note that, by definition, these random variables are bounded in the interval $[-(\gamma + 1)B', 1 + (\gamma+1)B'] \subset [-B, B]$. Furthermore, let us define the notation $\mathbb{E}_{X'}[\cdot] = \mathbb{E}\left[\cdot \middle| \{X_{k,n}, A_{k,n}\}_{n=1}^N\right]$ and let

$$
W = \frac{1}{N}\sum_{n=1}^N e^{\eta \widehat{S}_n} \qquad \text{and} \qquad \overline{W} = \frac{1}{N}\sum_{n=1}^N e^{\eta S_n}.
$$

We start by observing that, by Jensen's inequality, we obviously have $\mathbb{E}_{X'}\left[W\right] \le \overline{W}$. Furthermore, by using the inequality $e^u \le 1 + u + u^2$ that holds for all $u \le 1$, we can further write

$$\begin{aligned}\overline{W} &\le \frac{1}{N}\sum_{n=1}^{N}\left(1 + \eta S_n + \eta^2 S_n^2\right) \le \mathbb{E}_{X'}\left[\frac{1}{N}\sum_{n=1}^{N}\left(1 + \eta \widehat{S}_n\right)\right] + \eta^2 S_n^2 \\ &\le \mathbb{E}_{X'}\left[\frac{1}{N}\sum_{n=1}^{N} e^{\eta \widehat{S}_n}\right] + \eta^2 S_n^2 = \mathbb{E}_{X'}\left[W\right] + \eta^2 B^2,\end{aligned}$$

where in the last line we used the inequality $1 + u \le e^u$ that holds for all u and our upper bound on $\widehat{S}_n$. Thus, taking expectations with respect to X', we get

$$\mathbb{E}\left[W\right] \le \mathbb{E}\left[\overline{W}\right] \le \mathbb{E}\left[W\right] + \eta^2 B^2. \tag{5.22}$$

To proceed, we define the function

$$f(s_1, s_2, \dots, s_N) = \frac{1}{N}\sum_{n=1}^{N} e^{\eta s_n}$$

and notice that it satisfies the bounded-differences property

$$\begin{aligned}f(s_1, s_2, \dots, s_n, \dots, s_N) - f(s_1, s_2, \dots, s_n', \dots, s_N) &= \frac{1}{N}\left(e^{\eta s_n} - e^{\eta s_n'}\right) \\ &\le \frac{\eta e^{2\eta B}}{N}.\end{aligned}$$

Here, the last step follows from Taylor's theorem that implies that there exists a $\chi \in (0,1)$ such that

$$e^{\eta s_n'} = e^{\eta s_n} + \eta e^{\eta \chi (s_n' - s_n)}$$

holds, so that $e^{\eta s_n'} - e^{\eta s_n} = \eta e^{\eta \chi (s_n' - s_n)} \le \eta e^{2\eta B}$, where we used the assumption that $|s_n - s_n'| \le 2B$ in the last step. Notice that our assumption $\eta B \le 1$ further implies that $e^{2\eta B} \le e^2$. Thus, also noticing that $W = f(S_1, \dots, S_N)$, we can apply McDiarmid's inequality that to show that the following holds with probability at least $1 - \delta'$:

$$\left|W - \mathbb{E}\left[W\right]\right| \le \eta e^2 \sqrt{\frac{\log(2/\delta')}{2N}}. \tag{5.23}$$

Now, let us observe that the difference between the LBE and its empirical counterpart can be written as

$$\widehat{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) = \frac{1}{\eta}\log\left(W\right) - \frac{1}{\eta}\log\left(\mathbb{E}\left[\overline{W}\right]\right) = \frac{1}{\eta}\log\left(\frac{W}{\mathbb{E}\left[\overline{W}\right]}\right).$$

