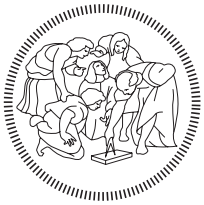


POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY

EXPLOITING ENVIRONMENT CONFIGURABILITY IN REINFORCEMENT LEARNING

Doctoral Dissertation of:
Alberto Maria Metelli

Supervisor:
Prof. Marcello Restelli

Tutor:
Prof. Nicola Gatti

The Chair of the Doctoral Program:
Prof. Barbara Pernici

2020 – Cycle XXXIII

Abstract

In the last decades, *Reinforcement Learning* (RL) has emerged as an effective approach to address complex control tasks. The formalism typically employed to model the sequential interaction between the artificial agent and the environment is the *Markov Decision Process* (MDP). In an MDP, the agent perceives the state of the environment and performs actions. As a consequence, the environment transitions to a new state and generates a reward signal. The goal of the agent consists of learning a policy, i.e., a prescription of actions, that maximizes the long-term reward.

In the traditional setting, the environment is assumed to be a fixed entity that cannot be altered externally. However, there exist several real-world scenarios in which the environment can be modified to a limited extent and, therefore, it might be beneficial to act on some of its features. We call this activity *environment configuration*, that can be carried out by the agent itself or by an external entity, such as a configurator. Although environment configuration arises quite often in real applications, this topic is very little explored in the literature.

In this dissertation, we aim at formalizing and studying the diverse aspects of environment configuration. The contributions are theoretical, algorithmic, and experimental and can be broadly subdivided into three parts.

The first part of the dissertation introduces the novel formalism of *Configurable Markov Decision Processes* (Conf-MDPs) to model the configuration opportunities offered by the environment. At an intuitive level, there exists a tight connection between environment, policy, and learning process. We explore the different nuances of environment configuration, based on whether the configuration is fully auxiliary to the agent's learning process (cooperative setting) or guided by a configurator having an objective that possibly conflicts with the agent's one (non-cooperative setting).

In the second part, we focus on the cooperative Conf-MDP setting and we investigate the *learning problem* consisting of finding an agent policy and an environment configuration that jointly optimize the long-term reward. We provide algorithms for solving finite and continuous Conf-MDPs and experimental evaluations are conducted on both synthetic and realistic domains.

The third part addresses two specific applications of the Conf-MDP framework: *policy space identification* and *control frequency adaptation*. In the former, we employ environment configurability to improve the identification of the agent's perception and actuation capabilities. In the latter, instead, we analyze how a specific configurable environmental parameter, the control frequency, can affect the performance of the batch RL algorithms.

Sommario

Negli ultimi decenni, l'*Apprendimento per Rinforzo* (Reinforcement Learning, RL) è emerso come un approccio efficace per affrontare complessi problemi di controllo. Il formalismo che viene solitamente impiegato per modellare l'interazione sequenziale tra l'agente artificiale e l'ambiente è il *Processo Decisionale di Markov* (Markov Decision Process, MDP). In un MDP, l'agente percepisce lo stato dell'ambiente e compie delle azioni. Come conseguenza, l'ambiente evolve in un nuovo stato e genera un segnale di ricompensa. L'obiettivo dell'agente consiste nell'apprendere una politica, cioè una prescrizione di azioni, che massimizza la ricompensa di lungo periodo.

Tradizionalmente, l'ambiente è considerato un'entità fissa che non può essere alterata dall'esterno. Tuttavia, esistono numerosi scenari reali in cui l'ambiente può essere modificato in modo limitato e, pertanto, può risultare conveniente agire su alcune delle sue proprietà. Chiamiamo questa attività *configurazione dell'ambiente*, che può essere effettuata dall'agente stesso o da un'entità esterna, come un configuratore. Nonostante la configurazione dell'ambiente emerga piuttosto frequentemente nelle applicazioni reali, questo argomento è esplorato molto poco nella letteratura.

In questa dissertazione, intendiamo formalizzare e studiare i vari aspetti della configurazione dell'ambiente. I contributi sono teorici, algoritmici e sperimentali e possono essere suddivisi, a grandi linee, in tre parti.

La prima parte della dissertazione introduce il nuovo formalismo dei *Processi Decisionali di Markov Configurabili* (Configurable Markov Decision Processes, Conf-MDPs) per modellare le opportunità di configurazione offerte dall'ambiente. A livello intuitivo, esiste una stretta connessione tra ambiente, politica e processo di apprendimento. Esploriamo le diverse sfumature della configurazione dell'ambiente, a seconda che la configurazione sia esclusivamente ausiliaria al processo di apprendimento dell'agente (contesto cooperativo) o sia guidata da un configuratore con un obiettivo eventualmente conflittuale con quello dell'agente (contesto non cooperativo).

Nella seconda parte, ci concentriamo sui Conf-MDP cooperativi e investighiamo il *problema di apprendimento* che consiste nel trovare una politica dell'agente e una configurazione dell'ambiente che congiuntamente ottimizzano la ricompensa di lungo periodo.

Forniamo algoritmi per risolvere Conf-MDP finiti e continui e valutazioni sperimentali condotte sia in domini sintetici che realistici.

La terza parte affronta due specifiche applicazioni dei Conf-MDP: l'*identificazione dello spazio delle politiche* e l'*adattamento della frequenza di controllo*. Nel primo caso, facciamo uso della configurabilità dell'ambiente per migliorare l'identificazione delle capacità di percezione e attuazione dell'agente. Nel secondo caso, invece, analizziamo come uno specifico parametro configurabile dell'ambiente, la frequenza di controllo, possa impattare sulla performance degli algoritmi di RL batch.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols and Notation

Sets and Arithmetic

$\mathbb{N}$	set of natural numbers $\{0, 1, \dots\}$
$\mathbb{N}_{\geqslant l}$	set of natural numbers greater or equal than l
$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geqslant l}$	set of real numbers greater or equal than l
$\lvert\mathcal{X}\rvert$	cardinality of set $\mathcal{X}$
$2^{\mathcal{X}}$	power set of $\mathcal{X}$
$a \bmod b$	remainder of the integer division between $a \in \mathbb{N}$ and $b \in \mathbb{N}_{\geqslant 1}$
$a \operatorname{div} b$	quotient of the integer division between $a \in \mathbb{N}$ and $b \in \mathbb{N}_{\geqslant 1}$

Measures, Integration, and Operators

$\mathfrak{F}_{\mathcal{X}}$	a σ-algebra defined over a set $\mathcal{X}$
$\mathfrak{B}(\mathcal{X})$	Borel σ-algebra defined over a topological space $\mathcal{X}$
$\mathfrak{F}_1 \otimes \mathfrak{F}_2$	tensor-product σ-algebra induced by the σ-algebras $\mathfrak{F}_1$ and $\mathfrak{F}_2$
$\mathscr{M}(\mathcal{X})$	set of signed measures defined over the measurable space $(\mathcal{X}, \mathfrak{F}_{\mathcal{X}})$
$\mathscr{P}(\mathcal{X})$	set of probability measures defined over the measurable space $(\mathcal{X}, \mathfrak{F}_{\mathcal{X}})$
$\mathscr{B}(\mathcal{X})$	set of bounded measurable functions defined over $(\mathcal{X}, \mathfrak{F}_{\mathcal{X}})$ with real values
$X \sim \mu$	X is a random variable sampled from the probability measure μ
$\mu \ll \nu$	μ is absolutely continuous w.r.t. ν

$\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$	Radon-Nikodym derivative of μ w.r.t. ν
δ_x	Dirac delta measure centered in $x \in \mathcal{X}$
$\mathbf{1}_{\mathcal{U}}$	indicator function of the measurable set $\mathcal{U}$
$\mathop{\mathbb{E}}_{X\sim\mu}[f(X)]$	the expectation of the measurable function f under probability measure μ, i.e., $\mathbb{E}_{X\sim\mu}[f(X)] = \int_{\mathcal{X}} \mu(\mathrm{d}x) f(x)$
$\mathop{\mathbb{P}}_{X\sim\mu}[X \in \mathcal{U}]$	probability that random variable X belongs to the measurable set $\mathcal{U}$, i.e., $\mathbb{P}_{X\sim\mu}[X \in \mathcal{U}] = \mathbb{E}_{X\sim\mu}[\mathbf{1}_{\mathcal{U}}(X)]$
$\mathrm{Id}_{\mathcal{X}}$	identity operator on $\mathscr{B}(\mathcal{X})$, i.e., $\mathrm{Id}_{\mathcal{X}} f = f$
$\mathrm{M}_{\mathcal{X}}$	maximization operator on $\mathscr{B}(\mathcal{X})$, i.e., $\mathrm{M}_{\mathcal{X}} f = \sup_{x\in\mathcal{X}}\{f(x)\}$
a.s.	almost surely

Norms and Divergences

$\|f\|_{p,\mu}$	$L_p(\mu)$-norm under probability measure μ and $p \in [1,\infty)$, i.e., $\|f\|_{p,\mu}^p = \int_{\mathcal{X}} \mu(\mathrm{d}x)\, \|f(x)\|^p$
$\|f\|_{p,\mathcal{D}}$	empirical $L_p(\mu)$-norm computed using the dataset $\mathcal{D} = \{X_1, \ldots, X_n\} \subset \mathcal{X}$ and $p \in [1,\infty)$, i.e., $\|f\|_{p,\mathcal{D}}^p = \frac{1}{n}\sum_{i=1}^n \|f(X_i)\|^p$
$\|f\|_{\infty}$	L_∞-norm of, i.e., $\|f\|_{\infty} = \sup_{x\in\mathcal{X}} \{f(x)\}$
$\|\mu\|_{\mathrm{TV}}$	total variation norm of the signed measure μ
$\|Q\|_{\mathrm{TV},\nu}$	expectation of the total variation norm of $Q(\cdot\|x)$ under the probability measure ν, i.e., $\|Q\|_{\mathrm{TV},\nu} = \int_{\mathcal{X}} \nu(\mathrm{d}x)\, \|Q(\cdot\|x)\|_{\mathrm{TV}}$
$\mathrm{sp}(f)$	span seminorm of function f, i.e., $\mathrm{sp}(f) = \sup_{x\in\mathcal{X}}\{f(x)\} - \inf_{x\in\mathcal{X}}\{f(x)\}$
$D_\alpha(\mu\|\nu)$	α-Rényi divergence, with $\alpha \in [0,\infty]$, between μ and ν, i.e., $D_\alpha(\mu\|\nu) = \frac{1}{\alpha-1}\log \int_{\mathcal{X}} \left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right)^{\alpha} \mathrm{d}\nu$
$D_{\mathrm{KL}}(\mu\|\nu)$	KL-divergence between μ and ν, i.e., $D_{\mathrm{KL}}(\mu\|\nu) = \int_{\mathcal{X}} \log \frac{\mathrm{d}\nu}{\mathrm{d}\mu} \mathrm{d}\nu$

Operator Notation

μf	expectation of function f under the measure μ, i.e., $\mu f = \int_{\mathcal{X}} \mu(\mathrm{d}x) f(x)$
νQ	expectation of the conditional measure $Q(\cdot\|x)$ under the measure ν, i.e., $\nu Q = \int_{\mathcal{X}} \nu(\mathrm{d}x) Q(\cdot\|x)$

Markov Decision Processes

$\mathcal{S}$	state space
$\mathcal{A}$	action space
P	transition model

R	reward model
r	reward function
μ_0	initial state distribution
γ	discount factor
π	policy
Π^{SR}	set of Markovian stationary policies
$\mathcal{P}^{\mathrm{SR}}$	set of Markovian stationary transition models
Π^{HR}	set of history-dependent policies
$\mathcal{P}^{\mathrm{HR}}$	set of history-dependent transition models
P^π	state or state-action probability kernel
μ^π_γ	state or state-action γ-discounted stationary distribution
μ^π	state or state-action stationary distribution
$\mathcal{T}$	set of infinite-length trajectories
$\mathbb{P}^\pi$	infinite-length trajectory distribution
$\mathbb{p}^\pi$	probability density function of $\mathbb{P}^\pi$
$\mathbb{E}^\pi$	expectation operator under $\mathbb{P}^\pi$, i.e., $\mathbb{E}^\pi[f] = \mathbb{E}_{\tau\sim\mathbb{P}^\pi}[f(\tau)]$
G_γ	return of a trajectory
V^π	state value function of policy π
Q^π	state-action value function of policy π
A^π	advantage function of policy π
J^π	expected return of policy π
J^μ	expected return of the γ-discounted stationary distribution μ
T^π	Bellman expectation operator of policy π
V^*	optimal state value function
Q^*	optimal state-action value function
A^*	optimal advantage function
J^*	optimal expected return
T^*	Bellman optimality operator

Configurable Markov Decision Processes

R_{Ag} (R_{Conf})	agent's (configurator's) reward model[1]
r_{Ag} (r_{Conf})	agent's (configurator's) reward function

[1] The Ag and Conf subscripts are omitted for cooperative Conf-MDPs, i.e., when $r_{\mathsf{Ag}} = r_{\mathsf{Conf}}$.

$\mu_\gamma^{\pi,P}$	state, state-action, or state-action-next-state γ-discounted stationary distribution
$\mu^{\pi,P}$	state, state-action, or state-action-next-state stationary distribution
$\mathbb{P}_{\mathsf{Ag}}^{\pi,P}$ ($\mathbb{P}_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) infinite-length trajectory distribution for the agent
$\mathbb{p}_{\mathsf{Ag}}^{\pi,P}$ ($\mathbb{p}_{\mathsf{Conf}}^{\pi,P}$)	probability density function of $\mathbb{P}_{\mathsf{Ag}}^{\pi,P}$ ($\mathbb{P}_{\mathsf{Conf}}^{\pi,P}$)
$\mathbb{E}_{\mathsf{Ag}}^{\pi,P}$ ($\mathbb{E}_{\mathsf{Conf}}^{\pi,P}$)	expectation operator under $\mathbb{P}_{\mathsf{Ag}}^{\pi,P}$ ($\mathbb{P}_{\mathsf{Conf}}^{\pi,P}$), i.e., $\mathbb{E}_{\mathsf{Ag}}^{\pi,P}[f] = \mathbb{E}_{\tau\sim\mathbb{P}_{\mathsf{Ag}}^{\pi,P}}[f(\tau)]$ ($\mathbb{E}_{\mathsf{Conf}}^{\pi,P}[f] = \mathbb{E}_{\tau\sim\mathbb{P}_{\mathsf{Conf}}^{\pi,P}}[f(\tau)]$)
$V_{\mathsf{Ag}}^{\pi,P}$ ($V_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) state value function of policy π and transition model P
$Q_{\mathsf{Ag}}^{\pi,P}$ ($Q_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) state-action value function of policy π and transition model P
$U_{\mathsf{Ag}}^{\pi,P}$ ($U_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) state-action-next-state value function of policy π and transition model P
$A_{\mathsf{Ag}}^{\pi,P}$ ($A_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) policy or model advantage functions of policy π and transition model P
$\widetilde{A}_{\mathsf{Ag}}^{\pi,P}$($\widetilde{A}_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) coupled advantage function of policy π and transition model P
$J_{\mathsf{Ag}}^{\pi,P}$ ($J_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) expected return of policy π and transition model P
$T_{\mathsf{Ag}}^{\pi,P}$ ($T_{\mathsf{Conf}}^{\pi,P}$)	agent's (configurator's) Bellman expectation operator of policy π and transition model P
$V_{\mathsf{Ag}}^{*,P}$ ($V_{\mathsf{Conf}}^{\pi,*}$)	agent's (configurator's) best response state value function
$Q_{\mathsf{Ag}}^{*,P}$ ($Q_{\mathsf{Conf}}^{\pi,*}$)	agent's (configurator's) best response state-action value function
$U_{\mathsf{Ag}}^{*,P}$ ($U_{\mathsf{Conf}}^{\pi,*}$)	agent's (configurator's) best response state-action-next-state value function
$J_{\mathsf{Ag}}^{*,P}$ ($J_{\mathsf{Conf}}^{\pi,*}$)	agent's (configurator's) best response expected return
$T_{\mathsf{Ag}}^{*,P}$ ($T_{\mathsf{Conf}}^{\pi,*}$)	agent's (configurator's) best response Bellman operator
$V^{*,*}$	optimal cooperative state value function
$Q^{*,*}$	optimal cooperative state-action value function
$U^{*,*}$	optimal cooperative state-action-next-state value function
$J^{*,*}$	optimal cooperative expected return
$T^{*,*}$	optimal cooperative Bellman operator

CHAPTER 1

Introduction

Machine Learning (ML) is rapidly becoming pervasive in our world. Nowadays, we are constantly inundated by huge amounts of data coming from an always growing spectrum of sources: newspapers, radio, television, websites, social networks. In the meantime, we have at our disposal powerful computational tools that we bring with us wherever we go: smart-phones, tablets, computers. Data and information, as its refinement, are at the basis of any *decision-making* process. Everyday we make decisions based on the available information. Clearly, as information is essential for this process, in the meantime, an overload of information might be dangerous as well. Today, more than ever, the effective employment of information for decision-making has become a strategic goal; for the governments, clearly, but also for ordinary people.

Considered the huge amount of data in play, that cannot be managed by human being, at least in its crude form, we must resort to automatic, algorithmic, methods. ML provides suitable tools for this purpose. Tom M. Mitchell defined ML as "*the study of computer algorithms that allow computer programs to automatically improve through experience*" (Mitchell, 1997). We immediately notice that the main character of this process is a *computer*, or, using a more technical lexicon, an *artificial agent*. This definition implicitly depicts the mechanism at the basis of the mentioned improvements: the presence of experience, i.e., fresh information, that triggers a process of *learning*. Learning is at the basis of the biological and intellectual development of any living being and experience is the engine of this process. Therefore, ML is in all respects a part of *Artificial Intelligence* (AI). The peculiar feature of ML is the constant presence of data. Data are usually generated by natural or artificial processes that are typically affected by noise. Consequently,

the learning process is intrinsically performed *under uncertainty*. This makes probability and statistics reference tools for ML, which is sometimes referred to as *statistical learning* (Hastie et al., 2009) to highlight this connection.

From a taxonomic point of view, ML paradigms can be subdivided into three categories, based on the fundamental features of the problem they address. *Supervised learning* aims at mapping data (input) to a value (output or target), that can be either a symbol or a real value. The learning process involves observing a *training* dataset of input-output pairs with the goal of inferring the *pattern* hidden in the dataset (Bishop, 2007; Mohri et al., 2012). Within supervised learning, we can distinguish between *regression* if the output is a real number or *classification* if, instead, the target is a class from a finite set. Supervised learning is probably the most widespread and developed area of ML. Examples of successful applications are image classification (Lu and Weng, 2007), which has nowadays overcome the human performance, recommendation systems (Bobadilla et al., 2013), hand-written recognition (Puigcerver, 2017), to mention a few. Another paradigm of ML is *unsupervised learning*, whose goal consists in identifying patterns in the data without having a target value to predict (Ghahramani, 2003). Examples of unsupervised learning tasks are clustering (Xu and Tian, 2015), anomaly detection (Chandola et al., 2009), and latent variable models (Skrondal and Rabe-Hesketh, 2007). Finally, the third area of ML is *Reinforcement Learning* (RL, Sutton and Barto, 2018), where the high-level goal consists of learning a sequence of decisions in an unknown environment, so as to maximize some utility function. Thus, while in supervised and unsupervised learning there is no notion of sequentiality, as the decision is one-shot and it has no consequences on the future, in RL the sequential nature of the interaction is essential. In a sense, RL takes a perspective that is closer to classical AI (Russell and Norvig, 2010) in which the presence of an agent performing decisions is explicit, while in the other paradigms the role of the agent tends to be more blurred. Finally, we can look at RL as the most general ML setting, since both supervised and unsupervised learning can be reduced to it.

1.1 What is Reinforcement Learning?

When we think of the process of learning for human beings, we realize that *interaction* with the surrounding environment plays a crucial role. Human beings acquire abilities in different ways, but all of them involve a certain degree of interaction with either the external environment or other agents (biological or artificial). A baby, an example of a biological agent, learns how to walk in a *trial and error* fashion. They try the first movement and then they likely fall down, so they try another one and, sooner or later, they manage to stay upright. No teacher is, in principle, needed in this process, as the effects of the movement are associated with a feedback signal (falling down or staying upright) that tells the baby whether it was profitable or not. This feedback triggers an adjustment in the behavior and, hopefully, over multiple trials, leads to the realization of the ultimate goal of walking. Thus, as supported by intuition, exercising the connection between the agent and the environment helps the former to figure out, and consequently exploit, the causal relations linking the actions to their effects in the specific environment. Clearly, numerous and diverse examples of analogous learning processes carried out by human beings exist, such as learning how to drive a car, how to play chess, or how to cook a cake. All of them are characterized by the same basic ingredients: an agent interacting with an

environment and a feedback signal evaluating the success of the actions the agent plays. These elements constitute the fundamental elements of RL. The term *reinforcement* was introduced for the first time in behavioral psychology and defined by Burrhus F. Skinner as "*a consequence applied that will strengthen an organism's future behavior whenever that behavior is preceded by a specific antecedent stimulus*" (Skinner, 1938; Schultz, 2015).

In this dissertation, we take the AI perspective and we focus on RL as the "*computational approach to learning from interaction*" (Sutton and Barto, 2018). The entities involved are the (artificial) *agent* and the *environment*. The agent is characterized by some perception and actuation capabilities. The perception defines the ability to measure the state of the environment. Thus, the perceived (or observed) state can be either the complete internal environment state, in this case, we speak of full observability, or an *observation* hiding some features, i.e., we are in a partially observable setting. The actuation possibilities, instead, are concerning the ability to perform actions on the environment. Whenever an action is played, it produces an evolution of the environment state and the agent is provided with a feedback signal, the *immediate reward*. According to the AI terminology, the agent is goal-directed, i.e., it acts with the purpose of finding the proper actions, so as to maximize some utility function. In RL, such a utility function is defined as a notion of *long-term reward*, i.e., the cumulative (possibly discounted) sum of the immediate rewards collected during the agent's experience. This closed-loop interaction, despite being a simplification of the one actually carried out by biological agents, is sufficiently expressive to model numerous interesting real-world situations, such as controlling an industrial robot (Meyes et al., 2017; Gu et al., 2017), autonomous driving (Kiran et al., 2020), playing videogames (Mnih et al., 2013, 2015), robotic locomotion (Haarnoja et al., 2019).

1.2 Why Environment Configurability?

Besides the remarkable success demonstrated in recent years, RL appears to be deeply rooted in the definition of the environment as an immutable entity out of any control. In the traditional model, the agent can *indirectly* control the environment by means of the performed actions, but cannot *directly* change the environment dynamics. This is certainly true in a large number of applications, although we can identify a huge number of examples in which a "partial control" on the environment can be exercised.

For instance, a human car driver has at their disposal a number of possible vehicle *configurations* they can act on (e.g., seasonal tires, stability, vehicle attitude, engine model, automatic speed control, and parking aid system) to improve the driving style or quicken the process of learning a good driving policy. Another example is the interaction between a student and an automatic teaching system: the teaching model can be tailored to improve the student's learning experience (e.g., increasing or decreasing the difficulties of the questions or the speed at which the concepts are presented). It is worth noting that the active entity in this configuration process might be the agent itself or an external *supervisor* (or *configurator*) guiding the learning process. Another example is product placement in a supermarket. A supervisor can dynamically adapt where to place the products in order to maximize customer satisfaction. Differently from the previous examples, it might be possible that the configurator (e.g., the supermarket staff) has a goal that is different from that of the agent (e.g., the customer). Similarly, a street network could be configured, by

changing the semaphore transition times or the direction of motion. The goal of the network designer is to limit/control the average traffic, whereas the drivers try to reduce their journey time. A similar setting arises in the project of a website in which the user desires to find the information they need as fast as possible, whereas the website owner wants to orientate the user towards specific pages or contents.

In all these scenarios, whenever altering some portions of the environment or some environmental parameters is allowed, we speak of *environment configurability*. Environment configuration arises in several real-world scenarios, with different objectives, involving different levels of cooperation and competition between agents and configurators. Therefore, we believe, the nature of this kind of interaction deserves additional study and this dissertation pursues this high-level goal. Before presenting the concrete contribution of the dissertation, we briefly discuss why, in our opinion, the models already employed in the literature turn out to be inappropriate to capture the peculiarities of environment configurability.

Why not a unique agent? Representing the environment configurability in the agent's model when the environment is under the control of an external configurator is certainly inappropriate. Even when the environment configuration is carried out by the agent itself, this approach would require the inclusion of "configuration actions" to allow the agent to configure the environment directly as a part of its policy. However, the configuration activity cannot be placed at the same level as the agent's learning process. Configuring the environment may be more expensive and dangerous than updating the agent's policy and may occur on a different time scale w.r.t. the agent's learning process. Moreover, such a formulation would prevent distinguishing, during the process, the effects of the policy from those of the environment, making it difficult to finely constrain the configurations, and recovering, a posteriori, the agent's policy.

Why not a multi-agent system? When there is no supervisor, the agent is the only learning entity and the environment is completely passive. Even, in the presence of a supervisor, adopting a multi-agent approach would be misleading and would certainly introduce a complexity that is not needed. The supervisor acts externally, at a different level and could be, possibly, totally transparent to the learning agent. Indeed, the supervisor does not operate inside the environment but it is in charge of selecting a suitable configuration, based on its interests (and possibly on those of the agent), whereas the agent has to learn an optimal behavior in the given environment. In this sense, the configurator could be thought of as an agent in a *hierarchical* multi-agent RL problem (Ghavamzadeh et al., 2006). Nevertheless, this framework introduces additional issues related to communication and cooperation, that are not considered in our Conf-MDP.

1.3 Original Contributions

This dissertation pursues essentially three goals, that correspond to the three parts in which it is subdivided. First, we aim at *formalizing* the notion of environment configurability, study its properties, and provide suitable solution concepts. Second, we address the *learning* problem, i.e., the problem of finding an optimal agent's policy and environment

configuration. Third, we study some applicative scenarios in which environment configurability plays a central role. In the following, we survey the main contributions of the three parts of the dissertation.

1.3.1 Modeling Environment Configurability

The first part of the dissertation (Part I) is devoted to the formalization of environment configurability and represents mainly a theoretical contribution. We propose a novel extension of the traditional Markov Decision Process (MDP, Puterman, 2014), named *Configurable Markov Decision Process* (Conf-MDP), in order to properly represent the configurability possibilities of the environment. We extend the traditional tools for MDPs to Conf-MDPs, including value functions, and we propose suitable Bellman operators and the corresponding Bellman equations.

Then, we devise a taxonomy for the Conf-MDPs, according to the properties of the interaction between the agent and the configurator. Specifically, we identify different settings based on whether the agent is aware of the configurator presence and whether their objectives coincide. This latter distinction reveals two wide settings that characterize the interaction between the agent and the supervisor: the cooperative and the non-cooperative setting.

For the cooperative setting, we introduce the optimality conditions which define when an agent's policy together with an environment configuration can be considered optimal. Moreover, we show that this setting can be reduced to a standard MDP and, thus, it inherits most of the properties of the traditional case, including the existence of an optimal policy and environment configuration pair.

Then, we focus on the non-cooperative setting in which defining a notion of optimality is less immediate. Indeed, when the configurator has a goal that is different from that of the agent, we need to resort to game-theoretic equilibria in order to obtain a suitable solution concept. Depending on whether the agent is aware of the supervisor, we propose to employ different equilibria (Shapley, 1953). For both settings, we present the corresponding value functions and, whenever possible, we discuss the extensions of the Bellman operators and equations.

1.3.2 Learning in Cooperative Configurable Markov Decision Processes

In the second part of the dissertation (Part II), we focus on the cooperative Conf-MDP setting and we study the learning problem consisting of finding an agent's policy together with an environment configuration so as to maximize the long-term reward. This part represents primarily an algorithmic and experimental contribution.

We start with the simpler setting in which the environment is characterized by a finite state-action space. We propose a learning algorithm, *Safe Policy Model Iteration* (SPMI), inspired by the safe learning approaches to RL (Kakade and Langford, 2002; Pirotta et al., 2013b), that updates the policy and the environment configuration based on the maximization of a lower bound on the performance improvement. This way it is possible to derive strong theoretical guarantees on the performance gain between two consecutive iterations. We present an experimental evaluation on two illustrative domains, inspired by the mo-

tivating examples of Conf-MDPs, to show the advantages of environment configuration over traditional fixed-environment learning.

Then, we move to the continuous environment case and we devise an approach, able to overcome the main limitations of SPMI, i.e., the need for knowing the environment model and the fact that it can be applied to finite state-action problems only. The new algorithm, *Relative Entropy Model Policy Search* (REMPS) lies in the family of trust-region methods (Schulman et al., 2015) and extends REPS (Peters et al., 2010) to account for environment configuration learning. REMPS looks for the stationary distribution that maximizes the long-term reward, by constraining the search in a neighborhood of the current sampling distribution. We empirically evaluate REMPS on both synthetic and realistic domains, including an experiment for the car configuration task, built on top of the TORCS simulator (Loiacono et al., 2010).

1.3.3 Applications of Configurable Markov Decision Processes

The last part of the dissertation (Part III) is dedicated to the study of two applicative scenarios in which environment configuration can play a relevant role. This part includes algorithmic, theoretical, and experimental contributions.

The first application we examine is *policy space identification*, i.e., the problem of identifying the space of policies that an agent can access during the learning process. The notion of optimal policy, in a learning process, it tightly connected to the agent's perception and actuation possibilities, combined with its ability to map states to actions. Knowing the agent's policy space can be particularly convenient when environment configuration is possible. Indeed, agents optimizing the same objective but having access to different policy spaces might benefit from different environment configurations. We propose two *identification rules* based on likelihood ratio testing (Barnard, 1959; Casella and Berger, 2002) to identify the policy parameters an agent can access. Environment configurability is also exploited to place the agent in a suitable configuration in which it is induced to reveal the parameters it can access. Empirical results to validate the approach are presented as well as applications on Conf-MDPs and Imitation Learning (IL, Osa et al., 2018).

The second application we explore is the *control frequency adaptation*. RL problems are typically formulated as discrete-times problems, but often derive from the time discretization of continuous-time ones. Thus, the control frequency is a relevant design choice and can be considered an environmental parameter that can be configured. We address the setting in which we are provided with a finely discretized MDP and we model the reduction of the control frequency as the repetition of an action for a fixed amount of consecutive steps, called *persistence*. We show how varying the persistence affects the agent's performance. Then, we provide an algorithm, *Persistent Fitted Q-Iterations* (PFQI), inspired to FQI (Ernst et al., 2005), able to learn the value function at different persistences and we propose a heuristic persistence selection method. PFQI is evaluated on benchmark domains as well as in a realistic trading environment.

This dissertation reports the content of four research papers. Three of them are published at ICML (International Conference on Machine Learning) and one is currently under review for Machine Learning (Springer). Table 1.1 reports the list of papers, including the publication venue, the link to the paper, the link to the code, and the contributions

of the author of this dissertation. Throughout the dissertation, we decided to focus the presentation on the methodological aspects, favoring the theoretical and algorithmic contributions. To make the presentation fluid, we decided not to include some material related to experimental evaluations that were instead provided in the original papers, but we report suitable references whenever necessary.

1.4 Overview

The dissertation is organized in three parts that are preceded by two chapters that introduce the foundations of sequential decision-making and reinforcement learning. These chapters must not be intended to provide an exhaustive overview of the topic, but just as an introduction, tailored to the needs of the subsequent chapters. We conclude the dissertation with a chapter that provides a discussion on the research contributions and on future directions. The detailed organization of the dissertation is described in the following.

- Chapter 2 introduces the fundamental notions of sequential decision-making, including the definition of Markov Decision Process (Puterman, 2014), policy, value functions, Bellman operators, and Bellman equations (Bellman, 1957). We discuss the optimality criteria employed for MDPs and we provide a brief overview of the exact solution methods.

- Chapter 3 provides a background on a selection of RL algorithms (Sutton and Barto, 2018), whose knowledge is essential to understand the subsequent chapters. We focus on temporal difference methods (Watkins and Dayan, 1992), approximate value and policy iteration (Munos, 2003; Scherrer, 2014), and policy search (Deisenroth et al., 2013).

Part I: Modeling Environment Configurability

This part aims at analyzing how to model the configuration opportunities offered by the environment and discuss the solution concepts suitable for the different natures of interaction between the agent and the configurator. Specifically, the contributions are organized in two chapters.

- Chapter 4 provides the motivations behind environment configuration, introduces the definition of Configurable Markov Decision Process and extends the notions of value function, Bellman operators, and Bellman equations to the new framework. Then, it provides a taxonomy for the Conf-MDPs and it concludes with a survey of the frameworks and approaches that share some similarities with the Conf-MDPs. Some parts of the chapter, especially the definition of Conf-MDP and a portion of the literature review, appeared in a preliminary version in Metelli et al. (2018a), whereas the remaining part is a novel contribution of this dissertation.

- Chapter 5 is devoted to the presentation of the solution concepts for Conf-MDPs. The chapter is conceptually organized in two sections that correspond to the cooperative and non-cooperative settings respectively. For both of them, we introduce the optimality conditions, present the value functions, Bellman operators, and Bellman equations and we discuss, whenever possible, the existence of optimal policies

Paper	Reference	Paper link	Code link	Contribution statement
Alberto Maria Metelli, Mirco Mutti, and Marcello Restelli *Configurable Markov Decision Processes* In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, pages 3488–3497.	Metelli et al. (2018a)	proceedings.mlr.press/v80/metelli18a.html	github.com/albertometelli/Configurable-Markov-Decision-Processes-ICML-2018	A.M.M. devised the definition of Conf-MDP, derived the SPMI algorithm, and its theoretical analysis.
Alberto Maria Metelli, Emanuele Ghelfi, and Marcello Restelli *Reinforcement Learning in Configurable Continuous Environments* In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, pages 4546–4555.	Metelli et al. (2019a)	proceedings.mlr.press/v97/metelli19a.html	github.com/albertometelli/remps	A.M.M. devised the REMPS algorithm, its theoretical analysis, and performed the experiments on the illustrative chain domain.
Alberto Maria Metelli, Guglielmo Manneschi, and Marcello Restelli *Policy Space Identification in Configurable Environments* CoRR, abs/1909.03984, 2019c. Under review for Machine Learning Springer.	Metelli et al. (2019c)	arxiv.org/abs/1909.03984	github.com/albertometelli/policy-space-identification	A.M.M. devised the identification rules, their analysis, and performed the experiments in the Conf-MDP and IL settings.
Alberto Maria Metelli, Flavio Mazzolini, Lorenzo Bisi, Luca Sabbioni, and Marcello Restelli *Control Frequency Adaptation via Action Persistence in Batch Reinforcement Learning.* In Proceedings of the 37th International Conference on Machine Learning, ICML 2020	Metelli et al. (2020a)	proceedings.mlr.press/v119/metelli20a.html	github.com/albertometelli/pfqi	A.M.M. derived the analysis of the performance loss, the error propagation result for PFQI, the persistence selection method, and performed part of the experiments.

Table 1.1: *Summary of the papers whose content is reported in this dissertation, including the link to the paper, the link to the code, and the contributions of the author of this dissertation.*

and environment configurations. Besides the optimality condition for the cooperative setting that was already presented in Metelli et al. (2018a), the content of this chapter is a contribution of the dissertation.

Part II: Learning in Cooperative Configurable Markov Decision Processes

This part is devoted to the study of the learning problem in cooperative Conf-MDPs, i.e., the problem of learning an optimal policy together with an optimal environment configuration. The content of this part is organized in two chapters.

- Chapter 6 focuses on the learning problem in finite Conf-MDPs. The content of the chapter is derived from Metelli et al. (2018a), although in this dissertation we provide an improved derivation of the learning algorithm.
- Chapter 7 addresses the learning problem in continuous Conf-MDPs as well as the problem of estimating the effects of the configurable parameters on the environment dynamics. The results presented in this chapter appeared in Metelli et al. (2019a).

Part III: Applications of Configurable Markov Decision Processes

This part addresses two applications in which the Conf-MDPs play an important role: policy space identification and control frequency adaptation. Specifically, the material of this part is organized in two chapters.

- Chapter 8 analyzes the problem of the identification of the policy space accessible to a learning agent, by observing its behavior in a configurable environment. A preliminary version of the content of this chapter appeared in the preprint Metelli et al. (2019c), but we include a more detailed comparison with the existing work as well as additional experiments in the imitation learning setting.
- Chapter 9 studies the problem of adapting the control frequency of a system, an environmental parameter that can be externally configured. We analyze the effect of changing the control frequency on the agent's performance and we apply this finding in the batch RL setting. The content of this chapter is derived from Metelli et al. (2020a).
- Chapter 10 revises the contributions of the dissertation, pointing out the main limitations of the present work, and proposing directions for future research.

In Appendix A, we report some additional results and the proofs we omit in the main text of the dissertation. In Appendix B, we present some properties of the policies belonging to the exponential family.

CHAPTER 2

Foundations of Sequential Decision-Making

2.1 Introduction

In Chapter 1, we have introduced informally the main elements at the basis of any RL problem: an artificial agent interacts with an environment by performing actions and sensing observations. The agent's learning process is guided by the reward signal and the agent's goal consists of finding a prescription of actions so as to maximize the long-term reward. The mathematical tool used to model this kind of interaction is the Markov Decision Processes formalism (MDP, Puterman, 2014). This chapter is devoted to the presentation of the fundamental elements of the sequential decision-making problems that will be employed in the subsequent chapters of the dissertation. For an extensive review of the numerous aspects of RL, we refer the reader to the Sutton and Barto's book (Sutton and Barto, 2018) and to the monographs (Szepesvári, 2010; Agarwal et al., 2019).

RL is intimately different, and arguably more challenging, than other machine learning paradigms, like supervised learning (Mitchell, 1997; Bishop, 2007). In (online) supervised learning, whenever a decision (e.g., a predicted label) is issued, immediate feedback is received and the goal is to optimize that feedback (e.g., minimize the classification error). The decision only determines the immediate feedback and it does not influence the subsequent ones or the corresponding feedback (Cesa-Bianchi and Lugosi, 2006). Instead, in the typical RL setting the effect of an action determines not only the immediate reward, but it affects the distribution of the subsequent states. Thus, RL deals with *sequential decision-making* problems. As a consequence, to learn an optimal action prescription (a *policy* in the RL terminology) an agent has to *plan*, being aware that the chosen actions might re-

alize their relevant consequences in the future. Indeed, since the ultimate goal consists in optimizing the long-term reward, it might be convenient to sacrifice some immediate reward because this choice will lead, in the future, to more profitable states (Sutton and Barto, 2018).

Any approach addressing the RL problem cannot disregard the way the environment evolves as an effect of the actions the agent plays, i.e., the *environment dynamics*. When the environment dynamics (and also the immediate reward) is known to the agent, finding an optimal policy can be addressed by means of *Dynamic Programming* (DP, Bellman, 1957). However, in most of the scenarios of interest, the environment dynamics is either unknown or captured by complex models (e.g., fluid dynamic models) that are computationally expensive to employ in practice. For these reasons, the RL algorithms need to figure out the environment dynamics by either modeling it directly or learning its effects implicitly. In both cases, in order to perform a modification in its behavior, the agent has to collect sufficient information to understand the environment dynamics. Thus, the agent faces the well-known *exploration-exploitation dilemma*, that formalizes the coexistence of two conflicting propensities. On one hand, the agent should *explore* the environment to understand the effects of its actions. Intuitively, this suggests that the agent should visit every state and try every action indefinitely. However, on the other hand, to make the learning process converge to an optimal policy, exploration should be stopped, or progressively decreased, in favor of exploitation. Indeed, the agent needs to make use sooner or later, or *exploit*, the acquired knowledge to play what is believed to be an optimal action. When to stop exploration and begin exploitation or how to optimally mix the two phases is one of the most significant challenges of RL (Lattimore and Szepesvári, 2020).

Chapter Outline The chapter is organized as follows. In Section 2.2 we formalize the notion of Markov decision process and policy. Sections 2.3 and 2.4 are devoted to the presentation of the Markov reward process and Markov chains obtained by paring an MDP with a policy. Then, in Section 2.5, we introduce the performance indexes used to formalize the intuitive notion of long-term reward. Section 2.6 is dedicated to the value functions, the corresponding Bellman equations and operators for the discounted setting. In Section 2.7, we present the optimality conditions for the discounted setting, along with the optimal value functions and operators. We conclude in Section 2.8 with a brief presentation of the methods to solve discounted MDPs when the environment dynamics and the reward are known.

2.2 Markov Decision Processes

The interaction between an agent and an environment, in a sequential decision-making problem, is typically modeled by means of the Markov Decision Process (MDP, Puterman, 2014) formalism. We restrict our attention to the case of *discrete-time infinite-horizon* MDPs (Puterman, 2014), in which the time line is modeled as a discrete set of time instants, called *decision (time) steps*.[1] In each decision step, the agent perceives the state of the environment and it is required to perform an action. As an effect of the action, the environment evolves, according to its dynamics, into a new state and provides the agent with a real feedback, the immediate reward. The goal of the agent is to execute a prescription

[1] We will mention *continuous-time* MDPs (Doya, 1995) in Chapter 9.

of actions, called policy, so as to maximize a notion of long-term reward, that encodes the sequential nature of the task.

The RL literature has extensively studied the MDP framework (e.g., Bertsekas and Tsitsiklis, 1996; Bertsekas, 2005). The terminology was introduced for the first time in Bellman (1954) and the model components defined in Bellman (1957). Subsequently, a number of diverse formalizations have been proposed (e.g., Dubins et al., 2014; Blackwell, 1965). We present the following definition that trade-offs generality and accessibility.

Definition 2.1 (Markov Decision Process)**.** *A discrete-time infinite-horizon discounted Markov Decision Process (MDP) is a 6-tuple* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mu_0, R, \gamma)$ *where:*

- $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$ *is a non-empty measurable space called* state space*;*
- $(\mathcal{A}, \mathfrak{F}_{\mathcal{A}})$ *is a non-empty measurable space called* action space*;*
- $P : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S})$ *is the* transition model*, that for every state-action pair* $(s, a) \in \mathcal{S} \times \mathcal{A}$ *assigns a probability measure* $P(\cdot|s, a)$ *over the measurable space* $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$*;*
- $\mu_0 \in \mathscr{P}(\mathcal{S})$ *is the* initial state distribution*, that assigns probability measure over the measurable space* $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$*;*
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathscr{P}(\mathbb{R})$ *is the* reward model*, that for every state-action-state triple* $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ *assigns a probability measure* $R(\cdot|s, a, s')$ *over the measurable space* $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$*;*
- $\gamma \in [0, 1]$ *is the* discount factor.

For a complete and formal review of the MDP models refer to the distinguished Puterman's book (Puterman, 2014). In the following, we describe the components of our MDP definition.

State and Action Spaces The perception and actuation capabilities of the agent are modeled by means of the state space $\mathcal{S}$ and the action space $\mathcal{A}$ respectively, that can be either finite, countable infinite, or continuous.[2]

Environment Dynamics The dynamics of the environment is encoded in the transition model $P(\cdot|s, a)$ that for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ provides the probability distribution of the next state $s' \in \mathcal{S}$ when playing action a in state s. Unlike several authors (e.g., Puterman, 2014), we included in Definition 2.1, the initial state distribution μ_0 that provides the probability distribution of the initial state, i.e., the state at which the process is initialized. Whenever necessary, we will assume that $P(\cdot|s, a)$ and μ_0 admit a probability density function w.r.t. to the Lebesgue measure denoted with $p(s'|s, a)$, $\mu_0(s)$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ respectively. It is worth noting that the environment dynamics fulfills the *Markov property*, i.e., the distribution of the next state s' is a function of the current state s and action a only and it is independent of the past.[3]

[2] In Definition 2.1, we did not specify explicitly the σ-algebras $\mathfrak{F}_{\mathcal{S}}$ and $\mathfrak{F}_{\mathcal{A}}$. If $\mathcal{S}$ is finite or countable infinite we can choose the power set as σ-algebra, i.e., $\mathfrak{F}_{\mathcal{S}} = 2^{\mathcal{S}}$. If instead $\mathcal{S}$ is a topological space, like $\mathbb{R}^d$, we can resort to the Borel σ-algebra, i.e., $\mathfrak{F}_{\mathcal{S}} = \mathfrak{B}(\mathcal{S})$. Analogous considerations hold for $\mathfrak{F}_{\mathcal{A}}$. Sometimes we need a σ-algebra defined over the state-action space $\mathfrak{F}_{\mathcal{S}\times\mathcal{A}}$. In such a case, we can use the tensor-product σ-algebra

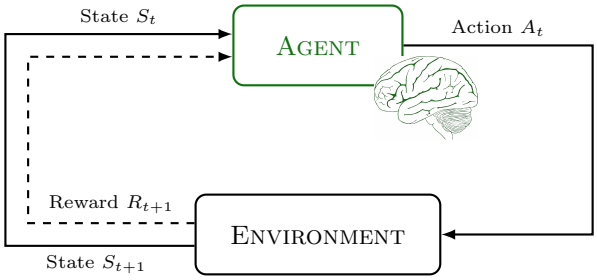


Figure 2.1: *Graphical representation of the interaction between an agent and an environment.*

Reward Function The reward generation process is governed by the reward model $R(\cdot|s, a, s')$ that provides the probability distribution of the reward when playing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and landing to state $s' \in \mathcal{S}$. We define the *reward function* $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ as the expected reward received when performing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and landing to stats $s' \in \mathcal{S}$:

$$r(s, a, s') = \int_{\mathbb{R}} rR(\mathrm{d}r|s, a, s').$$

Sometimes it is convenient to define the reward function by computing an expectation over the next state too.[4] The (next-state) expected reward function $r^P : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$r^P(s, a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s, a)r(s, a, s').$$

With negligible overloading of notation, we remove the superscript P, writing simply $r(s, a)$. A typical assumption that is widely employed in the RL literature is that the reward function is uniformly bounded.

Assumption 2.1 (Uniformly Bounded Reward)**.** *The reward function is uniformly bounded, i.e., there exists a finite constant $R_{\max} \in \mathbb{R}_{>0}$ such that:*

$$\|r\|_\infty = \sup_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} \{|r(s, a, s')|\} \leqslant R_{\max}.$$

Interaction The interaction between the agent and the environment starts at decision step $t = 0$ from state $S_0 \sim \mu_0$, i.e., sampled from the initial state distribution μ_0. For each decision step $t \in \mathbb{N}$, the agent selects an action $A_t \in \mathcal{A}$ that is executed in the environment.

$\mathfrak{F}_{\mathcal{S}\times\mathcal{A}} = \mathfrak{F}_{\mathcal{S}} \otimes \mathfrak{F}_{\mathfrak{F}}$.

[3] More formally, a discrete-time stochastic process $(X_t)_{t\in\mathbb{N}}$, defined on a measurable space $(\mathcal{X}, \mathfrak{F}_{\mathcal{X}})$ and adapted to the filtration $(\mathfrak{F}_t)_{t\in\mathbb{N}}$ satisfies the Markov property if for every $t \in \mathbb{N}_{\geqslant 1}$ and for every measurable set $\mathcal{U} \in \mathfrak{F}_{\mathcal{X}}$ it holds that (Durrett, 2010) $\mathbb{P}[X_t \in \mathcal{U}|\mathfrak{F}_t] = \mathbb{P}[X_t \in \mathcal{U}|X_{t-1}]$.

[4] For the MDPs, this simplification is w.l.o.g. for all the performance indexes since the transition model P is fixed. We will see that in Conf-MDPs, in which P can be modified, considering $r(s, a)$ instead of $r(s, a, s')$ may lead to trivial solutions.

As a result of the action execution, the environment transitions to the next state according to the transition model $S_{t+1} \sim P(\cdot|S_t, A_t)$ and provides the agent with the immediate reward generated by the reward model $R_{t+1} \sim R(\cdot|S_t, A_t, S_{t+1})$ and then the process is repeated. We restrict our attention to infinite-horizon MDPs, i.e., we assume that this interaction continues indefinitely. A state $s \in \mathcal{S}$ is called *terminal* (or *absorbing*) if no other states can be reached from s and all actions provide zero reward, i.e., $P(\cdot|s, a) = \delta_s$ and $R(\cdot|s, a, s) = \delta_0$ for every $a \in \mathcal{A}$. An MDP that contains a terminal state that is reachable with non-zero probability from any state is called *episodic*. Figure 2.1 illustrates the interaction for a single decision step.

Histories This interactive process generates a sequence of states, actions, and rewards. We define a *state-ending history* of length $T \in \mathbb{N}$ as a sequence of T state-action-reward triples followed by one state:

$$h = (s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_T, s_T) \in \mathcal{H}_{\mathcal{S},T},$$

where $\mathcal{H}_{\mathcal{S},T} = (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^T \times \mathcal{S}$ is the set of all state-ending trajectories of length T. Similarly, we define an *action-ending history* of length $T \in \mathbb{N}$ as a sequence of T state-action-reward triples followed by one state-action pair:

$$\tau = (s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T, a_T) \in \mathcal{H}_{\mathcal{A},T},$$

where $\mathcal{H}_{\mathcal{A},T} = (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^T \times \mathcal{S} \times \mathcal{A}$ is the set of all action-ending trajectories of length T.[5]

Trajectories We can push the definition of history to infinity by introducing an *(infinite-length) trajectory* as an infinite sequence of state-action-reward triples:

$$\tau = (s_0, a_0, r_1, s_1, a_1, r_2 \dots) = (s_t, a_t, r_{t+1})_{t \in \mathbb{N}} \in \mathcal{T},$$

where $\mathcal{T} = (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^{\mathbb{N}}$ is the set of all infinite-length trajectories. Given a discount factor $\gamma \in [0, 1]$ we define the *return function* $G_\gamma : \mathcal{T} \to \mathbb{R}$ for every trajectory $\tau = (s_t, a_t, r_{t+1})_{t \in \mathbb{N}}$ as the discounted sum of the rewards collected in the execution of τ:

$$G_\gamma(\tau) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}.$$

Given a trajectory $\tau \in \mathcal{T}$, for every $t_1, t_2 \in \mathbb{N}$ with $t_1 < t_2$ we denote with $\tau_{t_1:t_2} = (s_{t_1}, a_{t_1}, r_t, \dots s_{t_2-1}, a_{t_2-1}, r_{t_2})$ the *subtrajectory* delimited by the time indexes t_1 (included) and t_2 (excluded). In practice, *finite-length trajectories* are usually considered. In such a case, we denote with $T(\tau)$ the trajectory length.

We defer the discussion of the role of the discount factor γ in Section 2.5, after having introduced the performance indexes. We now focus on the notion of policy.

[5] Clearly, for every $T \in \mathbb{N}$ we have that $\mathcal{H}_{\mathcal{A},T} = \mathcal{H}_{\mathcal{S},T} \times \mathcal{A}$.

2.2.1 Policies

The role of the agent in an MDP consists in playing actions. The *policy* is the mathematical formalization of the strategy the agent employs to select the action to be played at each decision step, based on the history of previous observations. We start with the following general definition of a *history-dependent* policy and then we show the most significant particular cases.

Definition 2.2 (History-dependent Policy)**.** *A* history-dependent policy *is a sequence* $\pi = (\pi_t)_{t\in\mathbb{N}}$ *of functions* $\pi_t : \mathcal{H}_{\mathcal{S},t} \to \mathscr{P}(\mathcal{A})$ *that for every decision step* $t \in \mathbb{N}$ *and for every state-ending history* $h_t \in \mathcal{H}_{\mathcal{S},t}$ *of length* t *provide a probability measure* $\pi_t(\cdot|h_t)$ *over the measurable space* $(\mathcal{A}, \mathfrak{F}_{\mathcal{A}})$*. We denote with* Π^{HR} *the set of history-dependent policies.*

In this definition the distribution of the action is a function of the whole history $h_t = (s_0, a_0, r_1, \ldots, s_{t-1}, a_{t-1}, r_t, s_t)$ up to time $t \in \mathbb{N}$ and, possibly, explicitly depends on t. If the probability distribution of the action depends on the last state only, i.e., if for every pair of histories $h_t, h'_t \in \mathcal{H}_{\mathcal{S},t}$ having the same $s_t \in \mathcal{S}$ as the last state, we have that $\pi_t(\cdot|h_t) = \pi_t(\cdot|h'_t)$ a.s. for all $t \in \mathbb{N}$, we say that the policy is *Markovian*. In such a case, we abbreviate with $\pi_t(\cdot|s_t)$. Furthermore, if the policy does not depend explicitly on the decision step t, i.e., if $\pi_t(\cdot|s) = \pi_{t'}(\cdot|s)$ a.s. for all $t, t' \in \mathbb{N}$ and $s \in \mathcal{S}$, then we call it *stationary*. In such a case, we remove the subscript, simply writing $\pi(\cdot|s)$. We denote with Π^{SR} the set of Markovian stationary policies. We assume that $\pi(\cdot|s)$ admits a probability density function w.r.t. the Lebesgue measure, that overloading the notation, we denote with the same symbol $\pi(a|s)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. If for each state $s \in \mathcal{S}$ the policy provides probability to a single action (i.e., it is a Dirac delta measure), then we call it *deterministic*. In such a case, with little abuse of notation, we write $\pi : \mathcal{S} \to \mathcal{A}$, i.e., a function mapping states to actions, where $\pi(s)$ the action prescribed in state $s \in \mathcal{S}$. We denote with $\Pi^{\mathrm{SD}} = \mathcal{A}^{\mathcal{S}}$ the set of Markovian stationary deterministic policies. Whenever not differently specified, we will use term "policy" to denote a Markovian stationary policy.

2.3 Markov Reward Processes

An MDP $\mathcal{M}$ coupled with a policy $\pi \in \Pi^{\mathrm{SR}}$, induces a *Markov Reward Process* (MRP, Puterman, 2014) that is formalized by the 5-tuple $(\mathcal{S}, P^\pi, \mu_0, R^\pi, \gamma)$, where $P^\pi : \mathcal{S} \to \mathscr{P}(\mathcal{S})$ is the *state transition kernel* that for every state $s \in \mathcal{S}$ assigns a probability measure over $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$, defined for every $s' \in \mathcal{S}$ as:

$$P^\pi(\mathrm{d}s'|s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) P(\mathrm{d}s'|s, a). \tag{2.1}$$

Thus, $P^\pi(\cdot|s)$ represents the probability distribution of the next state $s' \in \mathcal{S}$ obtained by executing policy π in state s. Similarly, $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathscr{P}(\mathbb{R})$ provides for each state pair $(s, s') \in \mathcal{S} \times \mathcal{S}$ a probability measure over $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$, defined for every real number $r \in \mathbb{R}$ as:

$$R^\pi(\mathrm{d}r|s, s') = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) R(\mathrm{d}r|s, a, s').$$

Therefore, $R^\pi(\cdot|s,s')$ corresponds to the probability distribution of the reward obtained when starting from state s, executing an action according to π and landing to the next state s'. Moreover, we can define the state-next state reward function $r^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ and the state reward function $r^\pi : \mathcal{S} \to \mathbb{R}$, defined for every $s, s' \in \mathcal{S}$ as:

$$r^\pi(s,s') = \int_{\mathbb{R}} r R^\pi(\mathrm{d}r|s,s') = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) r(s,a,s'),$$
$$r^\pi(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) r(s,a).$$

From a control theory point of view, the MDP can be seen as a suitable formalization for an *uncontrolled* system, in which control is exercised externally by mean of the policy π. Instead, the MRP can be interpreted as a model for a *controlled* system in which the control intervention is already incorporated. Both models have in common the fact that, at each decision step, they output the immediate reward.

2.4 Markov Chains

Given an MRP, if we ignore the reward generation process, we obtain a *Markov Chain* (MC, Meyn and Tweedie, 1993), or Markov process. The (state) MC induced by policy $\pi \in \Pi^{\mathrm{SR}}$ in the MDP $\mathcal{M}$ is defined by the pair $(\mathcal{S}, P^\pi)$, where P^π is the state transition kernel, as defined in Equation (2.1). Thus, it describes the evolution of the state over time, when executing policy π in MDP $\mathcal{M}$. Sometimes it is useful to take a different point of view, focusing on the evolution of the state-action pairs over time. In such a case, we introduce the (state-action) MC induced by policy $\pi \in \Pi^{\mathrm{SR}}$ in the MDP $\mathcal{M}$ is defined by the pair $(\mathcal{S} \times \mathcal{A}, P^\pi)$. With little overloading of notation, we denote here with $P^\pi : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S} \times \mathcal{A})$ the *state-action transition kernel* that for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ provides a probability measure over $(\mathcal{S} \times \mathcal{A}, \mathfrak{F}_{\mathcal{S}\times\mathcal{A}})$, defined for every state-action pair $(s',a') \in \mathcal{S} \times \mathcal{A}$ as:

$$P^\pi(\mathrm{d}s', \mathrm{d}a'|s,a) = P(\mathrm{d}s'|s,a)\pi(\mathrm{d}a'|s).$$

Thus, P^π encodes the probability distribution of the next-state-next-action pair $(s',a') \in \mathcal{S} \times \mathcal{A}$ obtained starting from state s, playing action a, choosing the next state according to P, and selecting the next action according to π.

2.4.1 t-step Transition Kernels and Distributions

In both Markov chains introduced above, it is useful to define the state/state-action distributions after $t \in \mathbb{N}$ steps of interaction. More formally, for any $t \in \mathbb{N}_{\geqslant 1}$, the t-step state transition kernel $(P^\pi)^t : \mathcal{S} \to \mathscr{P}(\mathcal{S})$ and the t-step state-action transition kernel $(P^\pi)^t : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S} \times \mathcal{A})$ are recursively defined for every $(s,a), (s',a') \in \mathcal{S} \times \mathcal{A}$ as:

$$(P^\pi)^t(\mathrm{d}s'|s) = \int_{\mathcal{S}} (P^\pi)^{t-1}(\mathrm{d}s''|s) P^\pi(\mathrm{d}s'|s''),$$
$$(P^\pi)^t(\mathrm{d}s', \mathrm{d}a'|s,a) = \int_{\mathcal{S}\times\mathcal{A}} (P^\pi)^{t-1}(\mathrm{d}s'', \mathrm{d}a''|s,a) P^\pi(\mathrm{d}s', \mathrm{d}a'|s'', a''),$$

convening that $(P^\pi)^0(\mathrm{d}s'|s) = \delta_s(\mathrm{d}s')$ and $(P^\pi)^0(\mathrm{d}s', \mathrm{d}a'|s, a) = \delta_{(s,a)}(\mathrm{d}s', \mathrm{d}a')$. In operator form, the recursive definition is particularly clear and concise: $(P^\pi)^t = (P^\pi)^{t-1} P^\pi$.

Furthermore, given an initial state distribution $\mu_0 \in \mathscr{P}(\mathcal{S})$ we can introduce, for every $t \in \mathbb{N}$, the t-step state distribution $\mu^\pi_{\mu_0,t} \in \mathscr{P}(\mathcal{S})$, defined for every state $s \in \mathcal{S}$ as:

$$\mu^\pi_{\mu_0,t}(\mathrm{d}s) = \int_{\mathcal{S}} \mu_0(\mathrm{d}s')\,(P^\pi)^t(\mathrm{d}s|s').$$

To lighten the notation, we omit the subscript of the initial state distribution μ_0, whenever clear from the context, simply writing μ^π_t. Given the recursive nature of $(P^\pi)^t$, we can immediately recover the recursive relation for the t-step distribution: $\mu^\pi_t = \mu_0(P^\pi)^t = \mu^\pi_{t-1}P^\pi$. Clearly, we can also define the t-step state-action distribution $\mu^\pi_t \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$, defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$\mu^\pi_t(\mathrm{d}s, \mathrm{d}a) = \mu^\pi_t(\mathrm{d}s)\pi(\mathrm{d}a|s).$$

2.4.2 Stationary Distributions

Sometimes we are interested in looking at the distribution over all decision steps $t \in \mathbb{N}$ at once. In such a case, we need to combine the t-step distributions in an effective way. We consider the following general definition that averages the $(\mu^\pi_t)_{t\in\mathbb{N}}$ in an exponential way and then we provide the corresponding interpretation (Sutton et al., 1999a).

Definition 2.3 (γ-discounted Stationary Distribution)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a policy. The* state γ-discounted stationary distribution $\mu^\pi_{\mu_0,\gamma} \in \mathscr{P}(\mathcal{S})$ *is defined as the probability measure solution (if it exists) of the equation defined for every state $s \in \mathcal{S}$ as:*

$$\mu^\pi_{\mu_0,\gamma}(\mathrm{d}s) = (1-\gamma)\mu_0(\mathrm{d}s) + \gamma \int_{\mathcal{S}} \mu^\pi_{\mu_0,\gamma}(\mathrm{d}s')P^\pi(\mathrm{d}s|s'),$$

Similarly to the previous section, we will omit the dependence on the initial state distribution μ_0 when not generating confusion, simply writing μ^π_γ. When the discount factor $\gamma < 1$, we are guaranteed that the γ-discounted stationary distribution exists uniquely. Indeed, by using the Neumann series in operator form, we have:

$$\mu^\pi_\gamma = (1-\gamma)\mu_0\left(\mathrm{Id}_{\mathcal{S}} - \gamma P^\pi\right)^{-1} = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mu^\pi_t,$$

Whenever necessary, we assume that it admits probability density function w.r.t. Lebesgue measure, denoted with the same symbol. From an intuitive point of view, μ^π_γ is the normalized discounted sum of the visits to states, when playing policy π in MDP $\mathcal{M}$.

Instead, when $\gamma = 1$, Definition 2.3 reduces the *stationary distribution* recursively defined as $\mu^\pi = \mu^\pi P^\pi$. In finite Markov chains (when $\mathcal{S}$ is a finite set) the existence and uniqueness of a stationary distribution is ensured for irreducible chains (Serfozo, 2009). Moreover, if the chain is aperiodic the stationary distribution equals the *limiting distribution* (Serfozo, 2009): $\mu^\pi = \lim_{t\to\infty} \mu^\pi_t$. In other words, irreducible aperiodic Markov chains forget the initial state distribution μ_0. Additional technical conditions are necessary for infinite Markov chains (Asmussen, 2003).

Clearly, also for the γ-discounted stationary distributions, we can introduce the corresponding state-action version $\mu_\gamma^\pi \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ and state-action-next-state version $\mu_\gamma^\pi \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$, defined for every state-action-state triple $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ as:

$$\begin{aligned}\mu_\gamma^\pi(\mathrm{d}s, \mathrm{d}a) &= \mu_\gamma^\pi(\mathrm{d}s)\pi(\mathrm{d}a|s),\\ \mu_\gamma^\pi(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s') &= \mu_\gamma^\pi(\mathrm{d}s)\pi(\mathrm{d}a|s)P(\mathrm{d}s'|s, a).\end{aligned}$$

2.4.3 Trajectory Distributions

Given an MDP $\mathcal{M}$ and a policy $\pi \in \Pi^{\mathrm{SR}}$, we can characterize the distribution of the trajectories. Specifically, for every trajectory $\tau \in \mathcal{T}$ and $T \in \mathbb{N}$, we can express the probability measure $\mathbb{P}_T^\pi$ of the subtrajectories of length T induced by policy $\pi \in \Pi^{\mathrm{SR}}$ in MDP $\mathcal{M}$, defined for every $\tau_{0:T} = (s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$ as:

$$\mathbb{P}_T^\pi(\mathrm{d}\tau_{0:T}) = \mu_0(\mathrm{d}s_0) \prod_{t=0}^{T-1} \pi(\mathrm{d}a_t|s_t)P(\mathrm{d}s_{t+1}|s_t, a_t)R(\mathrm{d}r_{t+1}|s_t, a_t, s_{t+1}).$$

Clearly, the probability measure over subtrajectories of length $T+1$ can be easily defined recursively in terms of the probability measure over subtrajectories of length T:

$$\mathbb{P}_{T+1}^\pi(\mathrm{d}\tau_{0:T+1}) = \mathbb{P}_T^\pi(\mathrm{d}\tau_{0:T})\pi(\mathrm{d}a_T|s_T)P(\mathrm{d}s_{T+1}|s_T, a_T)R(\mathrm{d}r_{T+1}|s_T, a_T, s_{T+1}).$$

Finally, the probability measure $\mathbb{P}^\pi \in \mathscr{P}(\mathcal{T})$ over infinite-length trajectories is defined for every $\tau \in \mathcal{T}$ as the limit $\mathbb{P}^\pi(\mathrm{d}\tau) = \lim_{T\to\infty} \mathbb{P}_T^\pi(\mathrm{d}\tau_{0:T})$. Whenever necessary, we assume that these probability measures, $\mathbb{P}_T^\pi$ and $\mathbb{P}^\pi$, admit density function w.r.t. the Lebesgue measure, denoted with $\mathbb{p}_T^\pi$ for subtrajectories of length $T \in \mathbb{N}$ and $\mathbb{p}^\pi$ for infinite-length trajectories.