Thus, by combining Equations (5.22) and (5.23), we obtain that

$$\begin{aligned}\widehat{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) &= \frac{1}{\eta}\log\left(1 + \frac{W - \mathbb{E}\left[\overline{W}\right]}{\mathbb{E}\left[\overline{W}\right]}\right) \leq \frac{1}{\eta}\log\left(1 + \frac{W - \mathbb{E}\left[W\right]}{\mathbb{E}\left[\overline{W}\right]}\right) \\ &\leq \frac{W - \mathbb{E}\left[W\right]}{\eta\mathbb{E}\left[\overline{W}\right]} \leq e^4\sqrt{\frac{\log(2/\delta')}{2N}},\end{aligned}$$

where we used the inequality $\log(1+u) \leq u$ that holds for $u > -1$ and our assumption on η that implies $\overline{W} \geq e^{-2}$. Similarly, we can show

$$\begin{aligned}\mathcal{G}_k(\theta) - \widehat{\mathcal{G}}_k(\theta) &= \frac{1}{\eta}\log\left(1 + \frac{\mathbb{E}\left[\overline{W}\right] - W}{W}\right) \\ &\leq \frac{1}{\eta}\log\left(1 + \frac{\mathbb{E}\left[W\right] - W + \eta^2 B^2}{W}\right) \\ &\leq \frac{\mathbb{E}\left[W\right] - W + \eta^2 B^2}{\eta W} \leq e^4\sqrt{\frac{\log(2/\delta')}{2N}} + \eta e^2 B^2,\end{aligned}$$

This concludes the proof of the concentration result for a fixed θ.

In order to prove a bound that holds uniformly for all values of θ, we will consider a covering of the space of Q functions Q_θ bounded in terms of the supremum norm $\mathcal{Q} = \{Q_\theta : \theta \in \mathbb{R}^m, \left\|Q_\theta\right\|_\infty \leq B\}$. The corresponding set of parameters will be denoted as Θ. To define the covering, we fix an $\epsilon > 0$ and consider a set $\mathcal{C}_{\mathcal{Q},\epsilon} \subset \mathcal{Q}$ of minimum cardinality, such that for all $Q_\theta \in \mathcal{Q}$, there exists a $\theta' \in \mathcal{C}_{\mathcal{Q},\epsilon}$ satisfying $|\mathcal{G}_k(\theta) - \mathcal{G}_k(\theta')| \leq \epsilon$. Defining the covering number $\mathcal{N}_{\mathcal{Q},\epsilon} = |\mathcal{C}_{\mathcal{Q},\epsilon}|$ and $\epsilon = 1/\sqrt{N}$, we can combine the above concentration result with a union bound over the covering $\mathcal{C}_{\mathcal{Q},\epsilon}$ to get that

$$\sup_{\theta\in\Theta}\left|\mathcal{G}_k(\theta) - \widehat{\mathcal{G}}_k(\theta)\right| \leq \left(e^4 + 1\right)\sqrt{\frac{\log(2\mathcal{N}_{\mathcal{Q},\epsilon}/\delta)}{2N}} + \eta e^2 B^2$$

holds with probability at least $1 - \delta$. Upper-bounding the constants $e^2 < 8$ and $e^4 + 1 < 56$ concludes the proof. □

5.8.6 The proof of Theorem 5.5.2

The proof of this Theorem follows the same reasoning as the previous one. We proof the following general version of the Theorem:

Theorem 5.8.2. *(General statement) Let $\mathcal{Q} = \{Q_\theta : \left\|Q_\theta\right\|_\infty \leq B'\}$ for some $B' > 0$ and Θ be the corresponding set of parameter vectors, and let $\mathcal{N}_{\mathcal{Q},\epsilon}$ be* *the ϵ-covering number of $\mathcal{Q}$ with respect to the ℓ_∞ norm. Furthermore, define $B = 1 + (1+\gamma)B'$, and assume that $\eta B \leq 1$ holds. Then, with probability at least $1-\delta$, the following holds:*

$$\sup_{\theta\in\Theta} \left| \widetilde{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) \right| \leq 56\sqrt{\frac{\log(2\mathcal{N}_{\mathcal{Q},1/\sqrt{N}}/\delta)}{2N}}.$$

By bounding the covering number of our linear logistic Q-function class as $\mathcal{N}_{\mathcal{Q},\epsilon} \leq (1 + 4B/\epsilon)^m$ we proof the version of the Theorem stated in Theorem 5.5.2.