We employ the following abbreviated notation for the expectation of a bounded measurable function $f \in \mathscr{B}(\mathcal{T})$ taken w.r.t. infinite-length trajectories:

$$\mathbb{E}^\pi[f(\tau)] \coloneqq \mathop{\mathbb{E}}_{\tau\sim\mathbb{P}^\pi}[f(\tau)] = \int_{\mathcal{T}} \mathbb{P}^\pi(\mathrm{d}\tau) f(\tau).$$

Remark 2.1. *The stationary distribution μ_γ^π and the trajectory distribution $\mathbb{P}^\pi$ provide different views of the agent-environment interaction. While with the stationary distributions we focus on the states or state-action pairs, with the trajectory distributions we look at the whole trajectory, i.e., state-action-reward triples sequences. Both can be used to compute expectations of functions defined over $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathbb{R}$, as shown in the following result.*

Lemma 2.1 (D'Oro et al. (2020), Lemma A.2)**.** *Let $\mathcal{M}$ be an MDP, $\pi \in \Pi^{\mathrm{SR}}$ be a policy, and $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathbb{R})$ be a bounded measurable function. Then, it holds that:*

$$\mathop{\mathbb{E}}_{\substack{S,A,S'\sim\mu_\gamma^\pi \\ R\sim R(\cdot|S,A,S')}} \big[f(S, A, S', R)\big] = (1-\gamma)\mathbb{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t f(S_t, A_t, S_{t+1}, R_{t+1})\right].$$

This is particularly relevant when function $f(s, a, s', r) = r$ leading to the return function:

$$\mathop{\mathbb{E}}_{\substack{S,A,S' \sim \mu_\gamma^\pi \\ R \sim R(\cdot|S,A,S')}} [R] = (1-\gamma)\mathbb{E}^\pi \left[G_\gamma(\tau) \right],$$

where $\tau = (S_t, A_t, R_{t+1})_{t \in \mathbb{N}}$. Unfortunately, this equivalence holds for the expectation only and cannot be extended straightforwardly to higher-order moments (Bisi et al., 2020).

2.5 Performance Indexes

At the beginning of this chapter, we have stated informally that the goal of an agent in an MDP consists of finding a policy that maximizes some notion of long-term reward. In this section, we formalize it by presenting the main *performance indexes* employed in RL. Formally, a performance index is a mapping $(\mathcal{M}, \pi) \mapsto J_{\mathcal{M}}^\pi$ that, given an MDP $\mathcal{M}$ and a policy $\pi \in \Pi^{\text{HR}}$ provides a real number $J_{\mathcal{M}}^\pi \in \mathbb{R}$, i.e., the performance of policy π in MDP $\mathcal{M}$. When there is no confusion, we will drop the dependence of $\mathcal{M}$, abbreviating with J^π.

2.5.1 Expected Total Reward

The simplest formalization of the intuitive notion of long-term reward is the *expected total reward* J_{tot}^π. Given a policy $\pi \in \Pi^{\text{HR}}$, J_{tot}^π is defined as the expected sum of the rewards collected along an infinite-length trajectory, with no discounting:

$$J_{\text{tot}}^\pi = \lim_{T \to \infty} \mathbb{E}^\pi \left[\sum_{t=0}^{T-1} R_{t+1} \right] = \lim_{T \to \infty} \mathbb{E}^\pi \left[G_1(\tau_{0:T}) \right]. \tag{2.2}$$

Unfortunately, despite its intuitive definition, the expected total reward is often an ill-defined index. Indeed, the limit in Equation (2.2) might not exist or might be infinite, unless the MDP is episodic, i.e., it reaches an absorbing state almost surely.

2.5.2 Expected Total Discounted Reward or Expected Return

To overcome this limitation, the *expected total discounted reward*, also known as *expected return*, J^π is introduced. For a policy $\pi \in \Pi^{\text{HR}}$ and a discount factor $\gamma \in [0, 1]$, J^π is defined as the expected discounted sum of the rewards collected along an infinite-length trajectory:

$$J^\pi = \lim_{T \to \infty} \mathbb{E}^\pi \left[\sum_{t=0}^{T-1} \gamma^t R_{t+1} \right] = \lim_{T \to \infty} \mathbb{E}^\pi \left[G_\gamma(\tau_{0:T}) \right]. \tag{2.3}$$

Under the assumption on the boundedness of the reward (Assumption 2.1), the limit in Equation (2.3) exists finite and the expected total discounted return is always bounded by $|J^\pi| \leqslant \frac{R_{\max}}{1-\gamma}$. For this reason, we can exchange the limit with the expectation and rewrite its expression in the most common form: $J^\pi = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \right] = \mathbb{E}^\pi \left[G_\gamma(\tau) \right]$.

It is worth noting that the expected total discounted reward can be alternatively defined by means of the γ-discounted stationary distribution, as in Definition 2.3 (Sutton et al., 1999a):

$$J^{\pi} = \frac{1}{1-\gamma} \int_{\mathcal{S}} \mu_{\gamma}^{\pi}(\mathrm{d}s, \mathrm{d}a) r(s,a).$$

The Role of the Discount Factor It is immediate to realize the mathematical advantage of employing a discount factor $\gamma < 1$ in avoiding the divergence of the series in Equation (2.3). However, the discount factor admits other interpretations. From an economical point of view, the value of γ modules the interest the agent demonstrates in gaining reward in the future. A small discount factor, i.e., $\gamma \simeq 0$, is associated to a *myopic* attitude since the agent is more interested in obtaining reward in the present or near future. The extreme case $\gamma = 0$ reduces RL to supervised learning since the only interest of the agent is maximizing the immediate reward. Instead, large values of gamma, i.e., $\gamma \simeq 1$, model *far-sighted* agents that are willing to sacrifice immediate reward because they give importance even to far-future rewards. Finally, the discount factor can be interpreted also from a statistical perspective. Indeed, for every infinite-horizon discounted MDP $\mathcal{M}$ it is possible to define an episodic undiscounted MDP $\widetilde{\mathcal{M}}$ equivalent $\mathcal{M}$. In every state of $\widetilde{\mathcal{M}}$ there is a probability $1-\gamma$ to reach a zero-reward absorbing state, whichever action is played (Puterman, 2014). It is simple to prove that for any policy $\pi \in \Pi^{\mathrm{SR}}$, the expected total reward of π in the new MDP $\widetilde{\mathcal{M}}$ equals the expected total discounted reward of π in the original MDP $\mathcal{M}$. Thus, we can interpret γ as the probability that the interaction with the environment continues for another time step. It is worth noting that the length of a trajectory in $\widetilde{\mathcal{M}}$ is a geometric distribution of parameter $1-\gamma$. Thus, the expected length is $\frac{1}{1-\gamma}$, that is often referred as the *effective horizon* of the original MDP $\mathcal{M}$.

2.5.3 Average Reward

Finally, we present one last performance index that is employed in the literature: the *average reward* J_{avg}^{π}. Given a policy $\pi \in \Pi^{\mathrm{HR}}$, J_{avg}^{π} is defined as the expected average of the rewards collected along an infinite-length trajectory:

$$J_{\mathrm{avg}}^{\pi} = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}^{\pi} \left[\sum_{t=0}^{T-1} R_{t+1} \right] = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}^{\pi} \left[G_1(\tau_{0:T}) \right]. \tag{2.4}$$

Whenever the limit in Equation (2.4) does not exist, it is replaced with $\liminf$ or $\limsup$. The average reward, whenever it exists, under Assumption 2.1, is bounded by $|J_{\mathrm{avg}}^{\pi}| \leqslant R_{\max}$. If they both exist, J_{avg}^{π} can be expressed in terms of the stationary distribution (Sutton and Barto, 2018) as follows:

$$J_{\mathrm{avg}}^{\pi} = \int_{\mathcal{S}} \mu^{\pi}(\mathrm{d}s, \mathrm{d}a) r(s,a).$$

In the rest of the dissertation, we will mainly focus on the expected total discounted reward.

Finally, given a probability measure $\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ we define the expected return induced by μ as:

$$J^{\mu} = \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \mu(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s') r(s, a, s'). \tag{2.5}$$

2.6 Value Functions

A fundamental concept of RL is the *value function* (Sutton and Barto, 2018). Differently from the performance indexes presented in Section 2.5 that associate a single real number to each policy, the value functions provide an index that is defined in terms of the initial state choice $(\mathcal{M}, \pi, s) \mapsto V^{\pi}_{\mathcal{M}}(s)$ (or state-action pair choice $(\mathcal{M}, \pi, (s, a)) \mapsto Q^{\pi}_{\mathcal{M}}(s, a)$). Whenever clear from the context we will drop the dependence on the MDP $\mathcal{M}$. Value functions play a central role in value-based reinforcement learning since they allow deriving an optimal policy. We limit our presentation to the discounted case, i.e., $\gamma < 1$. We refer the reader to (Puterman, 2014) for the corresponding versions for the total and average rewards. Let us start by defining the *state value function*.

Definition 2.4 (State Value Function or V-function)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a policy. For every state $s \in \mathcal{S}$, the* state value function $V^{\pi} : \mathcal{S} \to \mathbb{R}$ *is defined as the expected return starting from state s and following policy π thereafter:*

$$V^{\pi}(s) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s\right].$$

The state value function finds its main application in *policy evaluation*, i.e., the process of computing the performance of a policy π in an MDP. However, V^{π} does not encompass enough information for *policy optimization*, i.e., the process of finding a policy with optimal performance without the knowledge of the transition model P. To this purpose, the *action value function* is introduced.

Definition 2.5 (State-Action Value Function or Q-function)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a policy. For every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the* state-action value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is defined as the expected return starting from state s, playing action a, and following policy π thereafter:*

$$Q^{\pi}(s, a) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s, A_0 = a\right].$$

Clearly, the V-function can be defined in terms of the Q-function by simply taking the expectation over the action space: $V^{\pi}(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) Q^{\pi}(s, a)$. In turn, the expected return, as defined in Equation (2.3), is the expectation of the V-function taken w.r.t. the choice of the initial state:

$$J^{\pi} = \int_{\mathcal{S}} \mu_0(\mathrm{d}s) V^{\pi}(s).$$

In some contexts, it is useful to define the *advantage function* $A^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defined for every $(s, a) \in \mathcal{S} \times \mathcal{A}$ as (Baird III, 1993):

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s), \tag{2.6}$$

that quantifies the performance gain obtained by playing action a in state s compared to executing policy $\pi(\cdot|s)$.

2.6.1 Bellman Equations and Operators

The definition of value function we provided above are trajectory-based, i.e., they are defined as the expected return collected along an infinite-length trajectory. When we focus on infinite-horizon MDPs and Markovian stationary polices, value function can also be expressed in a *recursive* form. The technical tools employed to obtain these relations are the *Bellman Equations* and the *Bellman Expectation Operators* (Bellman, 1957), that represent the concrete basis of many RL algorithms.

Definition 2.6 (Bellman Expectation Operators)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a policy. The* Bellman expectation operator *for the state value function $T^\pi : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S})$ and every state $s \in \mathcal{S}$ as:*

$$\left(T^\pi f\right)(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r(s,a,s') + \gamma f(s')\right). \tag{2.7}$$

The Bellman expectation operator *for the state-action value function $T^\pi : \mathscr{B}(\mathcal{S}\times\mathcal{A}) \to \mathscr{B}(\mathcal{S}\times\mathcal{A})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S}\times\mathcal{A})$ and every state-action pair $(s,a) \in \mathcal{S}\times\mathcal{A}$ as:*

$$\left(T^\pi f\right)(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') f(s',a')\right). \tag{2.8}$$

It is worth noting that both T^π are *linear* operators, that fulfill the monotonicity property and for $\gamma < 1$ they are a *contraction* in L_∞-norm (Bertsekas and Tsitsiklis, 1996; Puterman, 2014), i.e., for appropriately defined bounded measurable functions f and g it holds that:[6]

$$\left\|T^\pi f - T^\pi g\right\|_\infty \leqslant \gamma \left\|f - g\right\|_\infty.$$

As a consequence, thanks to the Banach fixed-point theorem (Banach, 1922), the T^π admit a unique fixed-point that are, respectively, the state value function V^π and the state action value function Q^π (Puterman, 2014). The corresponding fixed-point equations are called *Bellman Expectation Equations*:

$$V^\pi = T^\pi V^\pi,$$
$$Q^\pi = T^\pi Q^\pi.$$

2.7 Optimality Criteria

In this section, we focus on how to define a notion of optimality in the discounted setting ($\gamma < 1$) and on how to compute the optimal value function and an optimal policy. Let us start with the following definition that introduces the notion of optimality for the policy and for the value function (Puterman, 2014).

[6] It is worth noting that by recalling the definition of $r(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) r(s,a,s')$ and $r^\pi(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) r(s,a)$, we can rephrase the Bellman expectation operators in operator form. For the V-function $T^\pi f = r^\pi + \gamma P^\pi f$ with $f \in \mathscr{B}(\mathcal{S})$ and for the Q-function $T^\pi f = r + \gamma P^\pi f$ with $f \in \mathscr{B}(\mathcal{S}\times\mathcal{A})$.

Definition 2.7 (Optimality)**.** *Let $\mathcal{M}$ be an MDP. A history-dependent policy $\pi^* \in \Pi^{\mathrm{HR}}$ is* optimal *if for every state $s \in \mathcal{S}$ and history-dependent policy $\pi \in \Pi^{\mathrm{HR}}$ it holds that:*

$$V^{\pi^*}(s) \geqslant V^{\pi}(s). \tag{2.9}$$

The optimal state value function *is defined for every state $s \in \mathcal{S}$ as:*

$$V^*(s) = \sup_{\pi \in \Pi^{\mathrm{HR}}} \{V^{\pi}(s)\}. \tag{2.10}$$

The definition makes use of history-dependent policies but we can freely restrict the search to the Markovian stationary policies Π^{SR} since, in the discounted setting, for every history-dependent policy $\pi \in \Pi^{\mathrm{HR}}$ there exists a Markovian stationary policy $\pi' \in \Pi^{\mathrm{SR}}$ such that $V^{\pi}(s) = V^{\pi'}(s)$ (Puterman, 2014, Theorem 5.5.3).

2.7.1 Optimal Value Functions

The optimal state value function represents the best possible performance attainable in an MDP starting from every state. Analogously it is possible to define the *optimal state-action value function* defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$Q^*(s, a) = \sup_{\pi \in \Pi^{\mathrm{HR}}} \{Q^{\pi}(s, a)\}. \tag{2.11}$$

Clearly, V^* and Q^* are related by the identity $V^*(s) = \sup_{a \in \mathcal{A}}\{Q^*(s, a)\}$ for every state $s \in \mathcal{S}$. We can restrict the maximization to the Markovian stationary policies Π^{SR} also in this case. Similarly to the value function presented in Section 2.6, the optimal value functions can be expressed in terms of suitable Bellman operators.

Definition 2.8 (Bellman Optimality Operators)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a policy. The* Bellman optimality operator *for the state value function $T^* : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S})$ and every state $s \in \mathcal{S}$ as:*

$$(T^* f)(s) = \sup_{a \in \mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s' | s, a) \left(r(s, a, s') + \gamma f(s') \right) \right\}. \tag{2.12}$$

The Bellman optimality operator *for the state-action value function $T^* : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathscr{B}(\mathcal{S} \times \mathcal{A})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:*

$$(T^* f)(s, a) = \int_{\mathcal{S}} P(\mathrm{d}s' | s, a) \left(r(s, a, s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ f(s', a') \right\} \right). \tag{2.13}$$

Compared to the Bellman expectation operators, introduced in Section 2.6.1, the Bellman optimality operators are no longer linear due to the presence of the supremum. Nevertheless, for $\gamma < 1$ they preserve the monotonicity and the contraction in L_∞-norm properties (Proposition 6.2.4, Puterman, 2014), i.e., for appropriately defined bounded measurable functions f and g it holds that:[7]

$$\|T^* f - T^* g\|_\infty \leqslant \gamma \|f - g\|_\infty.$$

[7] By introducing the maximum operator over the action space $\mathrm{M}_{\mathcal{A}} : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathscr{B}(\mathcal{S})$ defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ as: $(\mathrm{M}_{\mathcal{A}} f)(s) = \sup_{a \in \mathcal{A}}\{f(s, a)\}$, we can redefine the Bellman optimality operators in operator form. For the V-function $T^* f = \mathrm{M}_{\mathcal{A}}(r + \gamma P^{\pi} f)$ with $f \in \mathscr{B}(\mathcal{S})$ and for the Q-function $T^* f = r + \gamma P^{\pi} \mathrm{M}_{\mathcal{A}} f$ with $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$.

Furthermore, it is possible to prove that their unique fixed-points are, respectively, the state optimal value function V^* and the state-action optimal value function Q^* (Theorem 6.2.5, Puterman, 2014). The corresponding fixed-point equations are called *Bellman Optimality Equations*:

$$V^* = T^* V^*,$$
$$Q^* = T^* Q^*.$$

2.7.2 Greedy Policies

Before showing that under sufficiently general assumptions an optimal policy exists, we need to introduce the notion of greedy action and greedy policy.

Definition 2.9 (Greedy Actions and Policies)**.** *Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ be a bounded measurable function, for every state $s \in \mathcal{S}$ we say that an action $a^+ \in \mathcal{A}$ is greedy in state s if $f(s, a^+) = \sup_{a \in \mathcal{A}} \{f(s, a)\}$. A* greedy policy *w.r.t. a function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ is any policy $\pi^+ \in \Pi^{\mathrm{SR}}$ playing only greedy actions, i.e., for every state $s \in \mathcal{S}$ it holds that:*

$$\int_{\mathcal{A}} \pi^+(\mathrm{d}a|s) f(s, a) = \sup_{a \in \mathcal{A}} \{f(s, a)\}.$$

Consequently, if $\pi^+ \in \Pi^{\mathrm{SR}}$ is greedy w.r.t. to the function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$, the following identity involving the Bellman operators holds:

$$T^{\pi^+} f = T^* f.$$

2.7.3 Optimal Policies

The optimality condition in Definition 2.7 prescribes that a policy π^* for being optimal must yield a value function in each state $s \in \mathcal{S}$ at least as good as that of any other policy π, i.e., $V^{\pi^*}(s) \geqslant V^{\pi}(s)$. We start by defining the following *preorder* (or preference) relationship $\gtrsim$ on the space of Markovoian stationary policies Π^{SR}.

Definition 2.10 (Preorder on Π^{SR})**.** *Let $\mathcal{M}$ be an MDP. The preference relationship $\gtrsim \subseteq \Pi^{\mathrm{SR}} \times \Pi^{\mathrm{SR}}$ is defined for two policies $\pi, \pi' \in \Pi^{\mathrm{SR}}$ as:*

$$\pi \gtrsim \pi' \iff V^{\pi}(s) \geqslant V^{\pi'}(s), \quad \forall s \in \mathcal{S}. \tag{2.14}$$

The relationship $\gtrsim$ is clearly reflexive and transitive, but it is not antisymmetric (thus it is a preorder but not a partial order) since there might be policies that are different yielding the same value function. According to Definition 2.7, an optimal policy, if it exists, is a maximum according to the preference $\gtrsim$. A way to construct an optimal policy consists in deriving a greedy policy (Definition 2.9) w.r.t. the optimal Q-function, i.e., any policy $\pi^* \in \Pi^{\mathrm{SR}}$ such that for every state $s \in \mathcal{S}$:

$$\int_{\mathcal{A}} \pi^*(\mathrm{d}a|s) Q^*(s, a) = V^*(s). \tag{2.15}$$

The following result, that we report without proof, shows that under suitable conditions such a policy exists and it is optimal.

Theorem 2.2 (Theorem 6.2.7, Puterman (2014))**.** *Let $\mathcal{M}$ be an MDP. If the state space $\mathcal{S}$ is discrete and the supremum $V^*(s) = \sup_{a \in \mathcal{A}}\{Q^*(s,a)\}$ is attained for every state $s \in \mathcal{S}$, then:*

(a) there exists a Markovian stationary greedy policy $\pi^* \in \Pi^{\mathrm{SR}}$ w.r.t. to Q^*;

(b) π^* is an optimal policy, i.e., $\pi^* \gtrsim \pi$ for every policy $\pi \in \Pi^{\mathrm{SR}}$;

(c) there exists a deterministic Markovian stationary optimal policy.

Let us discuss more in detail the meaning of Theorem 2.2. The statement requires that the supremum is attained, i.e., for every state $s \in \mathcal{S}$ there must exist an action a^+ such that $Q^*(s,a^+) = V^*(s)$. If this is the case, a greedy policy π^* is well-defined as any policy that plays actions belonging to the set $\arg\max_{a \in \mathcal{A}}\{Q^*(s,a)\}$. The main statement of Theorem 2.2 is point (b) showing that such a greedy policy is an optimal policy in the sense of Definition 2.7. Clearly, since a greedy policy exists, it follows that a deterministic greedy policy exists too and, consequently, a deterministic optimal policy exists. Finally, note that (but this was already evident in Definition 2.7) that all optimal policies attain the optimal value function, i.e., $V^{\pi^*}(s) = V^*(s)$ for all states $s \in \mathcal{S}$. Unfortunately, the result holds only when the state space $\mathcal{S}$ is discrete. When this is not the case, even if the supremum is attained, an optimal policy might not exist (Blackwell, 1965). The discussion of the conditions under which the existence of an optimal policy (or an ϵ-optimal policy) is ensured is out of the scope of this dissertation. We refer the interested reader to Bertsekas and Shreve (2004); Dynkin et al. (1979) for more details. In the following, whenever necessary we will assume the existence of an optimal policy that can be expressed as a greedy policy w.r.t. Q^*.

In practical applications, the condition requiring that the policy maximizes the value function in all the states $s \in \mathcal{S}$ is often too demanding, especially when the search is carried out in a subset of Π^{SR}. For this reason, more relaxed definitions of optimality have been proposed, like the following.

Definition 2.11 (J-optimality)**.** *Let $\mathcal{M}$ be an MDP and let J be a performance index. A policy $\pi^* \in \Pi^{\mathrm{SR}}$ is J-optimal if for every policy $\pi \in \Pi^{\mathrm{SR}}$: $J^{\pi^*} \geqslant J^{\pi}$.*

Of course, since we are evaluating each policy by means of a scalar function, Definition 2.11 induces a complete preorder relation $\gtrsim_J$ on Π^{SR}. Typical choices for J are the performance indexes presented in Section 2.5. When we employ the expected return J^{π} then we can relate the notion J^{π}-optimality with the original notion of optimality. Indeed, any optimal policy according to Definition 2.7 is also optimal according to Definition 2.11, but, clearly, not vice versa.

2.8 Exact Solution Methods

In this section, we focus on the problem of finding an optimal policy, in the sense of Definition 2.7, when considering a finite MDP. We consider the full knowledge of the elements of the MDP, i.e., the transition model P and the reward function r. The fundamental idea at the basis of these algorithms is to first compute the optimal value function and then recover an optimal policy as a greedy policy. Although the knowledge of the

Algorithm 2.1: Value iteration (VI).

Input: MDP $\mathcal{M}$, horizon T
Output: approximately optimal policy $\pi^{(T)}$
Initialize $V^{(0)}$ arbitrarily
forall $t=0,1,\dots,T-1$ **do**
  $V^{(t+1)}(s)=\max_{a\in\mathcal{A}}\Big\{r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)V^{(t)}(s)\Big\}, \quad \forall s\in\mathcal{S}$ ▷ *Bellman Operator*
$\pi^{(T)}(s)\in\arg\max_{a\in\mathcal{A}}\Big\{r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)V^{(T)}(s')\Big\}, \quad \forall s\in\mathcal{S}$ ▷ *Greedy policy*
return $\pi^{(T)}$

environment dynamics is an unrealistic requirement in most applications of interest, these algorithms are particularly relevant as they represent the building block of the value-based RL algorithms. Section 2.8.1 and 2.8.2 are devoted to the presentation of the two *dynamic programming* algorithms: Policy Iteration (PI, Howard, 1960) and Value Iteration (VI, Bellman, 1957). Then, in Section 2.8.3 we present the Linear Programming approach (LP, Wang et al., 2007).

2.8.1 Value Iteration

The value iteration algorithm (Bellman, 1957) is the most straightforward method to solve a finite MDP and is based on the iterative application of the Bellman optimality operator T^*, for a given number of iterations $T \in \mathbb{N}$ (also known as optimization horizon). At the end of the process, VI outputs a greedy policy $\pi^{(T)}$ w.r.t. to the T-approximation of the value function $V^{(T)}$. The pseudocode of VI is reported in Algorithm 2.1. Thanks to the contraction property of T^*, it is immediate to prove that the sequence of value functions $(V^{(t)})_{t=0}^{T}$ generated by VI converges in L_∞-norm to the optimal state value function. Indeed, for every iteration $t \in \mathbb{N}_{\geqslant 1}$, we have:

$$\left\|V^{(t)} - V^*\right\|_\infty \leqslant \gamma \left\|V^{(t-1)} - V^*\right\|_\infty,$$

leading to a *linear* convergence rate (Puterman, 2014). Since, at each iteration, VI requires computing the optimal action in each state, that requires $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$ operations for each state, the computational complexity of T iterations of VI is of $\mathcal{O}(T|\mathcal{S}|^2|\mathcal{A}|)$.

2.8.2 Policy Iteration

Policy iteration (Howard, 1960) solves a finite MDP by explicitly representing the intermediate policies, during the considered iterations. Specifically, PI is composed of two phases that are repeated in sequence: i) *policy evaluation*; ii) *policy improvement*. For every iteration $t \in \{0, \dots, T-1\}$, the policy evaluation phase, given the current policy $\pi^{(t)}$, consists in computing its value function $V^{\pi^{(t)}}$. This step can be carried out in several ways, for instance by performing a repeated application of the Bellman expectation operator T^π or solving the linear system of Bellman equations. In practice, it is not always necessary to wait convergence to $V^{\pi^{(t)}}$, but a smaller number of applications of T^π is

Algorithm 2.2: Policy iteration (PI).

Input: MDP $\mathcal{M}$, horizon T
Output: approximately optimal policy $\pi^{(T)}$
Initialize $\pi^{(0)}$ arbitrarily
forall $t=0,1,\dots,T-1$ **do**
  Solve $V^{\pi^{(t)}}(s)=\sum_{a\in\mathcal{A}}\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)V^{\pi^{(t)}}(s')\right), \forall s\in\mathcal{S}$ ▷ *Evaluation*
  $\pi^{(t+1)}(s)\in\operatorname{argmax}_{a\in\mathcal{A}}\left\{r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)V^{\pi^{(t)}}(s')\right\}, \quad \forall s\in\mathcal{S}$ ▷ *Improvement*
return $\pi^{(T)}$

sufficient. This variant is known as *modified policy iteration* (Puterman and Shin, 1978). The policy improvement step, instead, consists in computing a greedy policy $\pi^{(t+1)}$ w.r.t. to the current approximation of the value function $V^{\pi^{(t)}}$. The pseudocode of PI is reported in Algorithm 2.2. A remarkable property of the PI is that it is guaranteed to provide a sequence of policies with non-decreasing performance.

Theorem 2.3 (Policy Improvement Theorem (Sutton and Barto, 2018))**.** *Let $\mathcal{M}$ be an MDP and $\pi, \pi' \in \Pi^{\mathrm{SR}}$ be two policies. If for every state $s \in \mathcal{S}$ it holds that:*

$$\int_{\mathcal{A}} \pi'(\mathrm{d}a|s) Q^{\pi}(s,a) \geqslant V^{\pi}(s),$$

then it holds that for every state $s \in \mathcal{S}$:

$$V^{\pi'}(s) \geqslant V^{\pi}(s).$$

The theorem shows that if a policy π' improves the one-step performance of π for all the states, then it will yields a better value function overall. Clearly, the one-step improvement condition is fulfilled by the greedy policy. Whenever PI stops it means that we have reached an optimal policy. Since it iterates over the deterministic greedy policies, that are at most $|\mathcal{A}|^{|\mathcal{S}|}$, PI converges in a finite number of iterations to an optimal policy. The computational cost of policy evaluation by solving a linear system is $\mathcal{O}\left(|\mathcal{S}|^3\right)$,[8] while the cost of the policy improvement is $\mathcal{O}\left(|\mathcal{S}|^2|\mathcal{A}|\right)$. Therefore, the computational cost of T iterations of PI is $\mathcal{O}\left(T|\mathcal{S}|^2\left(|\mathcal{S}|+|\mathcal{A}|\right)\right)$.

Compared to VI, PI models explicitly the policy, while VI simply considers the value function and the policy comes into place only at the end. It has been proved that, under suitable conditions, PI enjoys a *quadratic* convergence rate (Mansour and Singh, 1999; Puterman, 2014), compared to the linear rate of VI. This justifies the empirical evidence that PI usually converges faster than VI. Finally, it was proven that PI is *strongly polynomial* (Ye, 2011) and converges to the optimal policy in at most $\mathcal{O}\left(\frac{|\mathcal{A}|}{1-\gamma}\log\frac{1}{1-\gamma}\right)$ (Scherrer, 2013).

[8] For general square systems, the complexity can be reduced to $|\mathcal{S}|^{2.376}$ (Golub and Van Loan, 1996).

2.8.3 Linear Programming

The solution of finite MDPs can be also addressed by means of linear programming (Wang et al., 2007). In the discounted case ($\gamma < 1$), the primal LP problem can be stated as follows:

$$\begin{aligned} \min_{\mathbf{v}\in\mathbb{R}^{|\mathcal{S}|}} \quad & \sum_{s\in\mathcal{S}} \nu_0(s) v(s) \\ \text{s.t.} \quad & v(s) \geqslant r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) v(s') \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}, \end{aligned}$$

where $\nu_0 \in \mathscr{P}(\mathcal{S})$ is a distribution over the state space such that $\nu_0(s) > 0$ for all $s \in \mathcal{S}$. The optimization problem is a linear program with $|\mathcal{S}|$ variables and $|\mathcal{S}||\mathcal{A}|$ constraints. It is possible to prove that the solution of this problem is the optimal value function V^* and an optimal policy can be recovered, as usual, as a greedy policy w.r.t. V^*. Using the Lagrangian duality it is possible to rephrase the dual LP (Wang et al., 2007):

$$\begin{aligned} \max_{\mathbf{d}\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad & \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \nu(s,a) r(s,a) \\ \text{s.t.} \quad & \sum_{a\in\mathcal{A}} \nu(s',a) = (1-\gamma)\nu_0(s') + \gamma \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \nu(s,a) p(s'|s,a) \quad \forall s' \in \mathcal{S} \\ & \nu(s,a) \geqslant 0 \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}. \end{aligned}$$

The dual problem is an LP with $|\mathcal{S}||\mathcal{A}|$ variables and $|\mathcal{S}|$ constraints (neglecting the non-negativity constraints). Thus, it is in general preferred to solve the dual formulation. The solution of the dual LP ν^* is the γ-discounted stationary distribution induced by the initial state distribution ν_0 and an optimal policy (Wang et al., 2007). Thus, an optimal policy can be recovered a posteriori for every state action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ as:

$$\pi^*(a|s) = \frac{\nu^*(s,a)}{\sum_{a'\in\mathcal{A}} \nu^*(s,a')}.$$

Although the worst-case computational complexity results of solving MDPs with LP are better than those of VI and PI, typically the DP approaches tend empirically to converge faster (Littman, 1996).

CHAPTER 3

Reinforcement Learning Algorithms

The exact solution methods presented in Chapter 2, assume the full knowledge of the environment dynamics and of the reward function. These methods do not scale to large state-action spaces and they are inapplicable when the transition model or the reward functions are unknown. In real-world applications, typically, the environment dynamics is either approximated by a complex model, that is usually computationally expensive, or completely unknown. Consequently, sampling the environment becomes essential to estimate, implicitly or explicitly, its dynamics and reward generation process. The sampling process comes with uncertainty and managing uncertainty is at the basis of any RL algorithm. In the following, we provide an overview of a selection of RL algorithms. This chapter has no claim to be exhaustive; instead, it has to be considered just auxiliary to the effective understanding of the subsequent chapters. For a complete review of the RL algorithms, we refer the reader to the distinguished Sutton and Barto's book (Sutton and Barto, 2018).

Reinforcement Learning Dichotomies The literature has extensively studied the RL problem and proposed a heterogeneous variety of algorithms, that can be categorized according to different dimensions.

Model-based vs model-free Model-based RL algorithms (e.g., Deisenroth and Rasmussen, 2011; Nagabandi et al., 2018; Wang et al., 2019a) aim at explicitly estimating the transition model and the reward function of the environment and then employ them, possibly with an exact solution method, to derive an approximate value function and/or approximately optimal policy. Instead, model-free approaches (e.g., Mnih

et al., 2015; Schulman et al., 2015; Lillicrap et al., 2016; Duan et al., 2016) do not represent the transition model nor the reward function, but employ samples to directly estimate the value function and/or the optimal policy.

On-policy vs off-policy When the policy that is employed to collect samples is the same policy that is learned, then we speak of on-policy RL algorithms (e.g., Williams, 1992; Rummery and Niranjan, 1994; Jaksch et al., 2010). Whereas, if a *behavioral* (or baseline) policy is used to explore the environment and a different policy, named *target* policy, is optimized, we are in presence of an off-policy algorithm (e.g., Watkins and Dayan, 1992; Ernst et al., 2005; Silver et al., 2014; Schulman et al., 2017; Metelli et al., 2018b).

On-line vs off-line On-line (or *incremental*) algorithms perform the sample collection during the learning process. Thus, the algorithm has possibly access to fresh samples every iteration (e.g., Watkins and Dayan, 1992; Jaksch et al., 2010; Schulman et al., 2017). Instead, off-line (or *batch*) RL algorithms have access to a dataset of samples previously collected and no further interaction with the environment is allowed (e.g., Lange et al., 2012; Ernst et al., 2005; D'Oro et al., 2020). Clearly, off-line algorithms are necessarily off-policy.

Tabular vs function approximation When dealing with finite state-action MDPs, the value functions can be represented as a finite array. In such a case, we refer to tabular RL (e.g., Watkins and Dayan, 1992; Rummery and Niranjan, 1994). Clearly, when the size of the state-action space grows or becomes infinite, we need to employ a function space to approximate the value function or the optimal policy. In such a case, we speak of function approximation (e.g., Munos, 2005; Scherrer, 2014).

Value-based vs policy-based vs actor-critic Value-based (or *critic-only*) methods aim at learning an optimal value function and, then, derive the optimal policy as a greedy policy (e.g., Watkins and Dayan, 1992; Rummery and Niranjan, 1994; Munos, 2005; Scherrer, 2014). Policy-based (or *actor-only*) methods, instead, do not represent the value function but focus on directly learning an optimal policy (e.g., Williams, 1992; Baxter and Bartlett, 2001; Pirotta et al., 2013a). Finally, actor-critic approaches (e.g., Konda and Tsitsiklis, 1999; Lillicrap et al., 2016) combine the formers and model explicitly both the policy (actor) and the value function corresponding to the current policy (critic).

Chapter Outline The chapter is organized as follows. We start in Section 3.1 by revising the basics of temporal difference learning in tabular MDPs, with particular attention to SARSA and Q-learning algorithms. In Section 3.2, we survey the fundamental aspects of function approximation, with specific reference to approximate value iteration and approximate policy iteration and the corresponding error propagation results. Finally, in Section 3.3, we focus on policy search revising the policy gradients methods and trust-region methods.

3.1 Temporal Difference Methods

In this section, we briefly survey the *Temporal Difference* (TD) methods (Sutton, 1985, 1988), a class of online value-based RL algorithms. We start with the prediction problem, i.e., the problem of estimating the value function of a given policy (Section 3.1.1), in order to introduce the basic concepts. Then, we focus on the control problem that consists in learning the optimal value function (Section 3.1.2). We restrict our attention to *tabular* MDPs, i.e., problems with finite state-action spaces.

3.1.1 TD Prediction

Given a policy $\pi \in \Pi^{\mathrm{SR}}$, the *prediction* problem can be stated as estimating the value function V^π of π by observing some trajectories of interaction with the environment. The estimation process is performed iteratively; at each time step $t \in \mathbb{N}$, state $S_t \in \mathcal{S}$ is observed and the estimated value function is updated according to the following rule:

$$V^{(t+1)}(S_t) = (1 - \alpha^{(t)})V^{(t)}(S_t) + \alpha^{(t)}G^{(t)},$$

where $(\alpha^{(t)})_{t\in\mathbb{N}}$ is a learning rate schedule and $G^{(t)}$ is an estimator of the value of policy π in state S_t obtained from samples. The quantity $\delta^{(t)} = G^{(t)} - V^{(t)}$ is usually called *temporal difference error* (Sutton and Barto, 2018). Different choices of $G^{(t)}$ lead to different TD methods.

n-step Returns Considering a trajectory $\tau = (S_t, R_{t+1}, S_{t+1}, R_{t+2}, \dots)$, in which we have neglected the actions, starting in state $S_t \in \mathcal{S}$ and given $n \in \mathbb{N}_{\geqslant 1}$, we can define the n-step return as (Sutton, 1988):

$$G_n^{(t)} = \sum_{l=0}^{n-1} \gamma^l R_{t+l+1} + \gamma^n V^{(t)}(S_{t+n}).$$

Monte-Carlo vs Temporal Difference A particular case is $n = T(\tau)$, i.e., the trajectory length, leading to the *Monte-Carlo* (MC) return:

$$G_{\mathrm{MC}}^{(t)} = \sum_{l=0}^{T(\tau)-1} \gamma^l R_{t+l+1}.$$

Clearly, MC requires considering a full-length trajectory and it is applicable only to episodic MDPs, in which all trajectories are guaranteed to reach an absorbing state. MC has the desirable property of generating an unbiased estimator for V^π, i.e., $\mathbb{E}^\pi \left[G_{\mathrm{MC}}^{(t)} | S_t \right] = V^\pi(S_t)$ but, usually, it displays a large variance (Kearns and Singh, 2000). We can additionally distinguish between *first-visit* MC, in which whenever a state is encountered multiple times in a trajectory the update is performed for the first occurrence only, and *every-visit* MC, in which the update is performed at every occurrence. Another remarkable case is $n = 1$ that corresponds to the 1-step return, leading to the well-known TD(0) method (Sutton, 1988):

$$G_1^{(t)} = R_{t+1} + \gamma V^{(t)}(S_{t+1}).$$

The important property of TD(0), and more generally of all TD algorithms, is the *bootstrapping*, i.e., the reuse of the current estimate of the value function $V^{(t)}(S_{t+1})$ evaluated in the next state. TD overcomes the MC limitation of requiring an episodic MDP, as it is no longer necessary to wait for the end of the trajectory in order to perform an update. Moreover, TD is typically affected by a lower variance w.r.t. MC at the price of introducing a bias due to the bootstrapping operation. Nevertheless, TD estimator preserves consistency as the number of samples grows. Another relevant distinction is that MC does not exploit the Markov property of the environment, whereas TD does. This explains, at least at an intuitive level, why TD typically performs better than MC in Markovian environments (Sutton and Barto, 2018).

TD(λ) A way of unifying the n-step returns consists in combining them via an exponential averaging. Specifically, given $\lambda \in [0, 1]$, we can define λ-return as (Sutton, 1985):

$$G_{\lambda}^{(t)} = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_{n}^{(t)}.$$

Special cases are when $\lambda = 0$, that corresponds to the TD(0) update and $\lambda = 1$ that reduces to the first-visit MC method. In principle, to apply TD(λ) we have to wait for the end of the trajectory, in order to be able to compute all n-step returns (backward view). To overcome this limitation it is possible to employ *eligibility traces* (forward view) that quantify the impact of the current TD error on all states of the MDP (Singh and Sutton, 1996; Sutton and Barto, 2018). A bias-variance analysis of n-step TD and TD(λ) is provided in (Kearns and Singh, 2000).

3.1.2 TD Control

When moving from the prediction to the *control* problem, i.e., the problem of learning the optimal value function V^* of an MDP, we face additional challenges. First of all, since the ultimate goal consists in producing an approximation of the optimal policy, we need to estimate the Q-function, instead of the V-function, in order to output the corresponding greedy policy. The update rule for the Q-function is the following, defined for a state-action pair $(S_t, A_t) \in \mathcal{S} \times \mathcal{A}$ and $t \in \mathbb{N}$ as:

$$Q^{(t+1)}(S_t, A_t) = (1 - \alpha^{(t)}) Q^{(t)}(S_t, A_t) + \alpha^{(t)} G^{(t)},$$

where $(\alpha^{(t)})_{t \in \mathbb{N}}$ is a learning rate schedule and $G^{(t)}$ now is an estimator of the optimal value function $V^*(S_t)$ obtained from samples. Before showing how $G^{(t)}$ can be defined using on-policy and off-policy TD approaches, we focus on the second, and most relevant, challenge of control: *exploration*.

Exploration Strategies A crucial aspect of the control problem is that, in order to collect useful information from the environment for estimating the value function, an *exploration* strategy $(\pi^{(t)})_{t \in \mathbb{N}}$ is needed. We already introduced the *exploration-exploitation trade-off* in Chapter 2 as the dilemma between playing the action that is currently believed to be optimal (the greedy action w.r.t. to $Q^{(t)}$) and collecting new samples to refine the Q-function estimate.

Typical *undirected* exploration strategies are ϵ-*greedy* and *Boltzmann* exploration. In the former case, a greedy action $\arg\max_{a\in\mathcal{A}}\{Q^{(t)}(s,a)\}$ is played with probability $1-\epsilon$ whereas an action chosen uniformly in $\mathcal{A}$ is played with probability ϵ, with $\epsilon \in [0,1]$. Boltzmann exploration instead prescribes to play an action with a probability proportional to $\exp\left(\frac{Q^{(t)}(s,a)}{\tau}\right)$ where $\tau \in \mathbb{R}_{\geqslant 0}$ is called the *temperature*. Thus, actions with high estimated Q-function are exponentially preferred. Both exploration strategies can be made GLIE (Greedy in the Limit with Infinite Exploration), i.e., they converge to the greedy policy, under the assumption that $\epsilon \to 0$ for the ϵ-greedy and $\tau \to 0$ for the Boltzmann as $t \to \infty$.

Although these exploration strategies allow reaching convergence to the optimal Q-function under certain conditions (Singh et al., 2000), they are not *provably efficient*, unless unrealistic assumptions are enforced (Auer et al., 2002; Cesa-Bianchi et al., 2017). A number of approaches have been proposed in the literature to achieve provable efficiency employing more *directed* exploration strategies (e.g., Kearns and Singh, 2002; Brafman and Tennenholtz, 2002; Strehl et al., 2006; Strehl and Littman, 2008; Jaksch et al., 2010; Jin et al., 2018; Metelli et al., 2019b; Jin et al., 2020). A complete treatment of the exploration problem in RL is out of the scope of this dissertation.

On-policy TD Control In on-policy control, we estimate the Q-function of the policy $\pi^{(t)}$ we are currently running for exploration. It immediately follows that for convergence to the optimal Q-function, it is necessary that the exploration policy changes during the learning process, converging ultimately to the greedy policy. In order to define the term $G^{(t)}$ we can employ the same approaches used for prediction, with the only difference that we use the Q-function instead of the V-function. Specifically, given a trajectory $\tau = (S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \dots)$, we can define the SARSA(n) algorithm, based on the following n-step return (Rummery and Niranjan, 1994):

$$G^{(t)}_{\text{SARSA}(n)} = \sum_{l=0}^{n-1} \gamma^l R_{t+l+1} + \gamma^n Q^{(t)}(S_{t+n}, A_{t+n}).$$

Since we are considering MDPs with finite actions, we realize that it is not necessary to wait for action A_{t+n} as we can compute exactly the expectation over the action space, once we know the exploration policy $\pi^{(t)}$. This observation leads to the Expected SARSA(n) algorithm (van Seijen et al., 2009):

$$G^{(t)}_{\mathbb{E}\text{SARSA}(n)} = \sum_{l=0}^{n-1} \gamma^l R_{t+l+1} + \gamma^n \sum_{a\in\mathcal{A}} \pi^{(t)}(a|S_{t+n}) Q^{(t)}(S_{t+n}, a).$$

Clearly, by combining these terms with exponential average, just like in the prediction methods, we obtain the SARSA(λ) and Expected SARSA(λ) algorithms (Sutton and Barto, 2018). These algorithms converge to the optimal value function under the GLIE condition if every state-action pair is visited infinitely often and under the Robbins-Moore conditions on the learning rate (Singh et al., 2000).[1]

[1]The Robbins-Moore conditions require that $\sum_{t\in\mathbb{N}} \alpha^{(t)} = \infty$ and $\sum_{t\in\mathbb{N}} (\alpha^{(t)})^2 < \infty$.

Algorithm 3.1: Temporal Difference Control (TD).

Input: T number of iterations, $Q^{(0)}$ initial action-value function, $(\alpha^{(t)})_{t\in\mathbb{N}}$ learning rate schedule, $(\pi^{(t)})_{t\in\mathbb{N}}$ exploration policy schedule
Output: greedy policy $\widehat{\pi}$

forall $t = 0, \dots, T-1$ **do**
  Play action $A_t \sim \pi^{(t)}(\cdot|S_t)$
  Observe state S_{t+1} and the reward R_{t+1}
  Compute $G^{(t)}$ (e.g., using SARSA or Q-learning)
  $Q^{(t+1)}(S_t, A_t) = (1-\alpha^{(t)})Q^{(t)}(S_t, A_t) + \alpha^{(t)}G^{(t)}$
$\pi^{(T)} \in \arg\max_{a\in\mathcal{A}}\{Q^{(T)}(s,a)\}, \quad \forall s \in \mathcal{S}$
return $\pi^{(T)}$

Off-policy TD Control In the off-policy TD methods, we employ one policy to carry out exploration whereas we learn the value function of a different policy, specifically the one of the optimal policy. The most popular off-policy TD algorithm is *Q-learning* (Watkins and Dayan, 1992), based on the idea of applying an empirical version of the Bellman optimality operator T^*. Thus, given a trajectory $\tau = (S_t, A_t, R_{t+1}, S_{t+1}, \dots)$, we define the Q-learning return as:

$$G_{\text{QL}}^{(t)} = R_{t+1} + \gamma \max_{a\in\mathcal{A}} \left\{ Q^{(t)}(S_{t+1}, a) \right\}.$$

We can rewrite $G^{(t)} = \widehat{T}^* Q^{(t)}$, where $\widehat{T}^*$ is the empirical Bellman optimality operator, that is unbiased conditioned to the current state-action pair (S_t, A_t):

$$\mathbb{E}^{\pi}\left[\left(\widehat{T}^* f \right)(S_t, A_t) | S_t, A_t \right] = \left(T^* f \right)(S_t, A_t).$$

The convergence of Q-learning can be guaranteed even for non GLIE policies under the assumption that every state-action pair is visited infinitely often and under the Robbins-Moore conditions on the learning rate (Singh et al., 2000). The convergence rate of Q-learning was first studied in the asymptotic regime in (Szepesvári, 1997) and subsequently in (Even-Dar and Mansour, 2003; Beck and Srikant, 2012; Qu and Wierman, 2020; Li et al., 2020) with also finite-time guarantees. Numerous extensions of Q-learning using the multi-step TD(λ) approach have been proposed (Watkins, 1989; Peng and Williams, 1996), including unifying approaches, such as Q(σ) (Sutton and Barto, 2018).

The pseudocode of a general TD control algorithm is reported in Algorithm 3.1. For a complete view of TD methods refer to (Sutton and Barto, 2018, Chapter 5, 6, 7, and 12).

3.2 Function Approximation

The methods we have presented above leverage on the tabular representation available for finite MDPs. When the state-action space is too large or even continuous, tabular methods become infeasible. A possible path to overcome this problem is *discretization* (e.g., Uther and Veloso, 1998) or *state aggregation* (e.g., Singh et al., 1994) that allow recovering a

finite MDP that can be regarded as an approximation of the original one. Another way to approach the problem is *function approximation*. In this case, we decide to approximately represent the Q-function, by resorting to a function space $\mathcal{F} \subseteq \mathscr{B}(\mathcal{S} \times \mathcal{A})$. Thus, we look for the best approximation of the optimal Q-function Q^* within the space $\mathcal{F}$:

$$\widehat{Q} \in \arg\min_{f \in \mathcal{F}} \left\{ \|f - Q^*\|_{p,\rho} \right\}, \tag{3.1}$$

for some $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ evaluation distribution and $p \geqslant 1$. Given this approximation $\widehat{Q} \in \mathcal{F}$, we can derive the control policy $\widehat{\pi}$ as a greedy policy w.r.t. $\widehat{Q}$. The following result shows that a good approximation of Q^* determines a greedy policy whose performance is close to V^*.

Theorem 3.1 (Singh and Yee (1994), Corollary 2)**.** *Let $\widehat{Q} \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and let $\widehat{\pi} \in \arg\max_{a \in \mathcal{A}}\{\widehat{Q}(\cdot, a)\}$ be a greedy policy w.r.t. $\widehat{Q}$. Then for every state $s \in \mathcal{S}$ it holds that:*

$$V^{\widehat{\pi}}(s) \geqslant V^*(s) - \frac{2}{1-\gamma} \left\| \widehat{Q} - Q^* \right\|_\infty .$$

Clearly, $\mathcal{F}$ can be either a parametric or non-parametric function space and its choice needs to be guided by the usual *bias-variance trade-off* (Györfi et al., 2002; Bishop, 2007). There exists a significantly large surge of RL algorithms based on function approximation for both prediction and control, including Gradient TD methods (e.g., Boyan and Moore, 1994; Sutton et al., 2008, 2009; Maei et al., 2009), Least Squares Temporal Difference (LSTD, Bradtke and Barto, 1996; Boyan, 2002; Xu et al., 2002; Nedic and Bertsekas, 2003), Least Squares Policy Evaluation (LSPI, Bertsekas and Ioffe, 1996; Lagoudakis and Parr, 2003; Bertsekas et al., 2004). For an extensive review of approximate solution methods refer to (Sutton and Barto, 2018, Chapters 9, 10, and 11) and (Szepesvári, 2010, Sections 3 and 4). In the following we focus on batch RL methods, specifically on Approximate Dynamic Programming approaches (API, Bertsekas, 2005; Powell, 2007), which rephrase policy iteration and value iteration in a version obtained through samples.[2]

3.2.1 Approximate Value Iteration

Approximate Value Iteration (AVI, Gordon, 1995; Munos, 2005) can be thought as a version of VI in which the application of the Bellman optimal operator T^* is replaced by its empirical version $\widehat{T}^*$. We assume to be provided with a batch of transitions $\mathcal{D} = \{(S_i, A_i, S'_i, R_i)\}_{i=1}^n$ collected with a sampling distribution $\nu \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$. Clearly, whenever AVI is performed on continuous-state MDPs, we need to introduce an approximation space $\mathcal{F} \subset \mathscr{B}(\mathcal{S} \times \mathcal{A})$. Thus, at each iteration $t \in \mathbb{N}$, AVI is composed of two stages. Given the current approximation of the Q-function $Q^{(t)} \in \mathcal{F}$, we first perform an application of the empirical Bellman operator $\widehat{T}^* Q^{(t)}$. Then, we project back this quantity onto $\mathcal{F}$ by means of a projection operator $\Pi_{\mathcal{F}} : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathcal{F}$ that is typically implemented as a least squared regression. This procedure generates an *approximation* error:

$$\epsilon^{(t)} = T^* Q^{(t)} - Q^{(t+1)},$$

[2] We are going to present the API algorithms for estimating the Q-function, instead of the V-function, because we are mainly interested in the control problem rather than the prediction problem.

Algorithm 3.2: Approximate Value Iteration (AVI).

Input: J number of iterations, $Q^{(0)}$ initial action-value function, $\mathcal{F}$ function space, $\mathcal{D} = \{(S_i, A_i, S'_i, R_i)\}_{i=1}^n$ batch samples
Output: greedy policy $\pi^{(J)}$

forall $j = 0, \dots, J-1$ **do**
  $Y_i^{(j)} = \widehat{T}^* Q^{(j)}(S_i, A_i), \quad i \in \{1, ..., n\}$
  $Q^{(j+1)} \in \arg\min_{f \in \mathcal{F}} \left\{ \|f - Y^{(j)}\|_{2,\mathcal{D}}^2 \right\}$
$\pi^{(J)}(s) \in \arg\max_{a \in \mathcal{A}} \{Q^{(J)}(s,a)\}, \quad \forall s \in \mathcal{S}$
return $\pi^{(J)}$

where $Q^{(t+1)} = \Pi_{\mathcal{F}} \widehat{T}^* Q^{(t)}$. This error incorporates an estimation component, due to the usage of the empirical operator $\widehat{T}^*$ instead of the exact one T^* and a (properly called) approximation error due to the projection onto the function space $\mathcal{F}$. A pseudocode of AVI is reported in Algorithm 3.2. Some examples of AVI are tree-based Fitted Q-Iteration (FQI, Ernst et al., 2005), multilayer perceptron-based Fitted Q-Iteration (Riedmiller, 2005), and regularized Fitted Q-iteration (Farahmand, 2011). An extension to account for the continuous action spaces was proposed in (Antos et al., 2007). The theoretical analysis of the error propagation in AVI algorithms was studied extensively and progressively refined (Bertsekas and Tsitsiklis, 1996; Munos and Szepesvári, 2008; Antos et al., 2008; Farahmand, 2011). We report the following result due to (Farahmand, 2011).

Theorem 3.2 (Theorem 3.4 of (Farahmand, 2011))**.** *Let $p \geqslant 1$, $J \in \mathbb{N}_{\geqslant 1}$ and $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$. Then for any sequence $(Q^{(j)})_{j=0}^J \subset \mathcal{F}$ uniformly bounded by $Q_{\max} \leqslant \frac{R_{\max}}{1-\gamma}$, the corresponding $(\epsilon^{(j)})_{j=0}^{J-1}$ and for any $r \in [0,1]$ it holds that:*

$$\left\| Q^* - Q^{\pi^{(J)}} \right\|_{p,\rho} \leqslant \frac{2\gamma}{(1-\gamma)^2} \left[\frac{2}{1-\gamma} \gamma^{\frac{J}{p}} R_{\max} + C_{\mathrm{VI},\rho,\nu}^{\frac{1}{2p}}(J,r) \mathcal{E}^{\frac{1}{2p}}(\epsilon^{(0)}, \dots, \epsilon^{(J-1)}; r) \right],$$

where $C_{\mathrm{VI},\rho,\nu}$ is a concentrability coefficient whose expression together with $\mathcal{E}$ can be found in (Farahmand, 2011).

The concentrability coefficients $C_{\mathrm{VI},\rho,\nu}$ account for the distribution shift between the sampling distribution ν, the distribution generated by the sequence of policies $(\pi^{(j)})_{j=1}^J$ together with the evaluation distribution ρ. The approximation errors $(\epsilon^{(j)})_{j=1}^J$ can be further analyzed, based on the statistical learning properties of the function space $\mathcal{F}$ (Farahmand, 2011).

3.2.2 Approximate Policy Iteration

Approximate Policy Iteration (API, Scherrer, 2014) can be considered the sample-based version of PI, in which the evaluation step is performed in an approximate way and through samples. Specifically, like for AVI, we assume to be provided with a batch of samples $\mathcal{D} = \{(S_i, A_i, S'_i, R_i)\}_{i=1}^n$ collected with a sampling distribution $\nu \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$. At every iteration $t \in \mathbb{N}$, the *evaluation step*, i.e., the task of computing the value function $Q^{\pi^{(t)}}$ of the current policy $\pi^{(t)}$ is performed in an approximate way, by employing the

samples $\mathcal{D}$ and a function space $\mathcal{F}$. The result is an approximation $Q^{(t)}$ of $Q^{\pi^{(t)}}$. Instead, the improvement step can be performed by computing the greedy policy w.r.t. to $Q^{(t)}$, i.e., $\pi^{(t+1)} \in \arg\max_{a \in \mathcal{A}} \{Q^{(t)}(\cdot, a)\}$.

The evaluation step can be performed in different ways that can be broadly classified into two categories: *Bellman Residual Minimization* (BRM Antos et al., 2008) in which we optimize the error between the value function and the corresponding application of the Bellman operator $\left\| f - \widehat{T}^{\pi} f \right\|_{p,\mathcal{D}}$ and *Least Square Temporal Difference* (LSTD, Bradtke and Barto, 1996) in which we minimize the error between the value function and the projected application of the Bellman operator $\left\| f - \Pi_{\mathcal{F}} \widehat{T}^{\pi} f \right\|_{p,\mathcal{D}}$. The error propagation analysis for API can be carried out by employing differently defined errors:

$$\epsilon_{\text{BR}}^{(t)} = Q^{(t)} - T^{\pi^{(t)}} Q^{(t)},$$
$$\epsilon_{\text{AE}}^{(t)} = Q^{(t)} - Q^{\pi^{(t)}}.$$

$\epsilon_{\text{BR}}^{(t)}$ is the Bellman residual error, that accounts for how far the approximation $Q^{(t)}$ is from the being the fixed point of the operator $T^{\pi^{(t)}}$, whereas $\epsilon_{\text{AE}}^{(t)}$ is the approximation error that quantifies how well $Q^{(t)}$ approximates $Q^{\pi^{(t)}}$ (this includes an estimation and approximation error, just like in AVI). The following result due to (Farahmand, 2011) provides the error propagation.

Theorem 3.3 (Theorem 3.2 of Farahmand (2011))**.** *Let $p \geqslant 1$, $J \in \mathbb{N}_{\geqslant 1}$ and $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$. Then for any sequence $(Q^{(j)})_{j=0}^{J} \subset \mathcal{F}$ uniformly bounded by $Q_{\max} \leqslant \frac{R_{\max}}{1-\gamma}$, the corresponding $(\epsilon^{(j)})_{j=0}^{J-1}$ that can be either ϵ_{BR} or ϵ_{AE} and for any $r \in [0, 1]$ it holds that:*

$$\left\| Q^* - Q^{\pi^{(J)}} \right\|_{p,\rho} \leqslant \frac{2\gamma}{(1-\gamma)^2} \left[\frac{2}{1-\gamma} \gamma^{\frac{J}{p}} R_{\max} + C_{\text{PI},\rho,\nu}^{\frac{1}{2p}}(J, r) \mathcal{E}^{\frac{1}{2p}}(\epsilon^{(0)}, \ldots, \epsilon^{(J-1)}; r) \right],$$

where $C_{\text{PI},\rho,\nu}$ is a concentrability coefficient whose expression together with $\mathcal{E}$ can be found in (Farahmand, 2011).

Similarly to the AVI setting, the concentrability coefficients $C_{\text{PI},\rho,\nu}$ account for the distribution shift and are defined according to which error (ϵ_{BR} or ϵ_{AE}) is employed.

API with Non-Greedy Updates The most traditional API algorithms focus on managing the approximation error in the policy evaluation step (e.g., Lagoudakis and Parr, 2003; Lazaric et al., 2016) and then perform the policy improvement by computing the greedy policy w.r.t. to the approximated Q-function. It has been observed that this approach might lead to an oscillating behavior, that can be ascribed to the *discontinuity* introduced by the greedy step (Bertsekas, 2011; Wagner, 2011). For this reason, a line of research focused on *conservative* updates, in which the greedy improvement is replaced with more prudent updates. An example is Conservative Policy Iteration (CPI, Kakade and Langford, 2002) and subsequently Safe Policy Iteration (SPI, Pirotta et al., 2013b), in which the greedy update is replaced with a *soft* update. In these algorithms, the next policy is computed as a convex combination between the greedy and the old policy:

$$\pi^{(t+1)} = \alpha \pi^{+,(t)} + (1 - \alpha) \pi^{(t)},$$

where $\alpha \in [0, 1]$ and $\pi^{+,(t)}$ is the greedy policy. The value of the coefficient α is selected by optimizing a lower bound on the performance improvement that can be estimated using samples collected with $\pi^{(t)}$. These approaches succeeded to ensure strong theoretical guarantees on the performance improvement and, for this reason, can be considered examples of *safe* RL algorithms (e.g., Pirotta et al., 2013a; García and Fernández, 2015; Papini et al., 2017).

3.3 Policy Search

The methods we have presented so far are value-based, i.e., in order to learn the optimal policy, they first approximate the optimal Q-function and then derive an approximation of the optimal policy as a greedy policy. Clearly, those methods typically require that the action space is finite since they need to compute a maximization over the action space. When the action space is large the computation of the maximum becomes expensive. In these cases and whenever we desire to avoid action discretization, *Policy Search* (PS, Deisenroth et al., 2013) methods come into play. PS explicitly models the policy that is chosen in a suitable approximation space $\Pi \subset \Pi^{\mathrm{SR}}$. Formally, PS can be seen as the task of finding a policy $\pi \in \Pi$ that minimizes the distance between its value function V^{π} and the optimal value function V^*:

$$\widehat{\pi} \in \arg\min_{\pi \in \Pi} \left\{ \left\| V^{\pi} - V^* \right\|_{p,\rho} \right\}, \tag{3.2}$$

where $p \geqslant 1$ and $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ is an evaluation distribution. It is worth noting that both policy-based and value-based methods try to achieve the same objective, i.e., maximizing the performance of the learned policy, but while value-based methods employ the intermediate step of estimating the value function (Equation (3.1)), PS can directly focus on the policy (Equation (3.2)). The explicit presence of a policy space allows modeling restrictions in the behavior the agent can play that arise quite commonly in real-world applications. A large variety of approaches to PS have been proposed in the literature including model-based techniques (e.g., Ng and Jordan, 2000; Ko et al., 2007), expectation-maximization algorithms (e.g., Kober and Peters, 2008), variational inference (e.g., Neumann, 2011), and evolutionary computation (e.g., Heidrich-Meisner and Igel, 2009). In this section, we focus on two classes of approaches that will be relevant in the subsequent chapters.[3]

3.3.1 Policy Gradient Methods

Policy Gradient methods (PG, Williams, 1992; Baxter and Bartlett, 2001) are probably the most straightforward and widespread policy search algorithms. PG algorithms assume that the agent has access to a space of parametric policies:

$$\Pi_{\Theta} = \{\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \mathscr{P}(\mathcal{A}) : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\},$$

where Θ is called parameter space. The goal consists in finding the policy parametrization that maximizes the expected return $J(\boldsymbol{\theta})$ that is an abbreviation for $J^{\pi_{\boldsymbol{\theta}}}$ to highlight

[3] To simplify the mathematical treatment, we will assume that all relevant distributions admit probability density functions w.r.t. the Lebesgue measure.

the dependence on the parameter space. If Π_Θ is a space of *stochastic* and *differentiable* policies in $\boldsymbol{\theta}$, then the expected return $J(\boldsymbol{\theta})$ is differentiable in $\boldsymbol{\theta}$ as well. Stochasticity is essential to ensure exploration unless off-policy estimation techniques are employed (Silver et al., 2014). The gradient of the expected return $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is called *policy gradient* and the following result provides its expression.

Theorem 3.4 (Policy Gradient Theorem (Sutton et al., 1999a))**.** *Let $\mathcal{M}$ be an MDP and $\pi_{\boldsymbol{\theta}} \in \Pi_\Theta$ be a policy. If $\pi_{\boldsymbol{\theta}}$ is stochastic and differentiable in $\boldsymbol{\theta}$, then the policy gradient can be expressed as:*

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \int_{\mathcal{S}\times\mathcal{A}} \mu_\gamma^{\pi_{\boldsymbol{\theta}}}(\mathrm{d}s, \mathrm{d}a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a).$$

The policy gradient admits another expression that can be obtained by rephrasing the policy gradient theorem to the trajectory-based formulation (Peters and Schaal, 2008):

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\nabla_{\boldsymbol{\theta}} \log \mathbb{p}^{\pi_{\boldsymbol{\theta}}}(\tau) G_\gamma(\tau) \right] = \int_{\mathcal{T}} \mathbb{p}^{\pi_{\boldsymbol{\theta}}}(\tau) \nabla_{\boldsymbol{\theta}} \log \mathbb{p}^{\pi_{\boldsymbol{\theta}}}(\tau) G_\gamma(\tau) \mathrm{d}\tau, \tag{3.3}$$

where we recall that $G_\gamma(\tau) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$ is the trajectory return. The expression can be further simplified by observing that the log-gradient of the trajectory density function, for a given trajectory $\tau = (s_0, a_0, r_1, \dots)$ reduces to:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log \mathbb{p}^{\pi_{\boldsymbol{\theta}}}(\tau) &= \nabla_{\boldsymbol{\theta}} \log \left(\mu_0(s_0) \prod_{t=0}^{\infty} \pi_{\boldsymbol{\theta}}(a_t|s_t) p(s_{t+1}|s_t, a_t) r(s_{t+1}|s_t, a_t, s_{t+1}) \right) \\
&= \sum_{t=0}^{\infty} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t).
\end{aligned}$$

Once we computed the policy gradient, we can regard at the RL problem as *stochastic optimization*. The most straightforward optimization approach is a simple gradient ascent over the parameters, usually called *vanilla gradient* (Peters and Schaal, 2008):

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha^{(t)} \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}),$$

where $(\alpha^{(t)})_{t\in\mathbb{N}}$ is a learning rate schedule. More sophisticated approaches include *natural gradient* (Kakade, 2001; Peters et al., 2005), in which the policy gradient is premultiplied by the inverse Fisher Information Matrix (FIM, Fisher, 1922), second-order methods (Furmston and Barber, 2012; Manganini et al., 2015), and coordinate ascent (Papini et al., 2017).