Proof. As in the previous proof, we start fixing a confidence level $\delta' > 0$ and an arbitrary θ. We also reuse the notation

$$\overline{W} = \frac{1}{N}\sum_{n=1}^{N} e^{\eta S_n}.$$

Then, using that $\overline{W} = f(S_1, \ldots, S_N)$ for $f(s_1, s_2, \ldots, s_N)$ defined in the same way as in the previous proof, together with McDiarmid's inequality, we get that the following holds with probability at least $1-\delta'$:

$$|\overline{W} - \mathbb{E}\left[\overline{W}\right]| \leq \eta e^2 \sqrt{\frac{\log(2/\delta')}{2N}}. \tag{5.24}$$

Now, we can do the following:

$$\widetilde{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) = \frac{1}{\eta}\log\left(\overline{W}\right) - \frac{1}{\eta}\log\left(\mathbb{E}\left[\overline{W}\right]\right) = \frac{1}{\eta}\log\left(\frac{\overline{W}}{\mathbb{E}\left[\overline{W}\right]}\right).$$

Thus, by combining Equations (5.22) and (5.24), we obtain that

$$\begin{aligned}\widetilde{\mathcal{G}}_k(\theta) - \mathcal{G}_k(\theta) &= \frac{1}{\eta}\log\left(\overline{W}\right) - \frac{1}{\eta}\log\left(\mathbb{E}\left[\overline{W}\right]\right) = \frac{1}{\eta}\log\left(\frac{\overline{W}}{\mathbb{E}\left[\overline{W}\right]}\right) \\ &= \frac{1}{\eta}\log\left(1 + \frac{\overline{W} - \mathbb{E}\left[\overline{W}\right]}{\mathbb{E}\left[\overline{W}\right]}\right) \leq \frac{\overline{W} - \mathbb{E}\left[\overline{W}\right]}{\eta\mathbb{E}\left[\overline{W}\right]} \leq e^4\sqrt{\frac{\log(2/\delta')}{2N}},\end{aligned}$$

where we used the inequality $\log(1+u) \leq u$ that holds for $u > -1$ and our assumption on η that implies $\overline{W} \geq e^{-2}$. Similarly, we can show

$$\mathcal{G}_k(\theta) - \widetilde{\mathcal{G}}_k(\theta) = \frac{1}{\eta}\log\left(1 + \frac{\mathbb{E}\left[\overline{W}\right] - \overline{W}}{\overline{W}}\right) \leq \frac{\mathbb{E}\left[\overline{W}\right] - \overline{W}}{\eta\overline{W}} \leq e^4\sqrt{\frac{\log(2/\delta')}{2N}},$$

This concludes the proof of the concentration result for a fixed θ. By following the same reasoning as in the previous Theorem, we get that

$$\sup_{\theta \in \Theta} \left| \mathcal{G}_k(\theta) - \widetilde{\mathcal{G}}_k(\theta) \right| \leq (56) \sqrt{\frac{\log(2\mathcal{N}_{\mathcal{Q},\epsilon}/\delta)}{2N}}.$$

□

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this work we explored some of the possibilities that the linear programming approach for optimal control in MDPs can bring to reinforcement learning. We saw how different convex optimization tools and techniques can be used to derive efficient large-scale reinforcement-learning algorithms from this LP formulation.

One of the main ideas that this work tries to highlight is how building algorithms based on the LP approach has the clear benefit of giving an objective function that can be optimized with modern large-scale optimization methods. Furthermore, since those algorithms are fully based on convex optimization, the analysis become simple and transparent, and we can derive theoretical guarantees regarding their behaviour and performance. This is in part thanks to the extensive literature in convex optimization that, apart from providing a handful of tools and theory to develop efficient algorithms, allows us to analyze those algorithms in a straightforward manner. In concrete, in Chapters 4 and 5 we showed how these tools can be used successfully to derive algorithms that can work in large-scale problems, and we believe that it is a really promising direction to keep bringing theoretically sound algorithms.