Clearly, the policy gradient expression cannot be computed exactly in the RL setting since it requires the knowledge of the transition model and the reward function in order to compute either the γ-discounted stationary distribution $\mu_\gamma^{\pi_{\boldsymbol{\theta}}}$ or the trajectory density function $\mathbb{p}^{\pi_{\boldsymbol{\theta}}}$. In practice, we resort to estimators that can be computed from samples, such as *likelihood ratio methods* (Peters and Schaal, 2008), that we introduce in the following. The general pseudocode of PG is reported in Algorithm 3.3.

Algorithm 3.3: Policy Gradient (PG).

Input: MDP $\mathcal{M}$, number of iterations T, learning rate schedule $(\alpha^{(t)})_{t=0}^{T-1}$
Output: approximately optimal policy parameters $\boldsymbol{\theta}^{(T)}$

Initialize $\boldsymbol{\theta}^{(0)}$ arbitrarily
forall $t = 0, 1, \dots, T-1$ **do**
  Estimate the policy gradient $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)})$
  Update the parameters $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha^{(t)} \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)})$
return $\boldsymbol{\theta}^{(T)}$

REINFORCE The REINFORCE estimator (Williams, 1992) is obtained by rephrasing the policy gradient expression in Equation (3.3) in a sample-based version, in which we replace the expectation with the corresponding sample mean. Specifically, given a set of finite-length trajectories $\{\tau_i\}_{i=1}^n$ collected with $\mathbb{P}^{\pi_{\boldsymbol{\theta}}}$ the estimator is given for every $k \in \{1, \dots, p\}$ as:

$$\widehat{\nabla}^{\mathrm{RF}}_{\boldsymbol{\theta}_k} J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{T(\tau_i)-1} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_{\tau_i,t} | S_{\tau_i,t}) \right) \left(\sum_{t=0}^{T(\tau_i)-1} \gamma^t R_{\tau_i,t+1} - \mathbf{b}_k \right),$$

where $\mathbf{b} \in \mathbb{R}^p$ is a *baseline* (Peters and Schaal, 2008) that is used to reduce the variance of the estimate, while preserving the unbiasedness of the estimator. Indeed, it is possible to prove that for a non-random vector $\mathbf{b} \in \mathbb{R}^p$ we have that $\mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_{\tau_i,t} | S_{\tau_i,t}) \mathbf{b}_k \right] = 0$. Therefore, it is convenient to derive the value of the baseline that minimizes the variance of the estimator (Peters and Schaal, 2008), defined for every $k \in \{1, \dots, p\}$ as:

$$\mathbf{b}_k^{\mathrm{RF}*} = \frac{\mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\left(\sum_{t=0}^{\infty} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_t | S_t) \right)^2 G_\gamma(\tau) \right]}{\mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\left(\sum_{t=0}^{\infty} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_t | S_t) \right)^2 \right]}.$$

G(PO)MDP One of the main drawbacks of REINFORCE is the high variance of the gradient estimate. This phenomenon can be ascribed to the fact that REINFORCE does not leverage the *causality* between actions and rewards. Indeed, we immediately realize that the reward ad a given time step $t \in \mathbb{N}$ is independent of the actions performed at timesteps $t' > t$. This simple observation allows simplifying the expression of the policy gradient as:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{\infty} \left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_l | S_l) \right) \gamma^t R_{t+1} \right],$$

where we exploited the causality identity $\mathbb{E}^{\pi_{\boldsymbol{\theta}}} \left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_l | S_l) \gamma^t R_{t+1} \right] = \mathbf{0}$, whenever $l > t$. This allows deriving the G(PO)MDP estimator (Baxter and Bartlett, 2001), obtained from a set of trajectories $\{\tau_i\}_{i=1}^n$ and every $k \in \{1, \dots, p\}$ as:

$$\widehat{\nabla}^{\mathrm{G(PO)MDP}}_{\boldsymbol{\theta}_k} J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T(\tau_i)-1} \left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_{\tau_i,l} | S_{\tau_i,l}) \right) \left(\gamma^t R_{\tau_i,t+1} - \mathbf{b}_{t,k} \right),$$

where similarly to the REINFORCE estimator, $\mathbf{b}_t \in \mathbb{R}^p$ is a *time-dependent baseline*, whose optimal value minimizing the variance is obtained for every $t \in \mathbb{N}$ and $k \in \{1, \dots, p\}$ as (Peters and Schaal, 2008):

$$\mathbf{b}_{t,k}^{\mathrm{G(PO)MDP}*} = \frac{\mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_l|S_l)\right)^2 \gamma^t R_{t+1}\right]}{\mathbb{E}^{\pi_{\boldsymbol{\theta}}}\left[\left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\theta}_k} \log \pi_{\boldsymbol{\theta}}(A_l|S_l)\right)^2\right]}.$$

A study of the statistical properties of REINFORCE and G(PO)MDP estimators can be found in (Zhao et al., 2011; Pirotta et al., 2013a; Papini et al., 2019b).

3.3.2 Trust-Region Methods

PG methods are effective approaches to address continuous control tasks, especially in presence of continuous action spaces. However, they are online by nature, as a single batch of trajectories can be employed to perform just an individual update. Then, after each update, further interaction with the environment is needed to collect fresh samples. This is clearly inefficient since the same batch of samples could be used, in principle, to perform multiple updates. Moreover, PGs are *local* methods since they employ first-order information, such as the gradient, to identify an improvement direction. Other methods, instead, perform the optimization of the policy parameters in a neighborhood of the current parametrization. These methods are called *trust-region* (e.g., Schulman et al., 2015) and they are based on the idea that we can employ the samples collected with one (*behavioral*) policy to estimate the performance of other (*target*) policies, provided that the two policies are not too "dissimilar". In recent years, an incredibly large number of algorithms falling in this category have been proposed (e.g., Peters et al., 2010; Daniel et al., 2012; Schulman et al., 2015, 2017; Metelli et al., 2018b; Wang et al., 2019c,b; Metelli et al., 2020b). In this section, we start introducing *Importance Sampling* (IS, Owen, 2013) and then we revise two examples of trust-region methods, whose knowledge is necessary for the understanding of the subsequent chapters: *Relative Entropy Policy Search* (REPS, Peters et al., 2010) and *Policy Optimization via Importance Sampling* (POIS, Metelli et al., 2018b).

Importance Sampling The fundamental statistical tool at the basis of a large number of trust-region methods is *importance sampling* (Owen, 2013). Given two probability measures $\mu, \nu \in \mathscr{P}(\mathcal{X})$ and a bounded measurable function $f \in \mathscr{B}(\mathcal{X})$, IS allows estimating the expected value of function f under the target distribution μ, having samples collected with the behavioral distribution ν. Under the assumption that $\mu \ll \nu$, i.e., μ is absolutely continuous w.r.t. ν, the IS estimator reweights each sample with the *likelihood ratio* or *importance weight*:[4]

$$\widehat{J}_{\mu/\nu} = \frac{1}{n} \sum_{i=1}^{n} \frac{\mu(x_i)}{\nu(x_i)} f(x_i),$$

[4]We assume that μ and ν admit probability density function w.r.t. the Lebesgue measure, denoted with the same symbols.

where $\omega_{\mu/\nu}(x) = \frac{\mu(x)}{\nu(x)}$ is the importance weight and $x_i \sim \nu$ independently for all $i \in \{1, \dots, n\}$. This estimator is unbiased and its variance can be bounded in terms of the α-Rényi divergence between the probability measures μ and ν (Metelli et al., 2018b), a dissimilarity index between probability distributions (Rényi, 1961).

The Rényi divergence is defined for two probability measures $\mu, \nu \in \mathscr{P}(\mathcal{X})$ such that $\mu \ll \nu$, for every $\alpha \in [0, \infty]$ as:[5]

$$D_\alpha(\mu \| \nu) = \frac{1}{\alpha - 1} \log \int_{\mathcal{X}} \nu(x) \left(\frac{\mu(x)}{\nu(x)} \right)^\alpha \mathrm{d}x.$$

Furthermore, we define the exponentiated Rényi divergence as $d_\alpha(\mu \| \nu) = \exp\left(D_\alpha(\mu \| \nu)\right)$ (Cortes et al., 2010).

Based on the results of (Metelli et al., 2020b), the variance of the IS estimator can be bounded for every $\alpha \in [1, \infty]$ as:

$$\underset{x_i \sim \nu}{\mathbb{V}\mathrm{ar}} \left[\widehat{J}_{\mu/\nu} \right] \leqslant \frac{1}{n} \|f\|^2_{\frac{2\alpha}{\alpha-1}, \nu} \, d_{2\alpha}\left(\mu \| \nu\right)^{2 - \frac{1}{\alpha}}.$$

A common choice is $\alpha = 1$. As intuition suggests, the larger the divergence between the two distributions, the larger the variance. Indeed, in presence of significantly dissimilar distributions, the samples collected with one distribution provide poor information about the other. The extension of these results to multiple importance sampling (Veach and Guibas, 1995), i.e., the case in which multiple behavioral distributions are considered was provided in (Papini et al., 2019a; Metelli et al., 2020b).

Relative Entropy Policy Search Relative Entropy Policy Search (REPS, Peters et al., 2010; Daniel et al., 2012) is an information theoretic approach to PS that formulates the RL problem as finding the stationary distribution $\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ that maximizes the expected return. The search is constrained in a trust-region centered in the stationary distribution $\mu^\pi \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ induced by the current policy $\pi \in \Pi^{\mathrm{SR}}$ and formalized in terms of a KL-divergence constraint. The optimization problem can be stated in terms of the KL-divergence threshold $\kappa \in \mathbb{R}_{\geqslant 0}$ as:

$$\begin{aligned}
&\max_{\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A})} J^\mu = \int_{\mathcal{S} \times \mathcal{A}} \mu(s,a) r(s,a) \mathrm{d}s\mathrm{d}a \\
&\text{s.t.} \quad D_{\mathrm{KL}}(\mu \| \mu^\pi) = \int_{\mathcal{S} \times \mathcal{A}} \mu(s,a) \log \frac{\mu(s,a)}{\mu^\pi(s,a)} \mathrm{d}s\mathrm{d}a \leqslant \kappa, \\
&\qquad \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \mu(s,a) p(s'|s,a) \boldsymbol{\phi}(s) \mathrm{d}s\mathrm{d}a\mathrm{d}s' = \int_{\mathcal{S} \times \mathcal{A}} \mu(s',a') \boldsymbol{\phi}(s') \mathrm{d}s'\mathrm{d}a',
\end{aligned}$$

where the second constraint is a feature-based proxy of the recursive definition of stationary distribution (Definition 2.3), where $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^p$ is a feature function. The stationary distribution that solves the optimization problem can be stated in closed form, from which it is possible to derive the policy:

$$\pi'(a|s) \propto \exp \pi(a|s) \left(\frac{1}{\eta} \left(r(s,a) + \int_{\mathcal{S}} p(s'|s,a) \boldsymbol{\omega}^T \boldsymbol{\phi}(s') \mathrm{d}s' - \boldsymbol{\omega}^T \boldsymbol{\phi}(s) \right) \right),$$

[5] It is worth noting that when $\alpha = 1$, D_1 is the KL-divergence and when $\alpha = \infty$, $D_\infty(\mu \| \nu) = \log \operatorname{ess\,sup}_{\mathcal{X}} \left\{ \frac{\mathrm{d}\mu}{\mathrm{d}\nu} \right\}$.

where $\eta \in [0, \infty)$ and $\boldsymbol{\omega} \in \mathbb{R}^p$ are the Lagrangian parameters that can be computed by solving the dual problem:

$$g(\eta, \boldsymbol{\omega}) = \eta \log \int_{\mathcal{S}\times\mathcal{A}} \mu^{\pi}(s,a) \exp\left(\frac{1}{\eta}\left(r(s,a) + \int_{\mathcal{S}} p(s'|s,a)\boldsymbol{\omega}^T\boldsymbol{\phi}(s')\mathrm{d}s' - \boldsymbol{\omega}^T\boldsymbol{\phi}(s)\right)\right)\mathrm{d}s\mathrm{d}a + \eta\kappa.$$

In practice, when the policy that can be played by the agent belongs to a limited parametric policy space Π_Θ, the policy π' might not be representable within Π_Θ. For this reason, we need to perform a projection onto Π_Θ. In Daniel et al. (2012), the authors suggest to perform a moment projection, i.e., find the parameterization $\boldsymbol{\theta}' \in \Theta$ that minimizes the expected KL-divergence averaged over μ:

$$\boldsymbol{\theta}' \in \arg\min_{\boldsymbol{\theta}\in\Theta} \left\{ \int_{\mathcal{S}} \mu(s) D_{\mathrm{KL}}\left(\pi'(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s)\right) \mathrm{d}s \right\}.$$

This optimization can be performed through samples, leading to a maximum likelihood estimation that requires to perform IS in order to estimate the expectation under the new distribution μ (Daniel et al., 2012).

Policy Optimization via Importance Sampling Policy Optimization via Importance Sampling (POIS, Metelli et al., 2018b, 2020b) is an actor-only off-policy policy optimization algorithm that employs IS in order to perform multiple gradient steps with the same batch of samples. The algorithm has been proposed initially in (Metelli et al., 2018b) and subsequently refined in (Metelli et al., 2020b), thanks to the introduction of the per-decision IS techniques (Precup et al., 2000). If we have at our disposal a set of trajectories $\{\tau_i\}_{i=1}^n$ sampled by running a *behavioral* policy $\pi_{\boldsymbol{\theta}}$, we can estimate the performance of a *target* policy $\pi_{\boldsymbol{\theta}'}$ by resorting to the per-decision IS estimator:

$$\widehat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{T(\tau_i)-1} \gamma^t \omega_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) R_{\tau_i, t+1},$$

where $\omega_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) = \prod_{l=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(A_{\tau_i,l}|S_{\tau_i,l})}{\pi_{\boldsymbol{\theta}}(A_{\tau_i,l}|S_{\tau_i,l})}$ is the importance weight. This estimator is unbiased and its variance can be bounded as follows (Metelli et al., 2020b, Theorem 5):[6]

$$\mathbb{V}\mathrm{ar}^{\pi_{\boldsymbol{\theta}}}\left[\widehat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] \leqslant \frac{R_{\max}^2}{n}\frac{1+\gamma}{1-\gamma}\sum_{t=0}^{T-1}\gamma^{2t} d_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}'}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}}}\right),$$

where $d_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}'}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}}}\right)$ is the Rényi divergence between the t-step trajectory distributions. Based on these results, POIS optimizes a surrogate objective function in which the estimated performance $\widehat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ is penalized by a function of its variance bound:

$$\mathcal{C}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \widehat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta}) - \zeta\sqrt{\frac{1}{n}\sum_{t=0}^{T-1}\gamma^{2t} d_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}'}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}}}\right)}.$$

This optimization is carried out performing multiple gradient steps by employing the same batch of trajectories. Then, a new batch of trajectories is collected with the obtained policy.

[6] Here we provide a slightly looser bound for the variance that, we believe, is more readable.

The hyperparameter $\zeta > 0$ can be interpreted in a probabilistic fashion by looking at the objective as a lower bound on the true performance of the target policy $J(\boldsymbol{\theta}')$ (Metelli et al., 2018b). In practice, the Rényi divergence needs to be estimated from samples as well, leading to the estimator:

$$\widehat{d}_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}'}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{l=0}^{t}\frac{\pi_{\boldsymbol{\theta}'}(A_{\tau_i,l}|S_{\tau_i,l})}{\pi_{\boldsymbol{\theta}}(A_{\tau_i,l}|S_{\tau_i,l})}\right)^2.$$

Part I

Modeling Environment Configurability

CHAPTER 4

Configurable Markov Decision Processes

4.1 Introduction

In the previous chapters, we introduced the notion of Markov decision process as a mathematical formalism to model sequential decision-making problems under uncertainty in which a goal-directed agent interacts with an environment. There are two fundamental properties that the environment is typically assumed to satisfy: its dynamics is fixed and stationary, i.e., it does not change over time either as an effect of a natural process (e.g., a non-stationary environment) or as a consequence of some external intervention (e.g., some other party altering the transition probabilities). Clearly, this assumption, together with the other considered, such as the Markov property, are reasonable in a wide range of real-world applications and they are particularly convenient (from a theoretical point of view) to state the notion of optimality and assess the existence of optimal policies (Puterman, 2014).

Several exceptions to this scenario can be found in the literature. For instance, Markov decision processes with *imprecise probabilities* (e.g., Satia and Jr., 1973; Givan et al., 1997; Trevizan et al., 2007) represent an extension of the classical MDP model in which a form of ambiguity is admitted on the transition probabilities, modeled by means of an uncertainty set. Although these works mainly focus on the modelization, the notions of optimality, typically, are derived from the robust control literature (Nilim and Ghaoui, 2003; Iyengar, 2005), with the goal of finding a policy that maximizes utility under the worst possible transition model. Another line of research, in which modifications of the transition model occur through time, are the *non-stationary* MDPs (e.g., Bowerman, 1974; Hopp et al., 1987; Garcia and Smith, 2000). In these scenarios, the transition probabili-

ties, and possibly also the reward function, change over time as an effect of the intrinsic evolution of the environment.

Although the environment is no longer fixed, these models do not account for the possibility to dynamically alter the environmental parameters. However, we can imagine scenarios in which the environment modification is an effect of a *strategy* implemented by an external party with a precise goal. As mentioned in Chapter 1, we call this goal-directed process of modification *environment configuration* (Metelli et al., 2018a). We find an example of this intentional point of view in the game-theoretic interpretation of the objective functions employed in robust MDPs. Indeed, finding a robust optimal policy can be seen as solving a zero-sum game (Shapley, 1953) in which one agent acts on the policy with the goal of maximizing the expected return and the adversary acts on the transition model with the opposite goal (Nilim and Ghaoui, 2003). In this example, the intentional way of selecting the environment emerges beyond the specific modelization of uncertainty.

In this dissertation, we study the environment configuration as the process of changing some parameters of the environment, having an effect on the transition probabilities. This chapter is devoted to the presentation of the Configurable Markov Decision Processes (Conf-MDPs) introduced in Metelli et al. (2018a) in its various aspects.

Chapter Outline The chapter is organized as follows. We start in Section 4.2 providing an informal introduction to Conf-MDPs together with some motivational examples. Then, we formally define the Conf-MDP in Section 4.3. We proceed by introducing the value functions for the Conf-MDPs (Section 4.4) and the corresponding Bellman operators and equations (Section 4.5). Then, in Section 4.6, we provide a taxonomy of the various scenarios that arise when considering Conf-MDPs. We conclude in Section 4.7 with a survey of the literature connected with Conf-MDPs.

4.2 Motivations and Examples

Environment configuration might be performed in different ways, by different parties, and with different goals. A prime scenario of environment configuration is what we call the *cooperative setting*. Intuitively, in the cooperative setting, the process of environment configuration is "functional" (auxiliary) to the agent, i.e., it is directed to improve its learning experience. In turn, we can refine the interpretation by proposing two alternative views. First, we can look at environment configuration in a *static* way, where the goal is to find the environment that allows the agent to achieve the best performance possible at the end of the learning process. In other words, we select the best MDP to solve for the agent. In this setting, policy learning and environment configuration can be, in principle, viewed at the same level. However, it is not infrequent that modifying the environment is an activity to be performed carefully, maybe less frequently than policy updating and that might generate additional costs (computational or economical). Second, environment configuration can be seen in a *dynamic* manner, as a way of speeding up the learning process. Here the goal consists in finding the sequence of configurations that allows the agent to reach an optimal policy in the original environment as fast as possible. In this sense, environment configuration can be interpreted as a form of *curriculum learning* (Bengio et al., 2009; Ciosek and Whiteson, 2017; Florensa et al., 2017), although in curriculum learning the environment modification is typically simulated, while the underlying environment dy-

namic remains unchanged. To have a more clear idea of the opportunities of environment configuration, consider the following example.

Example 4.1 (F1 Driving)**.** *Suppose an F1 driver has to learn how to drive an F1 car. The environment is composed of the car, the road and governed by the physical laws that explain the functioning of the car and the interaction with the road. The driver, the agent in this process, has at their disposal a number of possible vehicle configurations they can act on: the kind of tires, the stability and the vehicle attitude, the engine model, and the wing orientation. Being the car part of the environment, we have a scenario in which it is possible to alter a part of the environment, i.e., some parameters of the vehicle, while other portions of the environment, like the road and the physical laws must remain fixed. It is worth noting that the environment configuration has a double purpose in this setting. First, we want to find the car configuration that is best suited to the driver (static). In this case, the configuration process can be carried out by the driver themself or by an external* configurator *entity, like a track engineer. Second, we might decide to train the driver making them try different vehicle configurations, maybe of increasing degree of "difficulty", to make the driver learning an optimal policy as fast as possible. In this second scenario, the presence of an external configurator in charge of selecting the sequence of vehicle settings is unavoidable.*

From the example, it emerges that the active entity in the configuration process might be the agent itself or an external supervisor/configurator guiding the learning process (e.g., the track engineer). The idea of *supervision* as a way of constraining the actions of an agent to induce the desired behavior has been previously introduced in the field of situation calculus (Giacomo et al., 2012; Banihashemi et al., 2016, 2018).

Another interesting aspect is that, in the cooperative setting, the environment configuration should be carefully performed and customized to the specific agent. Different agents might have different abilities, modeled, in the RL framework, as different perception and actuation possibilities. For this reason, even under the same objectives, the performance of a configuration is tightly related to the agent's capabilities. Therefore, the configurator has to be aware of the agent's policy space in order to wisely identify the configuration. The following example tries to clear this aspect.

Example 4.2 (Teacher-Student)**.** *Consider a student, representing the agent, interacting with an automatic teaching system, the environment. Different students have different learning abilities. Therefore, to maximize the knowledge acquired by the student, the teaching model should be tailored to the student needs. For instance, some students prefer a straight presentation of the theory and then dive into the examples. Instead, for other students starting a topic with an example and then moving to formalization is more effective. Other tools the teaching system can leverage are the kind of material employed to introduce the topics (e.g., pictures, plots, and videos). All these choices can be thought as environment configurations having effects on the transition probabilities that govern the student's learning process. Ultimately, the optimal choice of the teaching system configuration should be aware of the student's capabilities.*

Up to now, we have considered the setting (cooperative) in which, in some high-level sense, the agent and the configurator, whenever present, pursue the same goal, i.e., improving the agent learning experience, either by quicken the learning process or identifying the

most convenient MDP to solve. However, we can think of scenarios in which the goals of configurator and agent are *non-cooperative*. In these cases, the agent learns based on its own reward function, while the configurator aims at fulfilling a possibly different goal. Clearly, in these contexts the presence of a configurator party is essential. From a static perspective, we can look at the configurator as another agent with a different reward function. Instead, from a dynamic view point, the configurator might be interested in altering the environment to induce a certain learning behavior in the agent. The following example represents a case of these settings.

Example 4.3 (Supermarket)**.** *Consider the placement of products on the shelves of a supermarket. The supermarket director, or the persons in charge, should decide the product placement in order to, from an intuitive sense, maximize the supermarket profit. Simplifying, the supermarket, that is in charge of the configuration, might decide to act so that to maximize the amount of money spent by its customers. It is reasonable to assume that the customers, representing the agents in this setting, have different goals. For instance, a customer might be interested in minimizing the time needed to complete their shopping. We immediately realize that the supermarket and the customer objectives are different, probably not fully competitive, but also not fully cooperative. Moreover, we can assume that the agent, the customer, is unaware of the strategic behavior of the configurator (or tends to act not accounting for it).*

In the non-cooperative setting, it is important to understand the kind of interaction taking place between the agent and the configurator. Since the two entities act following different objectives, a way of addressing this scenario is to take inspiration from game-theoretic tools, in order to define an appropriate *solution concept*. A first possible situation is when the agent is *unaware* of the presence of the supervisor. In such a case, the configurator selects a configuration and the agent perceives the modification of the environment as a simple non-stationarity. Therefore, importing the game-theoretic terminology, the agent is a *best responder* that learns its optimal policy, which might change over time since the environment evolves, but without further strategic behavior. On the other hand, the configurator is, of course, aware of the agent's presence. This kind of interaction, thus, can be effectively modeled as a *leader-follower game* (Shapley, 1953), where the configurator being the leader and the agent being the follower. A reasonable solution concept is the Stackelberg equilibrium (Von Stackelberg, 1934) that corresponds to the configurator selecting the configuration that maximizes its performance under the agent's optimal policy, induced by that configuration. A different perspective, that positions the agent and the configurator on the same lever, is when the agent is *aware* of the presence of the configurator. Although we believe that this situation fits less to the real-world scenarios of interests in which environment configuration is interesting, it is worth looking at the type of agent-configurator interaction. The strategic interaction between the two entities is more visible and a suitable solution concept is the *Nash equilibrium* (Nash, 1951), in which neither the agent nor the configurator has individual interest to deviate from the equilibrium strategy. A particular case of this setting is when the agent and the configurator have perfectly competitive objectives, i.e., they play a zero-sum game. This is precisely the case that is considered in robust control literature.

Example 4.4 (Robust Control)**.** *As we already mentioned, in robust control (Nilim and Ghaoui, 2003) we seek the policy that maximizes the expected return under the worst*

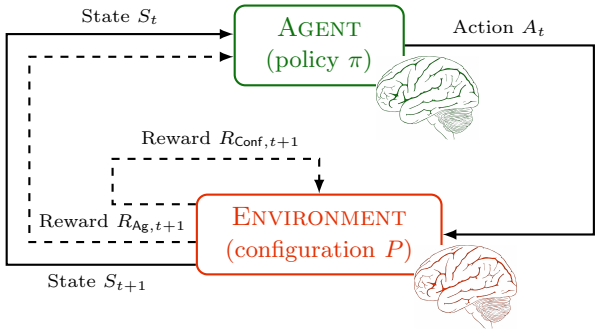


Figure 4.1: *Graphical representation of the interaction between an agent and an environment in a Conf-MDP.*

possible transition model (and possibly reward function) within the uncertainty set. *We can look at this process as the interaction between the agent learning the policy, that seeks to maximize the expected return, and a configurator selecting the transition model, that tries to minimize the expected return. The maximin solution typically employed corresponds to the Nash and to the Stackelberg equilibria of the game.*

It is worth noting that in all examples presented, the configuration activity is limited to a portion of the environment, having limited effect on a limited part of the transition probabilities. This represents an important asymmetry of environment configuration compared to policy learning. Although there exist situations, like industrial applications, in which the policy space accessible to the agent has to be limited (e.g., for safety reasons), in a large number of applications it is reasonable to consider the full space of Markovian stationary policies for policy learning. Instead, environment configuration is typically more constrained and the arbitrary alteration of the transition dynamics usually makes no sense, especially in scenarios involving natural phenomena in which the physical laws are clearly fixed. For this reason, it is common to restrict the power of the configurator to a set of configuration parameters that, indirectly, affect in a controlled manner the transition probabilities.

4.3 Definition

As we introduced in the previous section, a Conf-MDP can be thought of as an MDP in which it is possible to configure some environmental parameters, having the effect of altering the transition probabilities. To account for the presence of a configurator we consider two reward functions, one modeling the agent's goal and one for the configurator. The following definition formalizes the notion of Conf-MDP.

Definition 4.1 (Configurable Markov Decision Process)**.** *A discrete-time infinite-horizon discounted Configurable Markov Decision Process (Conf-MDP) is defined as a 6-tuple* $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R_{\mathsf{Ag}}, R_{\mathsf{Conf}}, \gamma)$ *where:*

- $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$ *is a non-empty measurable space called* state space*;*

- $(\mathcal{A}, \mathfrak{F}_{\mathcal{A}})$ *is a non-empty measurable space called* action space*;*
- $\mu_0 \in \mathscr{P}(\mathcal{S})$ *is the* initial state distribution *over the measurable space* $(\mathcal{S}, \mathfrak{F}_{\mathcal{S}})$*;*
- $R_{\mathsf{Ag}}, R_{\mathsf{Conf}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathscr{P}(\mathbb{R})$ *are the* agent *and* configurator reward models *respectively, that for every state-action-state triple* $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ *assign a probability measure* $R_{\mathsf{Ag}}(\cdot|s, a, s')$ *and* $R_{\mathsf{Conf}}(\cdot|s, a, s')$ *over the measurable space* $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$*;*
- $\gamma \in [0, 1]$ *is the* discount factor.

Thus, a Conf-MDP is obtained by removing from the definition of the MDP (Definition 2.1) the transition model P and introducing two reward functions: the agent R_{Ag} and the configurator R_{Conf} reward functions. Compared to the original definition of Conf-MDP (Metelli et al., 2018a) there are essentially two differences. First, we consider different reward functions for the agent and the configurator to model situations that were not considered in Metelli et al. (2018a), in which agent and configurator might have different, possibly conflicting, objectives. Second, we do not include the transition model space and the policy space in the definition of Conf-MDP. A graphical representation of the interaction between agent and environment in a Conf-MDP is reported in Figure 4.1.

Similarly to the case of MDPs, we introduce the *agent* and *configurator reward functions* $r_{\mathsf{Ag}}, r_{\mathsf{Conf}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, defined for every triple $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ as:

$$r_{\mathsf{Ag}}(s, a, s') = \int_{\mathbb{R}} r R_{\mathsf{Ag}}(\mathrm{d}r|s, a, s'),$$
$$r_{\mathsf{Conf}}(s, a, s') = \int_{\mathbb{R}} r R_{\mathsf{Conf}}(\mathrm{d}r|s, a, s').$$

Whenever necessary, we will assume that both r_{Ag} and r_{Conf} are uniformly bounded.

Assumption 4.1 (Uniformly Bounded Reward)**.** *The agent and configurator reward functions are uniformly bounded, i.e., there exists a finite constant* $R_{\max} \in \mathbb{R}_{>0}$ *such that:*

$$\|r_{\mathsf{Ag}}\|_\infty = \sup_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} \{|r_{\mathsf{Ag}}(s, a, s')|\} \leqslant R_{\max},$$
$$\|r_{\mathsf{Conf}}\|_\infty = \sup_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} \{|r_{\mathsf{Conf}}(s, a, s')|\} \leqslant R_{\max}.$$

4.3.1 Policies and Transition Models

A Conf-MDP is characterized by the presence of two entities: the agent and the configurator, that are in charge of performing different tasks in the model. The agent is in charge of selecting a policy, that is defined exactly as in the case of MDPs (Definition 2.2), so that to maximize the long-term reward generated by the immediate reward R_{Ag}. Instead, the configurator has the goal of selecting a transition model with the purpose of maximizing the long-term reward defined through the immediate reward R_{Conf}. Similarly to Definition 2.2, we provide the following general definition for the transition model.

Definition 4.2 (History-dependent Transition Model)**.** *A history-dependent transition model is a sequence $P = (P_t)_{t\in\mathbb{N}}$ of functions $P_t : \mathcal{H}_{\mathcal{A},t} \to \mathscr{P}(\mathcal{S})$ that for every decision step $t \in \mathbb{N}$ and for every action-ending history $h_t \in \mathcal{H}_{\mathcal{A},t}$ of length t provide a probability measure $P_t(\cdot|h_t)$ over the state space $\mathcal{S}$. We denote with $\mathcal{P}^{\mathrm{HR}}$ the set of history-dependent transition models.*

This general definition requires the transition model to select the next state based on a history $\tau = (s_0, a_0, r_1, \ldots, s_{t-1}, a_{t-1}, r_t, s_t, a_t)$ of length $t \in \mathbb{N}$. If the distribution of the next state depends on the current state-action pair (s_t, a_t) only, the transition model is called Markovian and abbreviated with $P_t(\cdot|s_t, a_t)$. Moreover, if the transition model does not depend explicitly on time, it is called stationary and, in such a case, we remove the subscript, simply writing $P(\cdot|s, a)$. We denote with $\mathcal{P}^{\mathrm{SR}}$ the set of Markovian stationary transition models. Whenever necessary, we assume that $P(\cdot|s, a)$ admits a probability density function that we denote with $p(s'|s, a)$ for every $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Finally, if for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ the transition model provides a probability to a single state, we call it deterministic. With little abuse of notation, we indicate the transition model $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ as a mapping from state-action pairs to next states, where $P(s, a)$ is the next state reached from playing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. We denote with $\mathcal{P}^{\mathrm{SD}} = \mathcal{S}^{\mathcal{S}\times\mathcal{A}}$ the set of Markovian stationary deterministic transition models. Whenever not differently specified, we will employ the term "transition model" (or simply "model") to denote a Markovian stationary transition model.

All definitions provided in Chapter 3, can be reused for the case of Conf-MDPs. Specifically, to highlight the dependence on the transition model P (that can be changed in a Conf-MDP) we will explicitly report it. For instance, $\mu_t^{\pi,P}$ is the t-step distribution, $\mu_\gamma^{\pi,P}$ is the γ-discounted stationary distribution, and $\mathbb{P}_{\mathsf{Ag}}^{\pi,P}$ (resp. $\mathbb{P}_{\mathsf{Conf}}^{\pi,P}$) is the distribution over infinite-length trajectories, induced by the policy-model pair $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ and the agent's reward model R_{Ag} (resp. the configurator's reward model R_{Conf}).

Moreover, we employ the following abbreviated notation for expectations of a bounded measurable function $f \in \mathscr{B}(\mathcal{T})$ taken w.r.t. infinite-length trajectories by employing the agent's and the configurator's reward functions respectively:

$$\mathbb{E}_{\mathsf{Ag}}^{\pi,P}[f(\tau)] \coloneqq \mathop{\mathbb{E}}_{\tau\sim\mathbb{P}_{\mathsf{Ag}}^{\pi,P}}[f(\tau)] = \int_{\mathcal{T}} \mathbb{P}_{\mathsf{Ag}}^{\pi,P}(\mathrm{d}\tau) f(\tau),$$

$$\mathbb{E}_{\mathsf{Conf}}^{\pi,P}[f(\tau)] \coloneqq \mathop{\mathbb{E}}_{\tau\sim\mathbb{P}_{\mathsf{Conf}}^{\pi,P}}[f(\tau)] = \int_{\mathcal{T}} \mathbb{P}_{\mathsf{Conf}}^{\pi,P}(\mathrm{d}\tau) f(\tau).$$

4.4 Value Functions

The notion of *value function* (Sutton and Barto, 2018) can be freely employed in the context of Conf-MDPs, with the straightforward notational adaptations. For a Conf-MDP, we have to distinguish between the agent and the configurator value functions. Specifically, a value function provides a mapping based on the choice of the initial state $(\mathcal{C}, \pi, P, s) \mapsto V_{\mathsf{Ag},\mathcal{C}}^{\pi,P}(s)$ (or state-action pair $(\mathcal{C}, \pi, P, (s, a)) \mapsto Q_{\mathsf{Ag},\mathcal{C}}^{\pi,P}(s, a)$) that are defined as in Section 2.6 but in terms of the agent's reward function R_{Ag}. Analogously, for the configurator, we employ the reward function R_{Conf}, and we denote $(\mathcal{C}, \pi, P, s) \mapsto$

$V^{\pi,P}_{\mathsf{Conf},\mathcal{C}}(s)$ (or state-action pair $(\mathcal{C},\pi,P,(s,a)) \mapsto Q^{\pi,P}_{\mathsf{Conf},\mathcal{C}}(s,a)$). Similarly, we drop the subscript $\mathcal{C}$, whenever clear from the context. For the Conf-MDPs, it is convenient to introduce a new value function that associates the performance index to a state-action-next-state triple: $(\mathcal{C},\pi,P,(s,a,s')) \mapsto U^{\pi,P}_{\mathsf{Ag},\mathcal{C}}(s,a,s')$ for the agent and $(\mathcal{C},\pi,P,(s,a,s')) \mapsto U^{\pi,P}_{\mathsf{Conf},\mathcal{C}}(s,a,s')$ for the configurator. We formally define it in the following.

Definition 4.3 (State-Action-Next-State Value Function or U-function)**.** *Let $\mathcal{C}$ be a Conf-MDP, $\pi \in \Pi^{\mathrm{SR}}$ be a policy, and $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model. The* state-action-next-state value function $U^{\pi,P}_{\mathsf{Ag}}, U^{\pi,P}_{\mathsf{Conf}} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ *are defined for every state-action-state triple $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ as the expected return starting from state s, playing action a, landing to state s', and following policy π thereafter:*

$$U^{\pi,P}_{\mathsf{Ag}}(s,a,s') = \mathbb{E}^{\pi,P}_{\mathsf{Ag}}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1} | S_0 = s, A_0 = a, S_1 = s'\right],$$
$$U^{\pi,P}_{\mathsf{Conf}}(s,a,s') = \mathbb{E}^{\pi,P}_{\mathsf{Conf}}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1} | S_0 = s, A_0 = a, S_1 = s'\right],$$

The relationship between the U-function and the Q-function is easily highlighted since the latter can be obtained as the expectation of the U-function over the next-state space: $Q^{\pi,P}_{\mathsf{Ag}}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) U^{\pi,P}_{\mathsf{Ag}}(s,a,s')$ and $Q^{\pi,P}_{\mathsf{Conf}}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) U^{\pi,P}_{\mathsf{Conf}}(s,a,s')$. Furthermore, we can define the *model advantage functions* as $A^{\pi,P}_{\mathsf{Ag}}, A^{\pi,P}_{\mathsf{Conf}} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ defined for every $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ as:

$$A^{\pi,P}_{\mathsf{Ag}}(s,a,s') = U^{\pi,P}_{\mathsf{Ag}}(s,a,s') - Q^{\pi,P}_{\mathsf{Ag}}(s,a),$$
$$A^{\pi,P}_{\mathsf{Conf}}(s,a,s') = U^{\pi,P}_{\mathsf{Conf}}(s,a,s') - Q^{\pi,P}_{\mathsf{Conf}}(s,a).$$

They quantify the performance gain obtained by selecting the next state s' when having played action a in state s compared to executing the transition model $P(\cdot|s,a)$. They are the equivalent of the *policy advantage functions* defined in Equation (2.6), that here we denote as $A^{\pi,P}_{\mathsf{Ag}}(s,a) = Q^{\pi,P}_{\mathsf{Ag}}(s,a) - V^{\pi,P}_{\mathsf{Ag}}(s)$ and $A^{\pi,P}_{\mathsf{Conf}}(s,a) = Q^{\pi,P}_{\mathsf{Conf}}(s,a) - V^{\pi,P}_{\mathsf{Conf}}(s)$ to highlight the dependence on the transition model P. We can combine the model and the policy advantage functions to get the *coupled advantage functions* $\widetilde{A}^{\pi,P}_{\mathsf{Ag}}, \widetilde{A}^{\pi,P}_{\mathsf{Conf}} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ defined for every $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ as:

$$\begin{aligned}\widetilde{A}^{\pi,P}_{\mathsf{Ag}}(s,a,s') &= U^{\pi,P}_{\mathsf{Ag}}(s,a,s') - V^{\pi,P}_{\mathsf{Ag}}(s)\\ &= A^{\pi,P}_{\mathsf{Ag}}(s,a,s') + A^{\pi,P}_{\mathsf{Ag}}(s,a),\end{aligned}$$

$$\begin{aligned}\widetilde{A}^{\pi,P}_{\mathsf{Conf}}(s,a,s') &= U^{\pi,P}_{\mathsf{Conf}}(s,a,s') - V^{\pi,P}_{\mathsf{Conf}}(s)\\ &= A^{\pi,P}_{\mathsf{Conf}}(s,a,s') + A^{\pi,P}_{\mathsf{Conf}}(s,a).\end{aligned}$$

The coupled advantage function models the gains experienced in selecting action a and next state s' from state s instead of playing policy $\pi(\cdot|s)$ and transition model $P(\cdot|s,a)$. They essentially combine the policy advantage functions and the model advantage functions to quantify their joint effect.

4.5 Bellman Equations and Operators

Similarly to the case of traditional MDPs, it is possible to rephrase the value functions in terms of the Bellman equations and introducing suitable Bellman operators. Concerning the V-function and the Q-functions the Bellman expectation operators are precisely those introduced in Section 2.6.1 instanced with the suitable immediate reward functions (r_{Ag} for the agent and r_{Conf} for the configurator). To highlight this difference and the dependence on the transition model P, we will denote them with $T^{\pi,P}_{\mathsf{Ag}}$ and $T^{\pi,P}_{\mathsf{Conf}}$ for the agent and the configurator respectively. For the U-function, instead, we provide the explicit definition of the corresponding operator.

Definition 4.4 (Bellman Expectation Operators)**.** *Let $\mathcal{C}$ be a Conf-MDP, $\pi \in \Pi^{\mathrm{SR}}$ be a policy and $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model. The* Bellman expectation operators *for the state-action-next-state value function $T^{\pi,P}_{\mathsf{Ag}}, T^{\pi,P}_{\mathsf{Conf}} : \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S}) \to \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ are defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ and every state-action-state triple $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ as:*

$$\left(T^{\pi,P}_{\mathsf{Ag}} f\right)(s,a,s') = r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \int_{\mathcal{S}} P(\mathrm{d}s''|s,a) f(s',a',s''),$$
$$\left(T^{\pi,P}_{\mathsf{Conf}} f\right)(s,a,s') = r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \int_{\mathcal{S}} P(\mathrm{d}s''|s,a) f(s',a',s'').$$

It is immediate to prove that $T^{\pi,P}_{\mathsf{Ag}}$ and $T^{\pi,P}_{\mathsf{Conf}}$ are γ-contractions in the L_∞-norm and, consequently, they admit unique fixed points that are the corresponding U-function.

Proposition 4.1. *Let $T^{\pi,P}_{\mathsf{Ag}}, T^{\pi,P}_{\mathsf{Conf}} : \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S}) \to \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ be the operators as in Definition 4.4. Then, if $\gamma \in [0,1)$ they are a γ-contraction in the L_∞-norm, i.e., for every bounded measurable function $f, g \in \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ we have:*

$$\left\| T^{\pi,P}_{\mathsf{Ag}} f - T^{\pi,P}_{\mathsf{Ag}} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty,$$
$$\left\| T^{\pi,P}_{\mathsf{Conf}} f - T^{\pi,P}_{\mathsf{Conf}} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty.$$

Furthermore, $U^{\pi,P}_{\mathsf{Ag}}$ and $U^{\pi,P}_{\mathsf{Conf}}$ are their unique fixed points, i.e., they satisfy the following Bellman equations:

$$U^{\pi,P}_{\mathsf{Ag}} = T^{\pi,P}_{\mathsf{Ag}} U^{\pi,P}_{\mathsf{Ag}},$$
$$U^{\pi,P}_{\mathsf{Conf}} = T^{\pi,P}_{\mathsf{Conf}} U^{\pi,P}_{\mathsf{Conf}}.$$

Proof. We prove the statement for the agent case only, as the configurator counterpart is analogous. Let $f, g \in \mathscr{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ and $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, we have:

$$\begin{aligned}
&\left| \left(T^{\pi,P}_{\mathsf{Ag}} f\right)(s,a,s') - \left(T^{\pi,P}_{\mathsf{Ag}} g\right)(s,a,s') \right| \\
&\qquad = \Bigg| r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \int_{\mathcal{S}} P(\mathrm{d}s''|s,a) f(s',a',s'') \\
&\qquad\quad - r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \int_{\mathcal{S}} P(\mathrm{d}s''|s,a) g(s',a',s'') \Bigg|
\end{aligned}$$

$$
\begin{aligned}
&= \gamma \left| \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \int_{\mathcal{S}} P(\mathrm{d}s''|s,a) \left(f(s',a',s'') - g(s',a',s'') \right) \right| \\
&= \gamma \sup_{s' \in \mathcal{S}, a'' \in \mathcal{A}, s'' \in \mathcal{S}} \left\{ \left| f(s',a'',s'') - g(s',a'',s'') \right| \right\} \\
&= \gamma \left\| f - g \right\|_\infty .
\end{aligned}
$$

Thus, by applying the supremum on the left hand side, we obtain:

$$
\begin{aligned}
\left\| T^{\pi,P}_{\mathsf{Ag}} f - T^{\pi,P}_{\mathsf{Ag}} g \right\|_\infty &= \sup_{s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ \left| \left(T^{\pi,P}_{\mathsf{Ag}} f \right)(s,a,s') - \left(T^{\pi,P}_{\mathsf{Ag}} g \right)(s,a,s') \right| \right\} \\
&\leqslant \gamma \left\| f - g \right\|_\infty .
\end{aligned}
$$

Since $\mathscr{B}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ is the set of bounded measurable functions having image in $\mathbb{R}$, it is a complete metric space w.r.t. the metric induced by the L_∞-norm. Thus, we can apply the Banach fixed point theorem (Banach, 1922) showing that $T^{\pi,P}_{\mathsf{Ag}}$ has a unique fixed point. It is straightforward from Definition 4.3 to prove that $U^{\pi,P}_{\mathsf{Ag}}$ is a fixed point of $T^{\pi,P}_{\mathsf{Ag}}$. □

Table 4.1 reports the value functions, the corresponding Bellman expectation operators and equations for the Conf-MDPs.

4.6 Taxonomy

At the beginning of the chapter we provided a series of motivational examples showing heterogeneous features of environment configuration. In this section, we propose an informal taxonomy of the problems that can be addressed using Conf-MDPs, based on four dimensions of classification.

Cooperative vs Non-Cooperative The first distinction is based on the agent and configurator's reward functions. If they share the same reward function, we say that we are in a *cooperative setting* in which agent and configurator act on different elements, the policy and the transition model respectively, with the goal of finding a policy-transition model pair that maximizes the long-term reward. Instead, if the reward functions are different, we are in a *non-cooperative* setting. Each of the actors attempts to optimize its own reward function. When the reward functions are opposite, we are in a fully competitive scenario that can be thought of as a zero-sum game.

Number of Agents In principle, there can be multiple configurators as well as multiple agents. For the sake of this dissertation, we restrict our attention to the case of a single configurator and a single agent. The distinction between the two entities is essential in the non-cooperative setting, while they can collapse into a unique entity in the cooperative setting. In such a case, we assume that the agent has additional capabilities for acting on the environment configuration.

Awareness In general, it is reasonable to assume that the entity entitled to the environment configuration is aware of the presence of the agent. Instead, the agent might not be aware of the presence of the supervisor. In this case, the environment modifications are perceived by the agent as a form of non-stationarity and, consequently, the strategic behavior is limited to the configurator, while the agent reduces to a best responder player. Instead,

	Agent	Configurator
V-function	$V_{\mathsf{Ag}}^{\pi,P}(s) = \mathbb{E}_{\mathsf{Ag}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s\right]$	$V_{\mathsf{Conf}}^{\pi,P}(s) = \mathbb{E}_{\mathsf{Conf}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s\right]$
	$V_{\mathsf{Ag}}^{\pi,P}(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a \mid s) \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma V_{\mathsf{Ag}}^{\pi,P}(s')\right)$	$V_{\mathsf{Conf}}^{\pi,P}(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a \mid s) \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma V_{\mathsf{Conf}}^{\pi,P}(s')\right)$
	$\left(T_{\mathsf{Ag}}^{\pi,P} f\right)(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a \mid s) \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma f(s')\right)$	$\left(T_{\mathsf{Conf}}^{\pi,P} f\right)(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a \mid s) \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma f(s')\right)$
Q-function	$Q_{\mathsf{Ag}}^{\pi,P}(s,a) = \mathbb{E}_{\mathsf{Ag}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s, A_0 = a\right]$	$Q_{\mathsf{Conf}}^{\pi,P}(s,a) = \mathbb{E}_{\mathsf{Conf}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s, A_0 = a\right]$
	$Q_{\mathsf{Ag}}^{\pi,P}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') Q_{\mathsf{Ag}}^{\pi,P}(s',a')\right)$	$Q_{\mathsf{Conf}}^{\pi,P}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') Q_{\mathsf{Conf}}^{\pi,P}(s',a')\right)$
	$\left(T_{\mathsf{Ag}}^{\pi,P} f\right)(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') f(s',a')\right)$	$\left(T_{\mathsf{Conf}}^{\pi,P} f\right)(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s' \mid s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') f(s',a')\right)$
U-function	$U_{\mathsf{Ag}}^{\pi,P}(s,a,s') = \mathbb{E}_{\mathsf{Ag}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s, A_0 = a, S_1 = s'\right]$	$U_{\mathsf{Conf}}^{\pi,P}(s,a,s') = \mathbb{E}_{\mathsf{Conf}}^{\pi,P}\left[\sum_{t=1}^{\infty}\gamma^t R_{t+1} \mid S_0 = s, A_0 = a, S_1 = s'\right]$
	$U_{\mathsf{Ag}}^{\pi,P}(s,a,s') = r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') \int_{\mathcal{S}} P(\mathrm{d}s'' \mid s',a') U_{\mathsf{Ag}}^{\pi,P}(s',a',s'')$	$U_{\mathsf{Conf}}^{\pi,P}(s,a,s') = r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') \int_{\mathcal{S}} P(\mathrm{d}s'' \mid s',a') U_{\mathsf{Conf}}^{\pi,P}(s',a',s'')$
	$\left(T_{\mathsf{Ag}}^{\pi,P} f\right)(s,a,s') = r_{\mathsf{Ag}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') \int_{\mathcal{S}} P(\mathrm{d}s'' \mid s',a') f(s',a',s'')$	$\left(T_{\mathsf{Conf}}^{\pi,P} f\right)(s,a,s') = r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a' \mid s') \int_{\mathcal{S}} P(\mathrm{d}s'' \mid s',a') f(s',a',s'')$

Table 4.1: *Summary of the value functions, Bellman expectation operators and Bellman expectation equations for Conf-MDPs.*

Number of agents	Setting	Rewards	Awareness	Solution Concept
1	Cooperative	$r_{\mathsf{Ag}} = r_{\mathsf{Conf}}$	-	Optimal
2	Cooperative	$r_{\mathsf{Ag}} = r_{\mathsf{Conf}}$	Yes/No	Optimal
2	Non-cooperative	$r_{\mathsf{Ag}} \neq r_{\mathsf{Conf}}$	Yes	Nash
2	Non-cooperative	$r_{\mathsf{Ag}} \neq r_{\mathsf{Conf}}$	No	Stackelberg
2	Zero-sum	$r_{\mathsf{Ag}} + r_{\mathsf{Conf}} = 0$	Yes/No	Nash $\equiv$ Stackelberg

Table 4.2: *Table summarizing the main features of the settings generated by the dimensions presented in Section 4.6.*

when the agent is aware of the configurator's presence, its behavior becomes strategic as well.

Solution Concepts In the cooperative setting, being the two reward functions equal, it is immediate to define a notion of optimality in which the policy-transition model pair jointly maximizes the expected return. In the non-cooperative setting, instead, we have to refer to game-theoretic notion of equilibrium. The choice of the solution concept has to account for the awareness the agent has on the configurator presence. In particular, if the agent is unaware of the configurator presence, we can look at the interaction as a leader-follower game and refer to the Stackelberg equilibrium. Instead, when both are aware of each other, the Nash equilibrium is a more appropriate solution concept. A particularly interesting case is when the reward functions are opposite, i.e., the interaction can be modeled as a zero-sum game. In such a case, the Nash and the Stackelberg equilibrium coincide.

Combining these dimensions generates several combinations, as illustrated in Table 4.2. Other dimensions could be considered as well. For instance, in the cooperative setting, when the configurator is present as an external entity, it might know or not the agent's reward function. If it does, then the configuration problem could be solved offline with no need for interaction. Instead, when the agent's reward is unknown to the configurator, interaction becomes essential. This distinction can be extended also to the non-cooperative setting. Nonetheless, we believe that our Conf-MDP model still misses capturing some relevant situations, especially the curriculum learning view of the configuration activity, mentioned in Section 4.1. We will discuss these issues in Chapter 10.

4.7 Related Literature

In this section, we provide a survey of the literature connected with the Conf-MDP framework. Specifically, we will focus on three macro topics.

- In Section 4.7.1, we discuss the models that are employed to represent uncertainty in the transition probabilities (Satia and Jr., 1973). These works mainly focus on modelization and are extensively employed by the robust control community (Nilim and Ghaoui, 2003).

- In Section 4.7.2, we present the models and solution concepts that are employed when the environment evolves naturally, i.e., the environmental modifications are a form of non-stationarity (Bowerman, 1974).
- In Section 4.7.3, we illustrate the models and approaches that assume the possibility to act explicitly on the environment in a strategic way (e.g., Zhang et al., 2009a; Keren et al., 2017).

4.7.1 The Environment is Known under Uncertainty

There are several real-world cases in which the environment dynamics can only be known under uncertainty. In this section, we revise this family of works. The interest in connection with our Conf-MDPs lies in the modelization techniques, accounting for multiple admissible transition models, as well as in the choice of the objective functions employed to select a suitable transition model among the admissible ones.

Markov Decision Processes with Imprecise Probabilities Markov Decision Processes with Imprecise Probabilities (MDPIPs, Satia and Jr., 1973; III and Eldeib, 1994; Bueno et al., 2017) are an extension of traditional MDPs in which the transition model is only known under uncertainty. Thus, the transition model is not expressed as a probability distribution, as in traditional MDPs, but it is specified by means of a set of probability distributions, defined for every for every $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$P(\cdot|s, a) \in \mathcal{K}(\cdot|s, a) \subseteq \mathscr{P}(\mathcal{S}),$$

$\mathcal{K}(\cdot|s, a)$ is named *transition credal sets*, also known as *uncertainty set* (Delgado et al., 2009). The applications of MDPIPs might be numerous, including studying the sensitivity of the value functions and the optimal policies under variation of the transition model and robust control. The original work (Satia and Jr., 1973) proposed two objectives: the *maximax* and *maximin*. The maximax objective seeks for the best policy and transition model in the credal sets so as to maximize the expected return, leading to the following value function defined for every $s \in \mathcal{S}$ as:

$$V^{\text{maximax}}(s) = \sup_{a \in \mathcal{A}} \sup_{P(\cdot|s,a) \in \mathcal{K}(\cdot|s,a)} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \left(r(s, a, s') + \gamma V^{\text{maximax}}(s') \right) \right\}.$$

Thus, V^{maximax} is an upper bound on the expected reward under the true model (Utkin and Augustin, 2005). Instead, the maximin criterion looks for the policy maximizing the expected return while considering the worst possible transition model, leading to the following value function defined for every $s \in \mathcal{S}$ as:

$$V^{\text{maximin}}(s) = \sup_{a \in \mathcal{A}} \inf_{P(\cdot|s,a) \in \mathcal{K}(\cdot|s,a)} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \left(r(s, a, s') + \gamma V^{\text{maximin}}(s') \right) \right\}.$$

V^{maximin} represents a lower bound on the expected return (Delgado et al., 2009) and this objective is closely related to the robust control literature (Nilim and Ghaoui, 2003). Variants of policy iteration have been proposed for solving both the problems (Satia and Jr., 1973). Other objectives can be employed, like the *maximix* objective that considers a convex combination of the maximax and maximin objectives, interval dominance, maximality, and E-admissibility (Seidenfeld, 2004; Kikuti et al., 2005).

Bounded-parameter Markov Decision Processes Bounded-parameter Markov Decision Processes (BMDPs, Givan et al., 1997; Ni and Liu, 2008) are a particular instance of MDPIPs in which the credal sets are assumed to be intervals. Specifically, a BMDP $\mathcal{M}_{\updownarrow}$ can be thought as a set of MDPs in which the transition probabilities are specified by means of lower and upper bounds on their values: $P_{\updownarrow}(\cdot|s,a) = [l(\cdot|s,a), u(\cdot|s,a)]$ where $l(\cdot|s,a) \leqslant u(\cdot|s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Additionally, BMDPs allow representing uncertainty on the reward function by means of analogous intervals $R_{\updownarrow}(\cdot|s,a)$. This allows extending the classical notion of value function, leading to the *interval value functions* defined for every state $s \in \mathcal{S}$ as:

$$V^{\pi}_{\updownarrow}(s) = \left[\inf_{\mathcal{M} \in \mathcal{M}_{\updownarrow}} \{V^{\pi}_{\mathcal{M}}(s)\}, \sup_{\mathcal{M} \in \mathcal{M}_{\updownarrow}} \{V^{\pi}_{\mathcal{M}}(s)\} \right].$$

Interval value functions can be compared by defining suitable ordering relationships on real intervals. Based on whether we employ the lower or the upper bound to sort value functions, we can define *pessimistic* and *optimistic* estimates of the true optimal value function. Besides modeling the uncertainty, BMDPs can be thought of as a way to represent an MDP obtained by means of the state aggregation of an original (primitive) MDP (Givan et al., 1997). In this way, we replace the probability of each individual transition (and the reward) with an interval. Interval policy evaluation and value iteration can be employed to analyze the sensitivity of the value function of a policy and the optimal value function to this form of aggregation (Givan et al., 1997).

Markov Decision Process with Set-valued Transition Another particularization of MDPIPs can be found in the Markov Decision Process with Set-valued Transition (MDPST, Trevizan et al., 2007, 2008). MDPSTs extend the MDP considering probability distributions over state sets, i.e., $m(\cdot|s,a) \in \mathscr{P}(F(s,a))$, where $F(s,a) \subseteq 2^{\mathcal{S}} \backslash \{\{\}\}$ for every $(s,a) \in \mathcal{S} \times \mathcal{A}$ are the reachable sets obtained by playing action a in state s. It can be proved that an MDPST induces an MDPIP where the credal sets are defined in terms of the reachable sets (Trevizan et al., 2007). It is possible to prove that the maximin criteria of MDPIP is equivalent to the following simplified objective for MDPSTs, in which the minimization over the transition model can be conveniently pushed inside the expectation:

$$V^{\text{maxmin}}(s) = \sup_{a \in \mathcal{A}} \left\{ \int_{F(s,a)} m(\mathrm{d}\mathcal{U}|s,a) \inf_{s' \in \mathcal{U}} \left\{ r(s,a,s') + \gamma V^{\text{maxmin}}(s') \right\} \right\}.$$

Robust Markov Decision Processes Up to now, we have discussed different extensions of the traditional MDP framework, all derived from the basic MDPIP model. These works are more focused on modeling uncertainty rather than the nature of the objective function employed to discriminate among the possible transition models. The *robust control* literature (Bagnell et al., 2001; Nilim and Ghaoui, 2003; Iyengar, 2005), instead, is based on the idea of learning a *robust policy*, i.e., a policy that maximizes the expected return under the worst admissible transition model. In this sense, robust control makes use of the maximin objective previously introduced. While a large part of the research effort is focused on *rectangular ambiguity sets* (Nilim and Ghaoui, 2003; Iyengar, 2005), it might

be of interest considering the possibility that the transition probabilities of different state-action pairs are related. In full generality, a transition model belonging to a subset of the mappings from state-action pairs to probability measures over the state space:

$$P \in \mathcal{K} \subseteq \{f : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S})\}.$$

Unfortunately, for general uncertainty sets it has been proven that solving the maximin problem is NP-hard (Wiesemann et al., 2013). Instead, a robust policy can be computed efficiently for specific choices of the ambiguity set. An example is s-rectangularity (Le Tallec, 2007), where the ambiguity set is defined separately for every state, modeling situations in which nature can see the last state but not the action. In this setting, it is possible to derive a robust policy in polynomial time. Similar results hold for (s, a)-rectangularity (Wiesemann et al., 2013), where for each state-action pair a separate ambiguity set is considered. This is essentially the model employed in the credal sets of MDPIP. The main difference between the two notions of rectangularity is that the optimal robust policy for (s, a)-rectangularity can be proven to be deterministic whereas for s-rectangularity the robust policy might be stochastic (Wiesemann et al., 2013). Several successive works extended both the models of uncertainty (e.g., Goyal and Grand-Clement, 2018; Mannor et al., 2016) and the objective functions considered (e.g., Delage and Mannor, 2010).

4.7.2 The Environment Changes Naturally

In the previous section, we have illustrated the formalizations that account for multiple transition models to represent a lack of knowledge. These approaches do not admit the possibility that the environment changes over time and, consequently, during the learning process. In this section, we present the modelizations and the approaches that consider a "natural" evolution of the transition probabilities over time.

Non-Stationary Markov Decision Processes A Non-Stationary Markov Decision Process (NSMDP, Bowerman, 1974) is an extension of the traditional MDP model that allows the environment dynamics and the reward function to change over time. Formally, the transition model $P = (P_t)_{t\in\mathbb{N}}$ and reward model $R = (R_t)_{t\in\mathbb{N}}$ are parametrized by the time index $t \in \mathbb{N}$. Non-stationarity can be seen as a form of *partial observability* (Kaelbling et al., 1998) since the time index can be interpreted as a state mode that is not observed by the agent. For this reason, a NSMDP can be always transformed in an MDP, by simply adding the time variable in the state space. This evolution over time is *natural*, i.e., not determined by an external, intentional, intervention. As intuition suggests, in NSMDP makes sense to consider non-stationary policies $\pi = (\pi_t)_{t\in\mathbb{N}}$. Consequently, the value functions need to be indexed by time as well. For every $t \in \mathbb{N}$, the following Bellman equation can be defined for every state $s \in \mathcal{S}$ as (Lecarpentier and Rachelson, 2019):

$$V_t^\pi(s) = \int_{\mathcal{A}} \int_{\mathcal{S}} \pi_t(\mathrm{d}a|s) P_t(\mathrm{d}s'|s,a) \left(r_t(s,a,s') + \gamma V_{t+1}^\pi(s')\right)$$

Concerning optimization, we have the following optimal Bellman equation, defined for every state $s \in \mathcal{S}$ and $t \in \mathbb{N}$ as:

$$V_t^*(s) = \sup_{a\in\mathcal{A}} \left\{ \int_{\mathcal{S}} P_t(\mathrm{d}s'|s,a) \left(r_t(s,a,s') + \gamma V_{t+1}^*(s')\right) \right\},$$

from which we can derive a greedy policy π_t^*. NSMDPs can be treated with more generality referring to the framework of Hidden-Mode Markov Decision Processes (Choi et al., 2001), a particular instance of POMDPs, in which some state modes are hidden to the decision-maker. Several works have addressed the problem of defining a suitable objective function and solution approaches (Lecarpentier and Rachelson, 2019), also in the robust control setting (Sinha and Ghate, 2016).

4.7.3 The Environment Changes Strategically

In this section, we consider the possibility that the environment transition function changes over time, not in a natural way, as in non-stationary models, but as an effect of the intervention of a strategic actor. This setting is closely related to our Conf-MDPs that assumes the presence of agent and configurator interacting with one another.

Environment Design A line of research that displays several similarities with Conf-MDPs is *environment design* (Zhang and Parkes, 2008; Zhang et al., 2009a), which was first introduced in the planning community with the *value-based policy teaching* (Zhang and Parkes, 2008). The fundamental idea is that in a learning process there may be an *interested party*, i.e., an entity different from the learning agent, that is allowed to change dynamically the reward function of the MDP, providing some incentives, to induce the agent displaying a certain behavior. The formalization of this process was provided with more generality in Zhang et al. (2009a). Specifically, an environment design problem is composed of:

- an environment $e \in \mathcal{E}$;
- an agent model (θ, f), where $\theta \in \mathcal{I}$ are the model parameters that represent the agent's preferences and capabilities and $f : \mathcal{I} \times \mathcal{E} \to \mathcal{X}$ is the agent function mapping a parameter and an environment to a decision, in the decision set $\mathcal{X}$;
- the interested party knows the environment e and the agent function f and can act by means of an *environment change* $\Delta \in \mathbf{\Delta}$. Based on the current environment $e \in \mathcal{E}$, agent's decision $x \in \mathcal{X}$, an environment change $\Delta \in \mathbf{\Delta}$ can be *admissible* if it belongs to the set $\Delta \in \mathrm{admissible}(e, x)$;
- the environment transition function $\mathcal{F} : \mathcal{E} \times \mathbf{\Delta} \to \mathcal{E}$ that, given the current environment and the environment change, provides the modified environment. The environment transition function is assumed to be known to the interested party;
- the goal function $\mathcal{G} : \mathcal{X} \times \mathbf{\Delta} \to \mathbb{R}$ that outputs the reward of the interested party.[1]

Therefore, in this model the interested party aims at finding the admissible environment change $\Delta \in \mathbf{\Delta}$ such that the agent's behavior in the modified environment $e' = \mathcal{F}(e, \Delta)$ optimizes the goal function $\mathcal{G}$:

$$\max_{\Delta \in \mathbf{\Delta}} \quad \mathcal{G}\left(x, \Delta\right)$$

[1] In Zhang et al. (2009a) a slightly more general definition is provided, in which the agent function f is admitted to output a set of decisions rather than a single decision and the agent function $\mathcal{G}$ depends on $\mathcal{I}$ and $\mathcal{E}$ too.

$$\begin{aligned} \text{s.t.} \quad & \Delta \in \text{admissible}(e, x) \\ & e' = \mathcal{F}(e, \Delta) \\ & x = f(\theta, e') \end{aligned}$$

It was proven that under certain convenient forms of the agent function the problem is tractable and can be addressed using LP (Zhang et al., 2009a).