One of the central lines of this work has been the study of the implications of relaxing the constraints and adding different kinds of regularization to the original LPs. Understanding the effect of both the relaxations and the regularization in the resulting optimization problems is crucial to derive meaningful algorithms. A key tool during the whole work has been to move between the primal and the dual to take advantage of both. When doing so, it is not clear at all (a priori) what will be the repercussion of using the different kinds of constraint relaxation and regularization. Indeed, our results are proved under rather restrictive assumptions and it remains unclear if our methods continue to perform well beyond these well-controlled scenarios. Understanding the subtle impact of these choices is of vital importance when trying to derive algorithms with desirable properties. In this sense, we believe that our work has generated some new knowledge that may be valuable for guiding future work on designing more efficient large-scale RL algorithms.

In Chapter 4, we studied a relaxed saddle-point linear problem for finding optimal policies. There, one of the main results was the characterization of a set of assumptions that allow a reduced-order saddle-point representation of the optimal policy. In particular, we showed that realizability is not enough to ensure the relaxed problem to be a good enough approximation of the original one and argued that a the newly identified coherence assumption is also necessary. This coherence assumption concerns the subspaces used for approximation and ensures that the value functions that appear as dual variables are able to penalize non-stationary probability distributions, making sure that the solution of the primal problem is stationary. This characterization is very transparent and gives insight about the problems that can appear when dealing with relaxations of this kind. Furthermore, it allows a clear analysis of optimization algorithms in this setting. These results open the door to asking what happens when the two assumptions are relaxed, which is a very interesting question that we leave as an open problem for future work. In the rest of this chapter we used the literature regarding the mirror prox algorithm to derive an efficient algorithm for policy optimization, and we explored different analysis techniques to study the performance of the output policy. Of special interest are the techniques used for connecting the duality gap of the solution output by our algorithm with the actual performance of the policy extracted from that solution, and we believe that these tools can be used to analyze similar algorithms. One of the main limitations of our model is that it requires full knowledge of the transition function, which is quite restrictive if we consider the reinforcement learning setting. We leave as future work the exploration of adaptations of our algorithm that make it possible to work with sample transitions.

In Chapter 5, we presented `Q-REPS`, a new RL algorithm with very desirable properties. The main features of `Q-REPS` that make it particularly interesting are the usage of Q-functions (in contrast to its predecessor `REPS`) that enable efficient model-free implementation, and a convex loss function for policy evaluation that due to its favourable properties could serve as an alternative to the widely used squared Bellman error. In this chapter we saw how this algorithm is derived entirely from a regularized and linearly relaxed version of a particular LP formulation of optimal control in MDPs. Again, this shows how a smart usage of convex optimization techniques such as constraint relaxation, Lagrangian decomposition of constraints, regularization and Lagrangian duality can be used to derive practical algorithms entirely rooted in theory, and how the convex analysis techniques can be then used to analyze these algorithms in a very transparent way.

Nevertheless, we also saw that our analysis have some shortcomings that should be addressed in future work. First, it would be of particular interest to be able to bound the Q-functions coming from the regularized problem, since it is a requisite for the analysis to work. Second, we strongly believe that the restrictive assumptions that we needed to derive the convergence guarantee are artifacts of our analysis. For this reason, we find it important as a next step to remove these restrictive assumptions and derive similar bounds for the original version of `Q-REPS` in more general setting than the tabular one. The factored linear MDP setting is a good candidate for which we think that it should be possible to derive such guarantees. This is because, as we showed, in this setting the relaxation that we proposed suffers from no approximation error. We also leave as future work to understand approximation errors without such structural assumption, and to study other more sophisticated relaxation methods.

Other important directions for future work are related to dealing with implementation aspects of `Q-REPS` that can be challenging in practical problems. Here, the first serious issue is the requirement of sampling from the discounted occupancy measure, that can not be done efficiently without having access to a reset action. The second issue regarding implemantability of `Q-REPS` is the necessity of storing the cumulative sum of all past logistic Q-functions that we find in the current algorithm specification, that is the one for which we have theoretical guarantees. Future works and algorithms following the same principles as `Q-REPS` should take into account these implementation issues and try to address them.