We can identify several similarities between this formulation of environment design and the Conf-MDPs. Indeed, if we look at the agent's function f as the agent's best response function, that is assumed to be known to the interested party (the configurator in our setting), this formulation resembles a form of leader-follower game taking place between agent and configurator. Indeed, the interested party (the configurator in our terminology) seeks for the environment change (analogous to the environment configuration) that optimizes its utility function $\mathcal{G}$ (the expected return for us) assuming that the agent will react as a best responder. However, the main limitation, we believe, is the assumption that the interested part knows the agent function. Subsequent works considered the setting in which the interested part objective consists of teaching a specific policy to the agent (Zhang et al., 2009b) or optimizing an environment tailored to the user needs by selecting online the available action set (Mahmud et al., 2014).

Utility Maximizing Design A particular instance of environment design is represented by Equi-Reward Utility Maximizing Design (ER-UMD, Keren et al., 2017). In ER-UMD the agent and the interested party share the same goal and the formulation is restricted to the MDP case. In this sense, ER-UMD resembles the cooperative view of Conf-MDPs. Specifically, if we denote with $J^{*,e} = \sup_{\pi \in \Pi^{\text{SR}}} \{J^{\pi,e}\}$ the optimal agent's performance in the environment $e \in \mathcal{E}$, the interested party looks for an admissible environment change (or sequence of admissible environment changes) so that the agent's performance is maximized:

$$\begin{aligned} \max_{\Delta \in \boldsymbol{\Delta}} \quad & J^{*,e'} \\ \text{s.t.} \quad & \Delta \in \text{admissible}(e) \\ & e' = \mathcal{F}(e, \Delta). \end{aligned} \tag{4.1}$$

Additionally in Keren et al. (2017) the possibility to associate to the environment modification a cost is considered. A cost function $\mathcal{C} : \boldsymbol{\Delta} \to \mathbb{R}_{\geqslant 0}$ is employed to brake ties among the optimal changes Δ^* solving the problem in Equation (4.1) preferring those of minimal cost. Heuristic approaches have been subsequently proposed to make the search for the sequence of changes tractable (Keren et al., 2018, 2019).

Cost-Aware Objectives A similar idea of including costs in the objective function was recently proposed in Silva et al. (2018), in a framework closer to the RL formulation. The idea consists in considering a parametric representation of the transition model $P_{\boldsymbol{\omega}}$ and evaluate the cost of every parametrization via function $C : \Omega \to \mathbb{R}_{\geqslant 0}$ leading to the problem:

$$\max_{\boldsymbol{\omega} \in \Omega} J^*(\boldsymbol{\omega}) - C(\boldsymbol{\omega}),$$

where $J^*(\boldsymbol{\omega}) = \sup_{\pi \in \Pi^{\text{SR}}} \left\{ J^{\pi, P_{\boldsymbol{\omega}}} \right\}$ is the expected return of the optimal policy in the MDP induced by the transition model $P_{\boldsymbol{\omega}}$. The problem is tackled by means of a gradient-based approach.

Threatened Markov Decision Process Threatened Markov Decision Process (TMDP, Gallego et al., 2019a,b) is a recently introduced model that assumes the presence of an opponent (a threatener) that performs a *threat action* selected within the set $\mathcal{B}$ that has an effect on the transition probabilities $P : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathscr{P}(\mathcal{S})$. Instead of tackling the problem from a game-theoretic perspective, the authors propose to augment the MDP accounting for the opponent presence, modeled with a prior belief $p_A(b|s)$ that the agent maintains on the threat action the opponent will play. A modified version of the Q-learning algorithm is proposed, in which the target value is defined in terms of an expectation w.r.t. to the belief $p_A(b|s)$:

$$Q(s, a) = \int p_A(\mathrm{d}b|s) Q(s, a, b).$$

In order to deal with the uncertainty on the opponent's policy p_A, the authors propose to either consider a non-strategic opponent or to employ level-k thinking mechanism (Gallego et al., 2019a).

Off-Environment Reinforcement Learning OFFER (Ciosek and Whiteson, 2017) addresses the issue of *significant rare events*, i.e., situations that occur in the environment with low probability but able to affect significantly the agent's performance. The authors propose an *off-environment* policy gradient method that, by means of a simulator, changes, during the learning process, the probability of the rare events so that the trained agent can learn to deal with them. Like the policy $\pi_{\boldsymbol{\theta}}$, the environment is parametrized $P_{\boldsymbol{\omega}}$, and the optimization on these parameters aims at minimizing the variance of the policy gradient:

$$\boldsymbol{\omega}^* \in \arg\min_{\boldsymbol{\omega} \in \Omega} \left\{ \mathbb{V}\mathrm{ar} \left[\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}_0 / \boldsymbol{\omega}) \right] \right\},$$

where $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}_0 / \boldsymbol{\omega})$ is an importance sampling estimator of the policy gradient of policy $\pi_{\boldsymbol{\theta}}$ under the true environment $P_{\boldsymbol{\omega}_0}$ having samples collected in environment $P_{\boldsymbol{\omega}}$. Since the presence of significant rare events is a source of variance, this objective tends, indirectly, to increase their probability to occur. This approach can be thought of as a form of curriculum learning (Bengio et al., 2009) but, differently from Conf-MDPs, the configuration is only simulated.

Adversarial Attacks in RL The vulnerability of deep learning classifiers to adversarial inputs is a well-known issue in image classification Chakraborty et al. (2018). More recently, this phenomenon has been studied in the field of RL. A first branch of approaches directly translates the techniques employed in image classification to RL, but they are clearly limited to deep RL architectures with a state representation based on images (Huang et al., 2017; Lin et al., 2017). In these cases, an attack is considered successful if it determines a significant worsening of the performance. More recently, the notion of *policy poisoning* (Ma et al., 2019) has been introduced. In this setting, the goal of the attacker is that of inducing the agent learning a specific policy. Such an attack can be

carried out in different modalities. A first example proposed in (Ma et al., 2019) consists in altering the reward function the agent observes. The paper proves that this intervention is sufficient to poison the policy, with no need of acting on the state-action representation. Furthermore, the possibility of acting on the transition model, in addition to the reward function, is accounted in (Rakhsha et al., 2020), where a cost of altering the environment is also quantified. These latter examples can be considered a way of operating in a non-cooperative Conf-MDP, where the configurator having the specific goal of policy poisoning. We point out that, since these approaches aim at forcing a particular policy, they cannot be directly mapped to the Conf-MDP definition, that requires a specific reward function for the configurator. Indeed, treating policy poisoning in the Conf-MDP framework would mean devising a suitable reward function inducing the desired policy under the optimal configuration.

CHAPTER 5

Solution Concepts for Configurable Markov Decision Processes

In Chapter 4, we have introduced the notion of configurable Markov decision process, as a formalism to model the (possible) presence of a configurator in charge of acting on the environmental parameters with a possibly non-cooperative goal compared to that of the agent. Furthermore, we provided a taxonomy that qualitatively classifies the settings that can emerge in the Conf-MDP framework and we discussed how they determine the most suited solution concepts. In this dissertation, we primarily focus on the cooperative setting in which a straightforward notion of optimality can be defined. The subsequent chapters, therefore, will be devoted to the study of the solution techniques for cooperative Conf-MDPs in both finite and continuous domains. In this chapter, instead, we provide the formalization of the solution concepts for the cooperative and non-cooperative settings.

Chapter Outline The chapter is organized as follows. In Section 5.1, we focus on the cooperative setting. We start by formally defining a cooperative Conf-MDP, we show how this setting can be reduced to a standard MDP, and we provide the optimality conditions, including the corresponding Bellman optimality operators and equations. In Section 5.2, we discuss the non-cooperative setting. Based on whether the agent is aware of the configurator presence, we propose solution concepts based on Nash and Stackelberg equilibria, and the corresponding value functions and operators.

5.1 Cooperative Setting

In the cooperative setting, the agent and the configurator share the same reward function, i.e., they act perusing the same objective. From a more formal point of view, we can define a *Cooperative Conf-MDP* as follows.

Definition 5.1 (Cooperative Conf-MDP)**.** *Let $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R_{\mathsf{Ag}}, R_{\mathsf{Conf}}, \gamma)$ be a Conf-MDP. $\mathcal{C}$ is a* Cooperative Conf-MDP *if for every state-action-state triple $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ it holds that:*

$$R_{\mathsf{Ag}}(\cdot|s, a, s') = R_{\mathsf{Conf}}(\cdot|s, a, s') \qquad \textit{almost surely.} \tag{5.1}$$

In this case, we will abbreviate the notation reporting just one reward function $R = R_{\mathsf{Ag}} = R_{\mathsf{Conf}}$ in the tuple $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R, \gamma)$.

This definition implies that also the reward functions are equal, i.e., $r_{\mathsf{Ag}}(s, a, s') = r_{\mathsf{Conf}}(s, a, s')$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. For notational convenience, we will remove the subscripts Ag and Conf from all the relevant quantities, as the distinction is not necessary. Specifically, we will denote the expectation of a bounded measurable function $f \in \mathscr{B}(\mathcal{T})$ under the infinite-length trajectory distribution as follows:

$$\mathbb{E}^{\pi,P}[f(\tau)] \coloneqq \mathop{\mathbb{E}}_{\tau \sim \mathbb{P}^{\pi,P}}[f(\tau)] = \int_{\mathcal{T}} \mathbb{P}^{\pi,P}(\mathrm{d}\tau) f(\tau).$$

The goal in a cooperative Conf-MDP consists in finding a policy-transition model pair that, jointly, maximize the long-term reward. This can be formalized, similarly to what is done in Chapter 2 by introducing the notion of optimal value function and optimal policy-transition model pair.

Definition 5.2 (Optimality in Conf-MDPs)**.** *Let $\mathcal{C}$ be an Conf-MDP. A policy-transition model pair $(\pi^*, P^*) \in \Pi^{\mathrm{HR}} \times \mathcal{P}^{\mathrm{HR}}$ is* optimal *if for every state $s \in \mathcal{S}$ and policy-transition model pair $(\pi, P) \in \Pi^{\mathrm{HR}} \times \mathcal{P}^{\mathrm{HR}}$ it holds that:*

$$V^{\pi^*,P^*}(s) \geqslant V^{\pi,P}(s). \tag{5.2}$$

The optimal state value function *is defined for every state $s \in \mathcal{S}$ as:*

$$V^{*,*}(s) = \sup_{\pi \in \Pi^{\mathrm{HR}}, P \in \mathcal{P}^{\mathrm{HR}}} \left\{ V^{\pi,P}(s) \right\}. \tag{5.3}$$

Before proceeding further, it is convenient to show that a cooperative Conf-MDP can be reduced to an equivalent MDP.

5.1.1 Reduction of Cooperative Conf-MDP to MDP

In this section, we show that a Cooperative Conf-MDP can be reduced to an "equivalent" MDP, as shown in the following result.

Theorem 5.1. *Let $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R, \gamma)$ be a cooperative Conf-MDP and let $\widetilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A} \times \mathcal{S}, \widetilde{P}, \mu_0, \widetilde{R}, \gamma)$ be an MDP defined as follows:*

- $\widetilde{P} : \mathcal{S} \times (\mathcal{A} \times \mathcal{S}) \to \mathscr{P}(\mathcal{S})$, *defined for every* $(s, (a, s')) \in \mathcal{S} \times (\mathcal{A} \times \mathcal{S})$ *and* $s'' \in \mathcal{S}$ *as:*
$$\widetilde{P}(\mathrm{d}s''|s, (a, s')) = \delta_{s'}(\mathrm{d}s''),$$
- $\widetilde{R} : \mathcal{S} \times (\mathcal{A} \times \mathcal{S}) \times \mathcal{S} \to \mathscr{P}(\mathbb{R})$, *defined for* $(s, (a, s'), s'') \in \mathcal{S} \times (\mathcal{A} \times \mathcal{S}) \times \mathcal{S}$ *and* $r \in \mathbb{R}$ *as:*
$$\widetilde{R}(\mathrm{d}r|s, (a, s'), s'') = R(\mathrm{d}r|s, a, s').$$

Let $(\pi, P) \in \Pi^{\mathrm{HR}} \times \mathcal{P}^{\mathrm{HR}}$ *be a policy-transition model pair for the Conf-MDP* $\mathcal{C}$*. Let us define the new policy* $\widetilde{\pi}$ *for the MDP* $\widetilde{\mathcal{M}}$ *for every* $t \in \mathbb{N}$*, state-ending history* $h_t \in \mathcal{H}_{\mathcal{S},t}$*, and* $(a, s') \in \mathcal{A} \times \mathcal{S}$ *as:*

$$\widetilde{\pi}_t(\mathrm{d}(a, s')|h_t) = \pi_t(\mathrm{d}a|h_t)P_t(\mathrm{d}s'|h_t, a).$$

Then, the value function induced by the policy-model pair $(\pi, P) \in \Pi^{\mathrm{HR}} \times \mathcal{P}^{\mathrm{HR}}$ *in the Conf-MDP* $\mathcal{C}$ *is equal to the value function induced by policy* $\widetilde{\pi}$ *in MDP* $\widetilde{\mathcal{M}}$*, i.e., for every state* $s \in \mathcal{S}$ *it holds that:*

$$V_{\mathcal{C}}^{\pi,P}(s) = V_{\widetilde{\mathcal{M}}}^{\widetilde{\pi}}(s).$$

Proof. $V_{\mathcal{C}}^{\pi,P}$ is the expectation of the return $\sum_{t=0}^{\infty} \gamma^t R_{t+1}$ under the trajectory distribution induced in $\mathcal{C}$ by the pair $(\pi, P) \in \Pi^{\mathrm{HR}} \times \mathcal{P}^{\mathrm{HR}}$, conditioned to the initial state $s \in \mathcal{S}$, denote it with $\mathbb{P}_{\mathcal{C}}$. Similarly, $V_{\widetilde{\mathcal{M}}}^{\widetilde{\pi}}$ is the expectation of the return under the trajectory distribution induced in $\widetilde{\mathcal{M}}$ by the pair $\widetilde{\pi}$, conditioned to the initial state $s \in \mathcal{S}$, denote it with $\mathbb{P}_{\widetilde{\mathcal{M}}}$. Thus, it suffices to prove that these distributions are the same. Let us consider the following derivation:

$$\begin{aligned}
\mathbb{P}_{\widetilde{\mathcal{M}}}(\mathrm{d}\tau) &= \delta_s(\mathrm{d}s_0) \prod_{t=0}^{\infty} \widetilde{\pi}(\mathrm{d}(a_t, s_t')|h_t)\widetilde{P}(\mathrm{d}s_{t+1}|s_t, (a_t, s_t'))\widetilde{R}(\mathrm{d}r_{t+1}|s_t, (a_t, s_t'), s_{t+1}) \\
&= \delta_s(\mathrm{d}s_0) \prod_{t=0}^{\infty} \pi_t(\mathrm{d}a|h_t)P_t(\mathrm{d}s_t'|h_t, a_t)\delta_{s_{t+1}}(\mathrm{d}s_t')\widetilde{R}(\mathrm{d}r_{t+1}|s_t, a_t, s_t') \\
&= \delta_s(\mathrm{d}s_0) \prod_{t=0}^{\infty} \pi_t(\mathrm{d}a|h_t)P_t(\mathrm{d}s_{t+1}|h_t, a_t)R(\mathrm{d}r_{t+1}|s_t, a_t, s_{t+1}) = \mathbb{P}_{\mathcal{C}}(\mathrm{d}\tau),
\end{aligned}$$

where we exploited the properties of the Dirac measure and the definitions of $\widetilde{P}$, $\widetilde{R}$, and $\widetilde{\pi}$. □

Intuition suggests that we can solve the equivalent MDP $\widetilde{\mathcal{M}}$ finding an optimal policy $\widetilde{\pi}^*$, that for every state prescribes both the action and the next state, and then derive an optimal policy-transition model pair (π^*, P^*). However, we do not see this as a constructive result, but just as a tool to reuse the results of the traditional MDP theory to the Conf-MDP setting.

Indeed, if the transition probabilities can be changed arbitrarily and there is no constraint on how frequently we can perform an update, we can simply translate the Conf-MDP in the equivalent MDP $\widetilde{\mathcal{M}}$. There are some reasons why this might not be convenient and sometimes impossible. First, the policy and the transition model might be under the control of different entities: the agent and the configurator, respectively. Moreover, they might perform updates at different time scales. In realistic scenarios, changing the environment might be an expensive operation (although at this level, we did not model cost) to be performed less frequently compared to policy updates. This is clearly more reasonable

in the non-cooperative setting. Even in the cooperative setting, when there is just the agent configuring the environment, it is worth noting that while it has knowledge on its policy space, it typically just knows which are the environment configurable parameters, but it ignores the effect on the transition probabilities.

Remark 5.1 (On the Complexity of solving Conf-MDPs)**.** *Thanks to the reduction provided above, solving a cooperative Conf-MDP with $|\mathcal{S}|$ states and $|\mathcal{A}|$ action is equivalent to solving an MDP with $|\mathcal{S}|$ states and $|\mathcal{A}||\mathcal{S}|$ actions. In particular, based on (Papadimitriou and Tsitsiklis, 1987), it follows that solving a Conf-MDP is for sure* P*-complete, i.e., if an efficient algorithm were available then all problems in* P *would be solvable efficiently in parallel. Actually, it was proven that solving deterministic MDPs is in* NC *(Papadimitriou and Tsitsiklis, 1987), i.e., deterministic MDPs can be solved efficiently in parallel. Since the reduction we propose generates a deterministic equivalent MDP, we can conclude that solving a cooperative Conf-MDP is in* NC*. In other words, solving a Conf-MDP is intrinsically simpler than solving an MDP.*

There are some observations that need to be discussed. We are considering the general setting in which we are allowed to change the probabilities of the transition model arbitrarily. We have already observed that this context is unrealistic in several scenarios of interest. Furthermore, the recent work (Silva et al., 2019) showed that solving a cooperative Conf-MDP is NP*-hard even when no explicit cost function is considered. This is not in contradiction with what we have stated above. Indeed, the result (Silva et al., 2019) is based on a reduction that considers a specific way in which the transition model can be modified, i.e., the search is restricted to a subset $\mathcal{P} \subset \mathcal{P}^{\mathrm{SR}}$ of the space of Markovian stationary transition models. This reflects the analogy with policy search. Indeed, when the optimization is restricted to a generic subset $\Pi \subset \Pi^{\mathrm{SR}}$ of the Markovian stationary policies the problem becomes* NP*-hard as well (Vlassis et al., 2012).*

5.1.2 Optimal Value Functions

We defined the optimal state value function $V^{*,*}$ of a cooperative Conf-MDP as the best performance we can obtain in the Conf-MDP starting from each state. Analogously, we can define the *optimal state-action value function* $Q^{*,*}$ and the *optimal state-action-next-state value function* $U^{*,*}$ defined for every $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ as:

$$Q^{*,*}(s,a) = \sup_{\pi \in \Pi^{\mathrm{HR}}, P \in \mathcal{P}^{\mathrm{HR}}} \left\{ Q^{\pi,P}(s,a) \right\},$$
$$U^{*,*}(s,a,s') = \sup_{\pi \in \Pi^{\mathrm{HR}}, P \in \mathcal{P}^{\mathrm{HR}}} \left\{ U^{\pi,P}(s,a,s') \right\}.$$

Clearly, given the reduction of Theorem 5.1 we immediacy observe that we can restrict w.l.o.g. the computation of the supremum to the space of Markovian stationary policies Π^{SR} and transition models $\mathcal{P}^{\mathrm{SR}}$. Similarly to the case of traditional MDPs, in the cooperative setting, we can define suitable Bellman optimal operators and equations. For the sake of brevity, we report those of the V-function only. We refer the reader to Table 5.1 for a complete view.

Definition 5.3 (Bellman Optimality Operator for Conf-MDPs)**.** *Let $\mathcal{C}$ be a cooperative Conf-MDP, $\pi \in \Pi^{\mathrm{SR}}$ be a policy, and $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model. The* Bellman

optimality operator *for the state value function* $T^{*,*} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ *is defined for every bounded measurable function* $f \in \mathscr{B}(\mathcal{S})$ *and every state* $s \in \mathcal{S}$ *as:*

$$\left(T^{*,*} f\right)(s) = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ r(s, a, s') + \gamma f(s') \right\}. \tag{5.4}$$

Compared to the Bellman operators defined for traditional MDPs (Definition 2.8), instead of computing the expectation w.r.t. the next state sampled from the (fixed) transition model P, we perform a maximization on both the action and the next state. Clearly, these operators are still a contraction in L_∞-norm when $\gamma < 1$.

Proposition 5.2. *Let* $T^{*,*} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ *be the operator as in Definition 5.3. Then, if* $\gamma \in [0, 1)$ *it is a* γ*-contraction in the* L_∞*-norm, i.e., for every bounded measurable functions* $f, g \in \mathscr{B}(\mathcal{S})$ *it holds that:*

$$\left\| T^{*,*} f - T^{*,*} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty.$$

Furthermore, $V^{*,*}$ *is its unique fixed point, i.e., it fulfills the following Bellman optimality equation:*

$$V^{*,*} = T^{*,*} V^{*,*}.$$

Proof. We limit the proof for the operator of the state value functions. The proof can be straightforwardly extended for the operators for the Q-function and U-function. Let $f, g \in \mathscr{B}(\mathcal{S})$ and $s \in \mathcal{S}$, we have:

$$\begin{aligned}
&\left| \left(T^{*,*} f\right)(s) - \left(T^{*,*} g\right)(s) \right| \\
&\qquad = \left| \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ r(s, a, s') + \gamma f(s') \right\} - \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ r(s, a, s') + \gamma g(s') \right\} \right| \\
&\qquad = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ \left| r(s, a, s') + \gamma f(s') - r(s, a, s') - \gamma g(s') \right| \right\} \\
&\qquad \leqslant \gamma \sup_{s' \in \mathcal{S}} \left\{ \left| f(s') - g(s') \right| \right\} \\
&\qquad = \gamma \left\| f - g \right\|_\infty,
\end{aligned}$$

Thus, by applying the supremum on the left hand side, we obtain:

$$\left\| T^{*,*} f - T^{*,*} g \right\|_\infty = \sup_{s \in \mathcal{S}} \left\{ \left| \left(T^{*,*} f\right)(s) - \left(T^{*,*} g\right)(s) \right| \right\} \leqslant \gamma \left\| f - g \right\|_\infty.$$

Since $\mathscr{B}(\mathcal{S})$ is the set of bounded measurable functions having image in $\mathbb{R}$, it is a complete metric space w.r.t. the metric induced by the L_∞-norm. Thus, we can apply the Banach fixed point theorem (Banach, 1922) showing that $T^{*,*}$ has a unique fixed point. To prove the Bellman equation $V^{*,*} = T^{*,*} V^{*,*}$ we follow the reasoning of Theorem 6.2.2 of Puterman (2014), showing that for every $f \in \mathscr{B}(\mathcal{S})$ if $f = T^{*,*} f$ then $f = V^{*,*}$. Let us first prove that if $f \geqslant T^{*,*} f$ then $f \geqslant V^{*,*}$. If $f \geqslant T^{*,*} f$ it means that for all $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$, we have $f \geqslant T^{\pi,P} f$. Thus, for all $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ we have:

$$\begin{aligned}
f - V^{\pi,P} &\geqslant T^{\pi,P} f - T^{\pi,P} V^{\pi,P} \\
&= \gamma P^\pi \left(f - V^{\pi,P} \right) \geqslant 0,
\end{aligned}$$

where we exploited the fact that for a function $g \in \mathscr{B}(\mathcal{S})$ if $g \geqslant \gamma P^{\pi} g$ then $(\mathrm{Id}_{\mathcal{S}} - \gamma P^{\pi})\, g \geqslant 0$ that, in turn, implies $g \geqslant 0$ whenever $\gamma < 1$ thanks to Lemma 4.2 of (Munos, 2007). Since $f \geqslant V^{\pi,P}$ holds for all $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$, we also have $f \geqslant \sup_{\pi \in \Pi^{\mathrm{SR}}, P \in \mathcal{P}^{\mathrm{SR}}} \{V^{\pi,P}\} = V^{*,*}$. Now, we have to prove that if $f \leqslant T^{*,*} f$ then $f \leqslant V^{*,*}$. We proceed analogously, recalling that if $f \leqslant T^{*,*} f$, there exists $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ such that $f \leqslant T^{\pi,P} f$. Consequently, there exists $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ such that:

$$\begin{aligned} f - V^{\pi,P} &\leqslant T^{\pi,P} f - T^{\pi,P} V^{\pi,P} \\ &= \gamma P^{\pi} \left(f - V^{\pi,P} \right) \leqslant 0. \end{aligned}$$

Since $f \leqslant V^{\pi,P}$ holds for at least one pair $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$, we can conclude that $f \leqslant \sup_{\pi \in \Pi^{\mathrm{SR}}, P \in \mathcal{P}^{\mathrm{SR}}} \{V^{\pi,P}\} = V^{*,*}$. Combining these two statements, we conclude that if $f = T^{*,*} f$ then $f = V^{*,*}$. As an alternative, we could simply observe that these operators can be defined in terms of the Bellman optimality operators of the equivalent MDP $\widetilde{\mathcal{M}}$ of Theorem 5.1, from which we derive all the relevant properties. □

Moreover, $V^{*,*}$, $Q^{*,*}$, and $U^{*,*}$ are related by the following identities, holding for every $(s, a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned} Q^{*,*}(s,a) &= \sup_{s' \in \mathcal{S}} \left\{ U^{*,*}(s,a,s') \right\}, \\ V^{*,*}(s) &= \sup_{a \in \mathcal{A}} \left\{ Q^{*,*}(s,a) \right\} = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ U^{*,*}(s,a,s') \right\}. \end{aligned}$$

5.1.3 Greedy Policy-Transition Model Pairs

As the Q-function in an MDP allows defining the notion of greedy policy, the U-function allows introducing the notion of *greedy policy-transition model pairs*.

Definition 5.4 (Greedy Policy-Transition Model Pairs)**.** *Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be a bounded measurable function, for every state $s \in \mathcal{S}$, we say that an action-state pair $(a^+, (s')^+) \in \mathcal{A} \times \mathcal{S}$ is greedy in state s if $f(s, a^+, (s')^+) = \sup_{(a,s') \in \mathcal{A} \times \mathcal{S}} \{f(s,a,s')\}$. A* greedy policy-transition model pair *w.r.t. a function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ is any policy-transition model pair $(\pi^+, P^+) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ selecting only greedy pairs, i.e., for every state $s \in \mathcal{S}$, we have:*

$$\int_{\mathcal{A}} \pi^+(\mathrm{d}a|s) \int_{\mathcal{S}} P^+(\mathrm{d}s'|s,a) f(s,a,s') = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ f(s,a,s') \right\}.$$

Thus, if $(\pi^+, P^+) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ is greedy w.r.t. to the function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ the following identity involving the Bellman operators holds:

$$T^{\pi^+, P^+} f = T^{*,*} f.$$

5.1.4 Optimal Policy-Transition Model pairs

In the previous sections, we discussed the notion of optimal value function and derived suitable Bellman equations and operators. We study now the existence of the optimal policy-transition model pairs, that we have introduced in Definition 5.2. We proceed analogously to the case of traditional MDP, defining a suitable preference $\gtrsim$ relationship on the space of Markovian stationary policy-transition model pairs $\Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$.

	Agent and Configurator
V-function	$V^{*,*}(s) = \sup_{\pi \in \Pi^{\mathrm{SR}}, P \in \mathcal{P}^{\mathrm{SR}}} \left\{ V^{\pi,P}(s) \right\}$
	$V^{*,*}(s) = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ r(s,a,s') + \gamma V^{*,*}(s') \right\}$
	$\left(T^{*,*} f\right)(s) = \sup_{a \in \mathcal{A}, s' \in \mathcal{S}} \left\{ r(s,a,s') + \gamma f(s') \right\}$
Q-function	$Q^{*,*}(s,a) = \sup_{\pi \in \Pi^{\mathrm{SR}}, P \in \mathcal{P}^{\mathrm{SR}}} \left\{ Q^{\pi,P}(s,a) \right\}$
	$Q^{*,*}(s,a) = \sup_{s' \in \mathcal{S}} \left\{ r(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ Q^{*,*}(s',a') \right\} \right\}$
	$\left(T^{*,*} f\right)(s,a) = \sup_{s' \in \mathcal{S}} \left\{ r(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ f(s',a') \right\} \right\}$
U-function	$U^{*,*}(s,a,s') = \sup_{\pi \in \Pi^{\mathrm{SR}}, P \in \mathcal{P}^{\mathrm{SR}}} \left\{ U^{\pi,P}(s,a,s') \right\}$
	$U^{*,*}(s,a,s') = r(s,a,s') + \gamma \sup_{a' \in \mathcal{A}, s'' \in \mathcal{S}} \left\{ U^{*,*}(s',a',s'') \right\}$
	$\left(T^{*,*} f\right)(s,a,s') = r(s,a,s') + \gamma \sup_{a' \in \mathcal{A}, s'' \in \mathcal{S}} \left\{ f(s',a',s'') \right\}$

Table 5.1: *Summary of the value functions, Bellman optimal operators and Bellman optimality equations for cooperative Conf-MDPs.*

Definition 5.5 (Preorder on $\Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$)**.** *Let $\mathcal{C}$ be a cooperative Conf-MDP. The preference relationship $\gtrsim \subseteq (\Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}) \times (\Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}})$ is defined for two policy-transition model pairs $(\pi, P), (\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ as:*

$$(\pi, P) \gtrsim (\pi', P') \iff V^{\pi,P}(s) \geqslant V^{\pi',P'}(s), \quad \forall s \in \mathcal{S}. \tag{5.5}$$

This relationship inherits all the properties of the one defined for traditional MDPs (Definition 2.10) thanks to the reduction of Theorem 5.1. It is reflexive and transitive, but not antisymmetric. Based on the optimality conditions stated for Conf-MDPs (Definition 5.2), if an optimal policy-transition model exists, it must be a maximum of the preorder $\gtrsim$. The following result exploits the reduction of Theorem 5.1, to prove the existence of an optimal policy-transition model pair.

Theorem 5.3. *Let $\mathcal{C}$ be a cooperative Conf-MDP. If the state space $\mathcal{S}$ is discrete and the supremum $V^{*,*}(s) = \sup_{(a,s') \in \mathcal{A} \times \mathcal{S}} \{U^*(s,a,s')\}$ is attained for every state $s \in \mathcal{S}$, then:*

1. *there exists a Markovian stationary greedy policy-transition model pair (π^*, P^*) w.r.t. $U^{*,*}$;*
2. *(π^*, P^*) is an optimal policy-transition model pair, i.e., $(\pi^*, P^*) \gtrsim (\pi, P)$ for every policy-transition model pairs $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$;*
3. *there exists a deterministic Markovian stationary optimal policy and a deterministic Makovian stationary optimal transition model.*

Proof. The proof makes use of the reduction of Theorem 5.1. Consider the equivalent MDP $\widetilde{\mathcal{M}}$. Given Theorem 2.2, we know that there exists a policy $\tilde{\pi}^*$ for $\widetilde{\mathcal{M}}$ that is greedy w.r.t. $Q^*_{\widetilde{\mathcal{M}}}$. Moreover, we decide to pick a deterministic $\tilde{\pi}^*$. Thus, for every $s \in \mathcal{S}$ if $\tilde{\pi}^*(s) = (a, s')$ we define the deterministic policy $\pi^*(s) = a$ and $P^*(s, a) = s'$. Since $Q^*_{\widetilde{\mathcal{M}}} = U^{*,*}_{\mathcal{C}}$, it follows that π^* and P^* are greedy w.r.t. $U^{*,*}_{\mathcal{C}}$ and optimal for the cooperative Conf-MDP $\mathcal{C}$. □

This theorem resembles the one presented in Chapter 2 for traditional MDPs. Thanks to the reduction proposed in Theorem 5.1, the conclusions are essentially the same. There always exists an optimal policy-transition model pair, that can be defined as greedy w.r.t. to the suitable optimal value functions. Similarly to traditional MDPs, we can relax the definition of optimality considering a scalar objective function, instead of requiring optimality for every state. This leads to the following condition.