In conclusion, we think that the main takeaway of this project is that by working with the linear programming approach of optimal control in MDPs, we can derive algorithms based on convex optimization that are fully rooted to theory and with very desirable properties. Furthemore the convex optimization analysis allows us to understand these algorithms and derive performance bounds. We believe that this is a promising direction of research that will bring efficient large-scale algorithms with strong theoretical guarantees.

Bibliography

Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvári, Cs., and Weisz, G. (2019). Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020a). Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020b). Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory*, pages 64–66.

Antos, A., Szepesvári, Cs., and Munos, R. (2006). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. In *COLT 2006*, pages 574–588.

Ayoub, A., Jia, Z., Szepesvári, Cs., Wang, M., and Yang, L. F. (2020). Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*.

Bagnell, J. A. and Schneider, J. (2003). Covariant policy search.

Bas-Serrano, J., Curi, S., Krause, A., and Neu, G. (2021). Logistic q-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3610–3618. PMLR.

Bas-Serrano, J. and Neu, G. (2020). Faster saddle-point optimization for solving large-scale markov decision processes. In *Learning for Dynamics and Control*, pages 413–423. PMLR.

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.

Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 3 edition.

Bertsekas, D. P. (2008). Approximate dynamic programming.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities:A Nonasymptotic Theory of Independence*. Oxford University Press.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Bubeck, S. (2014). Convex Optimization: Algorithms and Complexity. 8(3):231–357.

Buşoniu, L., Lazaric, A., Ghavamzadeh, M., Munos, R., Babuška, R., and Schutter, B. D. (2012). Least-squares methods for policy iteration. *Reinforcement learning*, pages 75–109.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. (2020). Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR.

Chen, Y., Li, L., and Wang, M. (2018). Scalable bilinear π learning using state and action features. In *International Conference on Machine Learning*, pages 833–842.

Cheng, C.-A., Combes, R. T., Boots, B., and Gordon, G. (2020). A reduction from reinforcement learning to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3514–3524. PMLR.

Curi, S. (2020). Rl-lib - a pytorch-based library for reinforcement learning research. Github.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Liu, Z., Chen, J., and Song, L. (2018). SBEED: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning*, pages 1125–1134. PMLR.

de Farias, D. P. and Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865.

de Ghellinck, G. (1960). Les problèmes de décisions séquentielles. *Cahiers du Centre d'Études de Recherche Opérationnelle*, 2:161–179.

Deisenroth, M., Neumann, G., and Peters, J. (2013). A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142.

Denardo, E. V. (1970). On linear programming in a Markov decision problem. *Management Science*, 16(5):281–288.

Desai, V. V., Farias, V. F., and Moallemi, C. C. (2012). Approximate dynamic programming via a smoothed linear program. *Operations Research*, 60(3):655–674.

Even-Dar, E., Kakade, S. M., and Mansour, Y. (2009). Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736.

Feng, Y., Li, L., and Liu, Q. (2019). A kernel loss for solving the Bellman equation. In *Advances in Neural Information Processing Systems*, pages 15456–15467.

Fujimoto, S., van Hoof, H., Meger, D., et al. (2018). Addressing function approximation error in actor-critic methods. *Proceedings of Machine Learning Research*, 80.

Furmston, T. and Barber, D. (2010). Variational methods for reinforcement learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 241–248.

Geist, M., Piot, B., and Pietquin, O. (2017). Is the Bellman residual a bad proxy? In *Advances in Neural Information Processing Systems*, pages 3205–3214.

Geist, M., Scherrer, B., and Pietquin, O. (2019). A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. The MIT Press, Cambridge, MA.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143.

Jin, Y. and Sidford, A. (2020). Efficiently solving MDPs with stochastic mirror descent. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 4890–4900. PMLR.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274.

Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14:1531–1538.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Konda, V. and Tsitsiklis, J. (1999). Actor-critic algorithms. *Advances in neural information processing systems*, 12:1008–1014.

Lakshminarayanan, C. and Bhatnagar, S. (2015). A generalized reduced linear program for Markov decision processes. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, volume 15 of *AAAI*, pages 2722–2728.

Lakshminarayanan, C., Bhatnagar, S., and Szepesvári, C. (2017). A linearly relaxed approximate linear program for markov decision processes. *IEEE Transactions on Automatic control*, 63(4):1185–1191.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lazaric, A., Ghavamzadeh, M., and Munos, R. (2010). Finite-sample analysis of LSTD. In *ICML 2010*, pages 615–622.

Lee, D. and He, N. (2019). Stochastic primal-dual Q-learning algorithm for discounted MDPs. In *2019 American Control Conference (ACC)*, pages 4897–4902. IEEE.

Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*, volume 107. American Mathematical Soc.

Manne, A. S. (1960). Linear programming and sequential decisions. *Management Science*, 6(3):259–267.

Martinet, B. (1970). Régularisation d'inéquations variationnelles par approximations successives. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 4(R3):154–158.

Mehta, P. and Meyn, S. (2009). Q-learning and Pontryagin's minimum principle. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3598–3605. IEEE.

Mehta, P. G. and Meyn, S. P. (2020). Convex q-learning, part 1: Deterministic optimal control. *arXiv preprint arXiv:2008.03559*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857.

Nemirovski, A. (2004). Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251.

Neu, G., György, A., Szepesvári, Cs., and Antos, A. (2014). Online Markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59:676–691.

Neu, G., Jonsson, A., and Gómez, V. (2017). A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*.

Neu, G. and Pike-Burke, C. (2020). A unifying view of optimism in episodic reinforcement learning. *arXiv preprint arXiv:2007.01891*.

Nota, C. and Thomas, P. S. (2019). Is the policy gradient a gradient? *arXiv preprint arXiv:1906.07073*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Peters, J., Mulling, K., and Altun, Y. (2010). Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612.

Petrik, M. and Zilberstein, S. (2009). Constraint relaxation in approximate linear programs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 809–816.

Puterman, M. L. (1990). Markov decision processes. *Handbooks in operations research and management science*, 2:331–434.

Puterman, M. L. (1994). Appendix D: Linear Programming. *Markov Decision Processes Discrete Stochastic Dynamic Programming*, pages 610–612.

Rakhlin, A. and Sridharan, K. (2013). Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Rockafellar, R. T. (1976). Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., Lesner, B., and Geist, M. (2015). Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16:1629–1676.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schweitzer, P. J. and Seidmann, A. (1985). Generalized polynomial approximations in markovian decision processes. *Journal of mathematical analysis and applications*, 110(2):568–582.

Seneta, E. (2006). *Non-negative matrices and Markov chains*. Springer Science & Business Media.

Song, H. F., Abdolmaleki, A., Springenberg, J. T., Clark, A., Soyer, H., Rae, J. W., Noury, S., Ahuja, A., Liu, S., Tirumala, D., et al. (2019). V-mpo: on-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238*.

Strehl, A. L. and Littman, M. L. (2008). An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331.

Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction (second edition)*. online draft.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12:1057–1063.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103.

Thomas, P. (2014). Bias in natural actor-critic algorithms. In *International conference on machine learning*, pages 441–448. PMLR.

Tsitsiklis, J. N. and Van Roy, B. (1997). An analysis of temporal difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42:674–690.

Vieillard, N., Kozuno, T., Scherrer, B., Pietquin, O., Munos, R., and Geist, M. (2020a). Leverage the average: an analysis of regularization in rl. *arXiv preprint arXiv:2003.14089*.

Vieillard, N., Pietquin, O., and Geist, M. (2020b). Munchausen reinforcement learning. *Advances in Neural Information Processing Systems*, 33:4235–4246.

Wang, M. (2017). Primal-dual π learning: Sample complexity and sublinear run time for ergodic Markov decision problems. *arXiv preprint arXiv:1710.06100*.

Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. (2020). On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826.

Xie, T. and Jiang, N. (2020). Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Uncertainty in Artificial Intelligence*, pages 550–559.

Yang, L. F. and Wang, M. (2019). Sample-optimal parametric Q-learning using linearly additive features. In *36th International Conference on Machine Learning, ICML 2019*, pages 12095–12114.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.

Zimin, A. and Neu, G. (2013). Online learning in episodic markovian decision processes by relative entropy policy search. *Advances in neural information processing systems*, 26:1583–1591.