Definition 5.6 (J-optimality for Conf-MDPs)**.** *Let $\mathcal{C}$ be a cooperative Conf-MDP and let J be a performance index. A policy-transition model pair $(\pi^*, P^*) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ is J-optimal if for every policy-transition model pair $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$: $J^{\pi^*,P^*} \geqslant J^{\pi,P}$.*

Clearly, any policy-transition model pair that fulfills the optimality condition in Definition 5.3 also fulfills that of Definition 5.6, but not vice versa.

5.1.5 On Degenerate Solutions and Parametric Conf-MDPs

As intuition suggests, solving a Conf-MDP having access to the full set of Markovian stationary policy-transition model pairs might be of modest interest essentially for two reasons. First, in all real-world interesting scenarios it is not allowed to change the transition probabilities arbitrarily. This is because, typically, the environment dynamics incorporates both configurable and non-configurable parts. For instance, in the car driving example, the settings of the car influence the transition model and can be changed, although with some constraints (maybe related to safety). Instead, the physical laws governing the interaction between the tires and the road cannot be altered, and they also are part of the transition model. Second, with this full control on the environment, the optimal solution can be very degenerate. In this section, we investigate this phenomenon and we provide a formalization of *parametric* Conf-MDP.

Consider for instance a Conf-MDP with a reward function depending on the current state only $r(s)$. Since we are allowed to act on the policy as well as on the transition model we solve the problem by simply picking as optimal transition model the one that deterministically transitions to the state with the highest reward $P^*(s, a) \in \arg\max_{s' \in \mathcal{S}} \{r(s')\}$ and picking an arbitrary policy. More formally, consider the Bellman equation for the optimal value function, defined for every $s \in \mathcal{S}$:

$$V^{*,*}(s) = r(s) + \gamma \sup_{s' \in \mathcal{S}} \left\{ V^{*,*}(s') \right\} = r(s) + \frac{\gamma}{1-\gamma} \sup_{s' \in \mathcal{S}} \left\{ r(s') \right\},$$

where we simply observe that the action has no role since the reward is independent from it and using the fact that, in this case, $\sup_{s' \in \mathcal{S}} \{V^{*,*}(s')\} = \frac{1}{1-\gamma} \sup_{s' \in \mathcal{S}} \{r(s')\}$. Thus, P^*, as defined before, yields this performance (provided that the supremum is attained).

Suppose now that the reward function depends on the current state-action pair $r(s, a)$. To get an optimal solution, we simply need a myopic policy that maximizes the immediate reward $\pi^*(s) \in \arg\max_{a \in \mathcal{A}} \{r(s, a)\}$ and a transition model that deterministically moves

to the state with the highest reward $P^*(s,a) \in \arg\max_{s' \in \mathcal{S}}\{\max_{a' \in \mathcal{A}}\{r(s,a)\}\}$. This can be formalized by the following Bellman equation, defined for every $s \in \mathcal{S}$:

$$\begin{aligned} V^{*,*}(s) &= \sup_{a \in \mathcal{A}}\{r(s,a)\} + \gamma \sup_{s' \in \mathcal{S}} \left\{V^{*,*}(s')\right\} \\ &= \sup_{a \in \mathcal{A}}\{r(s,a)\} + \frac{\gamma}{1-\gamma} \sup_{s' \in \mathcal{S}} \sup_{a' \in \mathcal{A}}\{r(s',a')\}, \end{aligned}$$

where we exploited the fact that $\sup_{s' \in \mathcal{S}} \{V^{*,*}(s')\} = \frac{1}{1-\gamma} \sup_{s' \in \mathcal{S}} \sup_{a' \in \mathcal{A}}\{r(s',a')\}$. Thus, we observe that the choices of the policy do not influence those of the transition models, leading to completely independent problems. For this reason, whenever the supremums are attained, (π^*, P^*) is an optimal pair.

Finally, if we consider a reward function depending on the state-action-next-state triple $r(s,a,s')$ we start viewing a more interesting behavior. Indeed, while the policy just needs to maximize the immediate reward $\pi^*(s) \in \arg\max_{a \in \mathcal{A}}\{r(s,a)\}$, an optimal transition model is no longer trivial since it has to trade-off between the immediate reward (that in this case depends on the next state too) and the future reward. We formalize this phenomenon in the following Bellman equation, defined for every $s \in \mathcal{S}$:

$$V^{*,*}(s) = \sup_{s' \in \mathcal{S}} \left\{ \sup_{a \in \mathcal{A}}\{r(s,a,s')\} + \gamma V^{*,*}(s') \right\}.$$

These results highlight the important asymmetry between the agent and the configurator in a cooperative Conf-MDP when we enforce no constraint on the possible modifications on the transition models. Indeed, the agent is always myopic maximizing the immediate reward, whereas, when the reward function depends on the next state, the configurator experiences a trade-off.

As we already pointed out, this setting is quite unrealistic because the transition model typically encodes a configurable part of the environment as well as a non-configurable part. Therefore, it might be convenient to consider transition models that explicitly depends on a parameter $\mathcal{P}_\Omega = \{P_{\boldsymbol{\omega}} : \boldsymbol{\omega} \in \Omega \in \mathbb{R}^q\}$. Assuming that also the policy belongs to a suitable parametric space $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \in \mathbb{R}^p\}$, we can redefine the objective function in terms of the parameters $J(\boldsymbol{\theta}, \boldsymbol{\omega}) = J^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}$. Consequently, the optimization problem can be stated in terms of the policy and transition model parameters:

$$(\boldsymbol{\theta}^*, \boldsymbol{\omega}^*) \in \underset{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega}{\arg\max} \{J(\boldsymbol{\theta}, \boldsymbol{\omega})\}.$$

We will refer to this setting as *parametric* Conf-MDP.

5.2 Non-Cooperative Setting

The non-cooperative setting of Conf-MDPs admits arbitrary, possibly conflicting, reward functions R_{Ag} and R_{Conf}. In this scenario, the game-theoretic view of environment configuration becomes relevant. In this section, we provide a brief introduction to the topic, that we believe, deserves additional investigation as future research. Before formally defining the solution concepts and discussing their properties, let us define the notion of *best response value function* (Pérolat et al., 2017).[1]

[1] We limit our presentation to the Markovian stationary policies and transition models. It has to be studied if, in this non-cooperative setting, history-dependent policies and/or transition models play a more relevant role.

Definition 5.7 (Best Response Value Function)**.** *Let $\mathcal{C}$ be a Conf-MDP. Let $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model, the* agent best response value function *is defined for every state $s \in \mathcal{S}$ as:*

$$V_{\mathsf{Ag}}^{*,P}(s) = \sup_{\pi \in \Pi^{\mathrm{SR}}} \left\{ V_{\mathsf{Ag}}^{\pi,P}(s) \right\}.$$

Let $\pi \in \Pi^{\mathrm{SR}}$ be a policy, the configurator best response value function *is defined for every state $s \in \mathcal{S}$ as:*

$$V_{\mathsf{Conf}}^{\pi,*}(s) = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ V_{\mathsf{Conf}}^{\pi,P}(s) \right\}.$$

Thus, the agent's best response value function $V_{\mathsf{Ag}}^{*,P}$ represents the best performance achievable, fixing the transition model, and looking for the policy in Π^{SR}. This is actually the traditional value function V^* for an MDP with transition model P and inherits all the properties of standard MDPs. Instead, the configuration best response value function $V_{\mathsf{Conf}}^{\pi,*}$ represents the best performance achievable having fixed the policy and searching the transition model in $\mathcal{P}^{\mathrm{SR}}$. Clearly, we can define the best response value function in terms of the Q-function and the U-function as well. To keep the presentation concise, we limit to the V-function. The reader can refer to Table 5.2 for the complete overview of the best response value functions. It is immediate to realize that finding the best response configuration can be reduced to solving a particular MDP, in the same sense as in Theorem 5.1.

Theorem 5.4. *Let $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R_{\mathsf{Ag}}, R_{\mathsf{Conf}}, \gamma)$ be a Conf-MDP. Let $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model and $\widetilde{\mathcal{M}}_{\mathsf{Ag}} = (\mathcal{S}, \mathcal{A}, P, \mu_0, R_{\mathsf{Ag}}, \gamma)$ be an MDP. Then, for every $\pi \in \Pi^{\mathrm{SR}}$ and every state $s \in \mathcal{S}$ it holds that:*

$$V_{\mathcal{C},\mathsf{Ag}}^{\pi,P}(s) = V_{\widetilde{\mathcal{M}}_{\mathsf{Ag}}}^{\pi}(s). \tag{5.6}$$

Let $\pi \in \Pi^{\mathrm{SR}}$ be a policy and let $\widetilde{\mathcal{M}}_{\mathsf{Conf}} = (\mathcal{S} \times \mathcal{A}, \mathcal{S}, \widetilde{P}_{\mathsf{Conf}}, \widetilde{\mu}_{0,\mathsf{Conf}}, \widetilde{R}_{\mathsf{Conf}}, \gamma)$ be an MDP defined as follows:

- $\widetilde{P}_{\mathsf{Conf}} : (\mathcal{S} \times \mathcal{A}) \times \mathcal{S} \to \mathscr{P}(\mathcal{S} \times \mathcal{A})$*, defined for every* $((s,a), s') \in (\mathcal{S} \times \mathcal{A}) \times \mathcal{S}$ *and* $(s'', a'') \in \mathcal{S} \times \mathcal{A}$ *as:*

$$\widetilde{P}_{\mathsf{Conf}}(\mathrm{d}s'', \mathrm{d}a''|(s,a), s') = \delta_{s'}(\mathrm{d}s'')\pi(\mathrm{d}a''|s''),$$

- $\widetilde{\mu}_{0,\mathsf{Conf}} \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ *is defined for every* $(s,a) \in \mathcal{S} \times \mathcal{A}$ *as:*

$$\widetilde{\mu}_{0,\mathsf{Conf}}(\mathrm{d}(s,a)) = \widetilde{\mu}_0(\mathrm{d}s)\pi(\mathrm{d}a|s),$$

- $\widetilde{R}_{\mathsf{Conf}} : (\mathcal{S} \times \mathcal{A}) \times \mathcal{S} \times (\mathcal{S} \times \mathcal{A}) \to \mathscr{P}(\mathbb{R})$*, defined for every* $((s,a), s', (s'', a'')) \in (\mathcal{S} \times \mathcal{A}) \times \mathcal{S} \times (\mathcal{S} \times \mathcal{A})$ *and* $r \in \mathbb{R}$ *as:*

$$\widetilde{R}_{\mathsf{Conf}}(\mathrm{d}r|(s,a), s', (s'', a'')) = R_{\mathsf{Conf}}(\mathrm{d}r|s, a, s').$$

Then, for every $P \in \mathcal{P}^{\mathrm{SR}}$ define the policy $\widetilde{\pi}_{\mathsf{Conf}} : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S})$ for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $s' \in \mathcal{S}$ as:

$$\widetilde{\pi}_{\mathsf{Conf}}(\mathrm{d}s'|s, a) = P(\mathrm{d}s'|s, a).$$

Then, for every $P \in \mathcal{P}^{\mathrm{SR}}$ and for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ it holds that:

$$Q^{\pi,P}_{\mathcal{C},\mathsf{Conf}}(s, a) = V^{\widetilde{\pi}_{\mathsf{Conf}}}_{\widetilde{\mathcal{M}}_{\mathsf{Conf}}}(s, a). \tag{5.7}$$

Proof. The first part of the theorem is straightforward. Concerning the second part, it suffices to prove that the Bellman operator associated to (π, P) in the Conf-MDP $T^{\pi,P}_{\mathcal{C},\mathsf{Conf}}$ equals the Bellman operator associated to $\widetilde{\pi}_{\mathsf{Conf}}$ in the MDP $\mathcal{M}_{\mathsf{Conf}}$, i.e., $T^{\widetilde{\pi}_{\mathsf{Conf}}}_{\widetilde{\mathcal{M}}_{\mathsf{Conf}}}$. Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and let $(s, a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
\left(T^{\widetilde{\pi}_{\mathsf{Conf}}}_{\widetilde{M}_{\mathsf{Conf}}} f\right)(s, a) &= \int_{\mathcal{S}} \widetilde{\pi}(\mathrm{d}s'|s, a) \int_{\mathcal{S}\times\mathcal{A}} \widetilde{P}_{\mathsf{Conf}}(\mathrm{d}(s'', a'')|(s, a), s') \\
&\quad \times \left(\widetilde{r}_{\mathsf{Conf}}((s, a), s', (s'', a'')) + \gamma f(s'', a'')\right) \\
&= \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \int_{\mathcal{S}\times\mathcal{A}} \delta_{s'}(\mathrm{d}s'')\pi(\mathrm{d}a''|s'') \\
&\quad \times \left(\widetilde{r}_{\mathsf{Conf}}(s, a, s') + \gamma f(s'', a'')\right) \\
&= \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \int_{\mathcal{A}} \pi(\mathrm{d}a''|s') \left(\widetilde{r}_{\mathsf{Conf}}(s, a, s') + \gamma f(s', a'')\right) \\
&= \left(T^{\pi,P}_{\mathcal{C},\mathsf{Conf}} f\right)(s, a).
\end{aligned}$$

□

This result shows that in a Conf-MDP the task of searching for the optimal policy, i.e., the activity carried out by the agent, is essentially equivalent to the solution of an MDP, which was quite obvious. Less trivial is the reduction of the configurator's activity, i.e., the search of the optimal transition model, to the solution of a suitably defined MDP. Similarly to Theorem 5.1, we do not see Theorem 5.4 from an algorithmic viewpoint, but as a way to import the properties of standard MDPs to the case of non-cooperative Conf-MDPs. We now define the following *Bellman best response operators* (Pérolat et al., 2017).

Definition 5.8 (Bellman Best Response Operators)**.** *Let $\mathcal{C}$ be a Conf-MDP, $\pi \in \Pi^{\mathrm{SR}}$ be a policy, and $P \in \mathcal{P}^{\mathrm{SR}}$ be a transition model. The* agent Bellman best response operator *for the state value function $T^{*,P}_{\mathsf{Ag}} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S})$ and every state $s \in \mathcal{S}$ as:*

$$\left(T^{*,P}_{\mathsf{Ag}} f\right)(s) = \sup_{a\in\mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \left(r(s, a, s') + \gamma f(s')\right) \right\}. \tag{5.8}$$

The configurator Bellman best response operator *for the state value function $T^{*,P}_{\mathsf{Conf}} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ is defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S})$ and every state $s \in \mathcal{S}$ as:*

$$\left(T^{\pi,*}_{\mathsf{Conf}} f\right)(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \sup_{s'\in\mathcal{S}} \left\{ r(s, a, s') + \gamma f(s') \right\}. \tag{5.9}$$

It is immediate to notice that the agent Bellman best response operator $T^{*,P}_{\mathsf{Ag}}$ is the Bellman optimality operator T^* for traditional MDPs. For this reason, it inherits all the properties, especially the contraction property and the fact that $V^{*,P}_{\mathsf{Ag}}$ is its unique fixed point. The very same properties can be proved for the configurator Bellman best response operator $T^{\pi,*}_{\mathsf{Conf}}$.

Proposition 5.5. *Let $T^{*,P}_{\mathsf{Ag}}, T^{\pi,*}_{\mathsf{Conf}} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ be the operators as in Definition 5.8. Then, if $\gamma \in [0,1)$ they are a γ-contraction in the L_∞-norm, i.e., for every bounded measurable functions $f, g \in \mathscr{B}(\mathcal{S})$ it holds that:*

$$\left\| T^{*,P}_{\mathsf{Ag}} f - T^{*,P}_{\mathsf{Ag}} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty ,$$
$$\left\| T^{\pi,*}_{\mathsf{Conf}} f - T^{\pi,*}_{\mathsf{Conf}} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty .$$

Furthermore, $V^{*,P}_{\mathsf{Ag}}$ and $V^{\pi,*}_{\mathsf{Conf}}$ are their unique fixed points, i.e., they fulfill the following Bellman best response equations:

$$V^{*,P}_{\mathsf{Ag}} = T^{*,P}_{\mathsf{Ag}} V^{*,P}_{\mathsf{Ag}},$$
$$V^{\pi,*}_{\mathsf{Conf}} = T^{\pi,*}_{\mathsf{Conf}} V^{\pi,*}_{\mathsf{Conf}}.$$

Proof. The claims about $T^{*,P}_{\mathsf{Ag}}$ are trivial since $T^{*,P}_{\mathsf{Ag}}$ is the Bellman optimal operator in the MDP $\widetilde{\mathcal{M}}_{\mathsf{Ag}}$ defined in Theorem 5.4. Concerning $T^{\pi,*}_{\mathsf{Conf}}$, we can immediately prove $V^{\pi,*}_{\mathsf{Conf}}$ is its fixed point, we show that for every $f \in \mathscr{B}(\mathcal{S})$ we have that if $f = T^{\pi,*}_{\mathsf{Conf}} f$ then $f = V^{\pi,*}_{\mathsf{Conf}}$. The argument is analogous to that of Proposition 5.2. □

5.2.1 The Agent is Aware of the Configurator Presence

When the agent is aware of the presence of the configurator, its behavior becomes strategic, just like the configurator. This scenario can be thought of as a *simultaneous game* in which the agent selects the action and the configurator chooses the next state. For this reason, the Nash equilibrium (Başar and Olsder, 1998) can be a suitable solution concept for this kind of Conf-MDPs. We now rephrase the definition of Nash equilibrium for the case of Conf-MDPs.

Definition 5.9 (Nash Equilibrium in Conf-MDPs)**.** *Let $\mathcal{C}$ be a Conf-MDP. A policy $\pi^* \in \Pi^{\mathrm{SR}}$ and a transition model $P^* \in \mathcal{P}^{\mathrm{SR}}$ are a* Nash equilibrium *for the Conf-MDP $\mathcal{C}$ if for every state $s \in \mathcal{S}$, policy $\pi \in \Pi^{\mathrm{SR}}$, and transition model in $P \in \mathcal{P}^{\mathrm{SR}}$ it holds that:*

$$V^{\pi^*,P^*}_{\mathsf{Ag}}(s) \geqslant V^{\pi,P^*}_{\mathsf{Ag}}(s),$$
$$V^{\pi^*,P^*}_{\mathsf{Conf}}(s) \geqslant V^{\pi^*,P}_{\mathsf{Conf}}(s),$$

Given this definition, if $(\pi^*, P^*) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ is a Nash equilibrium of the Conf-MDP, $V^{\pi^*,P^*}_{\mathsf{Ag}}$ and $V^{\pi^*,P^*}_{\mathsf{Conf}}$ are fixed points of the corresponding best response operators T^{*,P^*}_{Ag} and $T^{\pi^*,*}_{\mathsf{Conf}}$. A particular case of interest, thanks to its convenient theoretical properties, is when the reward functions of agent and supervisor are opposite (Littman, 1994). In such a case, we refer to *zero-sum* Conf-MDP.

Definition 5.10 (Zero-Sum Conf-MDP)**.** *Let $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mu_0, R_{\mathsf{Ag}}, R_{\mathsf{Conf}}, \gamma)$ be a Conf-MDP. $\mathcal{C}$ is a* Zero-Sum Conf-MDP *if for every state-action-state triple $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ it holds that:*

$$R_{\mathsf{Ag}} + R_{\mathsf{Conf}} = 0 \qquad \textit{almost surely,} \tag{5.10}$$

where $R_{\mathsf{Ag}} \sim R_{\mathsf{Ag}}(\cdot|s, a, s')$ and $R_{\mathsf{Conf}} \sim R_{\mathsf{Conf}}(\cdot|s, a, s')$.

A zero-sum Conf-MDP models a fully competitive environment. It is immediate to realize, thank to the Von Neumann minimax theorem (Von Neumann, 1928), that the minimax value functions $V_{\mathsf{Ag}}^{\text{maximin}}, V_{\mathsf{Conf}}^{\text{maximin}} : \mathcal{S} \to \mathbb{R}$ can be defined for every state $s \in \mathcal{S}$ as follows:

$$\begin{aligned} V_{\mathsf{Ag}}^{\text{maximin}}(s) &= \sup_{\pi \in \Pi^{\text{SR}}} \inf_{P \in \mathcal{P}^{\text{SR}}} \left\{ V_{\mathsf{Ag}}^{\pi,P}(s) \right\} = \inf_{P \in \mathcal{P}^{\text{SR}}} \sup_{\pi \in \Pi^{\text{SR}}} \left\{ V_{\mathsf{Ag}}^{\pi,P}(s) \right\}, \\ V_{\mathsf{Conf}}^{\text{maximin}}(s) &= -V_{\mathsf{Ag}}^{\text{maximin}}(s). \end{aligned}$$

The value functions can be easily defined in terms of the corresponding Bellman minimax operators $T_{\mathsf{Ag}}^{\text{maximin}}, T_{\mathsf{Conf}}^{\text{maximin}} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$, defined for every state $s \in \mathcal{S}$ as (Busoniu et al., 2008):

$$\begin{aligned} \left(T_{\mathsf{Ag}}^{\text{maximin}} f\right)(s) &= \sup_{\pi \in \Pi^{\text{SR}}} \inf_{P \in \mathcal{P}^{\text{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \left(r_{\mathsf{Ag}}(s, a, s') + \gamma f(s') \right) \right\}, \\ \left(T_{\mathsf{Conf}}^{\text{maximin}} f\right)(s) &= -\left(T_{\mathsf{Ag}}^{\text{maximin}} f\right)(s). \end{aligned}$$

Thus, when the state-action space is finite, each application of the operator $T_{\mathsf{Ag}}^{\text{maximin}}$ requires the solution of a linear program. It can be proved that $T_{\mathsf{Ag}}^{\text{maximin}}$ is a γ-contraction in L_∞-norm (Busoniu et al., 2008). Thus, it admits a unique fixed point, that is the minimax value function $V_{\mathsf{Ag}}^{\text{maximin}}$. The same considerations hold for the configurator side.

Proposition 5.6. *Let $T_{\mathsf{Ag}}^{\textit{maximin}}, T_{\mathsf{Conf}}^{\textit{maximin}} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ be the operator defined before. Then, if $\gamma \in [0, 1)$ they are a γ-contraction in the L_∞-norm, i.e., for every bounded measurable functions $f, g \in \mathscr{B}(\mathcal{S})$ it holds that:*

$$\begin{aligned} \left\| T_{\mathsf{Ag}}^{\textit{maximin}} f - T_{\mathsf{Ag}}^{\textit{maximin}} g \right\|_\infty &\leqslant \gamma \left\| f - g \right\|_\infty, \\ \left\| T_{\mathsf{Conf}}^{\textit{maximin}} f - T_{\mathsf{Conf}}^{\textit{maximin}} g \right\|_\infty &\leqslant \gamma \left\| f - g \right\|_\infty. \end{aligned}$$

Furthermore, $V_{\mathsf{Ag}}^{\textit{maximin}}$ and $V_{\mathsf{Conf}}^{\textit{maximin}}$ are their unique fixed points, i.e., they fulfill the following Bellman optimality equations:

$$\begin{aligned} V_{\mathsf{Ag}}^{\textit{maximin}} &= T_{\mathsf{Ag}}^{\textit{maximin}} V_{\mathsf{Ag}}^{\textit{maximin}}, \\ V_{\mathsf{Conf}}^{\textit{maximin}} &= T_{\mathsf{Conf}}^{\textit{maximin}} V_{\mathsf{Conf}}^{\textit{maximin}}. \end{aligned}$$

Proof. Let $f, g \in \mathscr{B}(\mathcal{S})$ and $s \in \mathcal{S}$, we have:

$$\begin{aligned} &\left| \left(T_{\mathsf{Ag}}^{\text{maximin}} f\right)(s) - \left(T_{\mathsf{Ag}}^{\text{maximin}} g\right)(s) \right| \\ &\qquad = \left| \sup_{\pi \in \Pi^{\text{SR}}} \inf_{P \in \mathcal{P}^{\text{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s, a) \left(r_{\mathsf{Ag}}(s, a, s') + \gamma f(s') \right) \right\} \right. \end{aligned}$$

$$
\begin{aligned}
&\quad - \sup_{\pi\in\Pi^{\mathrm{SR}}} \inf_{P\in\mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma g(s') \right) \right\} \Bigg| \\
&\leqslant \sup_{\pi\in\Pi^{\mathrm{SR}}} \left| \inf_{P\in\mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma f(s') \right) \right\} \right. \\
&\quad \left. - \inf_{P\in\mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma g(s') \right) \right\} \right| \\
&\leqslant \sup_{\pi\in\Pi^{\mathrm{SR}}} \sup_{P\in\mathcal{P}^{\mathrm{SR}}} \left\{ \left| \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma f(s') \right) \right. \right. \\
&\quad \left. \left. - \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma g(s') \right) \right| \right\} \qquad \text{(P.1)} \\
&\leqslant \gamma \sup_{\pi\in\Pi^{\mathrm{SR}}} \sup_{P\in\mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{S}} P(\mathrm{d}s'|s,a) \left| f(s') - g(s') \right| \right\} \\
&\leqslant \gamma \sup_{s'\in\mathcal{S}} \left\{ \left| f(s') - g(s') \right| \right\} \\
&\gamma \left\| f - g \right\|_{\infty},
\end{aligned}
$$

where line (P.1) follows from observing that for two functions f and g it holds that:

$$
\left| \inf_{x\in\mathcal{X}} \{f(x)\} - \inf_{x\in\mathcal{X}} \{g(x)\} \right| \leqslant \sup_{x\in\mathcal{X}} \{|f(x) - g(x)|\}.
$$

By applying the supremum to the left hand side we get the contraction result. Thanks to the Banach's fixed point theorem (Banach, 1922), we conclude that the operator admits a unique fixed point. We need to prove that $V_{\mathsf{Ag}}^{\text{maximin}}$ is a fixed point of the operator $T_{\mathsf{Ag}}^{\text{maximin}}$. To this purpose, we prove that for every $f \in \mathscr{B}(\mathcal{S})$, if $f = T_{\mathsf{Ag}}^{\text{maximin}} f$ we have that $f = V_{\mathsf{Ag}}^{\text{maximin}}$. First, we prove that if $f \geqslant T_{\mathsf{Ag}}^{\text{maximin}} f$, then $f \geqslant V_{\mathsf{Ag}}^{\text{maximin}}$. Suppose that $f \geqslant T_{\mathsf{Ag}}^{\text{maximin}} f$, this means that for all $\pi \in \Pi^{\mathrm{SR}}$ there exists $P \in \mathcal{P}^{\mathrm{SR}}$ such that $f \geqslant T_{\mathsf{Ag}}^{\pi,P} f$, consequently:

$$
\begin{aligned}
f - V_{\mathsf{Ag}}^{\pi,P} &\geqslant T_{\mathsf{Ag}}^{\pi,P} f - T_{\mathsf{Ag}}^{\pi,P} V_{\mathsf{Ag}}^{\pi,P} \\
&= \gamma P^{\pi} \left(f - V_{\mathsf{Ag}}^{\pi,P} \right) \geqslant 0.
\end{aligned}
$$

Since the inequality $f \geqslant V_{\mathsf{Ag}}^{\pi,P}$ holds for all $\pi \in \Pi^{\mathrm{SR}}$ and a specific $P \in \mathcal{P}^{\mathrm{SR}}$, we have that $f \geqslant \sup_{\pi\in\Pi^{\mathrm{SR}}} \inf_{P\in\mathcal{P}^{\mathrm{SR}}} \{V_{\mathsf{Ag}}^{\pi,P}\} = V_{\mathsf{Ag}}^{\text{maximin}}$. The reverse claim, i.e., if $f \leqslant T_{\mathsf{Ag}}^{\text{maximin}} f$, then $f \leqslant V_{\mathsf{Ag}}^{\text{maximin}}$ can be proved analogously. Consequently, if $f = T_{\mathsf{Ag}}^{\text{maximin}} f$ then $f = V_{\mathsf{Ag}}^{\text{maximin}}$. □

5.2.2 The Agent is Unaware of the Configurator Presence

When the agent is unaware of the presence of the configurator, it cannot display a strategic behavior but it is reasonable to assume that it simply acts as a best responder. Thus, while the configurator acts in order to maximize its utility, the agent perceives the environment configuration as a form of non-stationarity and acts consequently. This kind of *sequential* interaction can be effectively modeled as a leader-follower game and the corresponding solution concept is the Stakelberg equilibrium (Conitzer and Sandholm, 2006).

Definition 5.11 (Stackelberg Equilibrium in Conf-MDPs)**.** *Let $\mathcal{C}$ be a Conf-MDP and let $\beta_{\mathsf{Ag}} : \mathcal{P}^{\mathrm{SR}} \to \Pi^{\mathrm{SR}}$ be a choice function in the set of agent best responses, i.e., for every transition model $P \in \mathcal{P}^{\mathrm{SR}}$, every state $s \in \mathcal{S}$ and every policy $\pi \in \Pi^{\mathrm{SR}}$:*

$$
V^{\beta_{\mathsf{Ag}}(P),P}(s) \geqslant V^{\pi,P}(s). \tag{5.11}
$$

A policy $\pi^* \in \Pi^{\mathrm{SR}}$ *and a transition model* $P^* \in \mathcal{P}^{\mathrm{SR}}$ *are a* β_{Ag}-Stackelberg equilibrium *for the Conf-MDP* $\mathcal{C}$ *if for every state* $s \in \mathcal{S}$*, policy* $\pi \in \Pi^{\mathrm{SR}}$*, and transition model in* $P \in \mathcal{P}^{\mathrm{SR}}$ *it holds that* $\pi^* = \beta_{\mathsf{Ag}}(P^*)$ *and for every* $s \in \mathcal{S}$*:*

$$V_{\mathsf{Ag}}^{\pi^*,P^*}(s) \geqslant V_{\mathsf{Ag}}^{\pi,P^*}(s),$$
$$V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P^*),P^*}(s) \geqslant V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P}(s),$$

Furthermore, we define the β_{Ag}-*Stackelberg state value function* for every state $s \in \mathcal{S}$ as:

$$V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} \right\}.$$

Clearly, different choices of the best response function β_{Ag} lead to different notions of Stackelberg equilibrium (Breton et al., 1988). Specifically, if ties are broken in favor of the configurator, we refer to *strong Stackelberg equilibrium*, whereas if ties are broken in favor of the agent, we refer to *weak Stackelberg equilibrium*. If $(\pi^*, P^*) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ is a β_{Ag}-Stackelberg Equilibrium then it must be that $\pi^* = \beta_{\mathsf{Ag}}(P)$ is a best response for the agent, i.e., $V_{\mathsf{Ag}}^{\pi^*,P^*}$ must be a fixed point of the agent best response operator T_{Ag}^{*,P^*}. Concerning the configurator choice, we can define the β_{Ag}-Stackelberg Bellman operator $T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ defined for every $s \in \mathcal{S}$ as:

$$\left(T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f\right)(s) = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma f(s') \right) \right\}.$$

This operator preserves most of the properties of the traditional Bellman operators, especially the contraction in L_∞-norm.

Proposition 5.7. *Let* $T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} : \mathscr{B}(\mathcal{S}) \to \mathscr{B}(\mathcal{S})$ *be the operator defined above. Then, if* $\gamma \in [0,1)$ *it is a* γ*-contraction in the* L_∞*-norm, i.e., for every bounded measurable functions* $f, g \in \mathscr{B}(\mathcal{S})$ *it holds that:*

$$\left\| T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f - T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} g \right\|_\infty \leqslant \gamma \left\| f - g \right\|_\infty.$$

Furthermore, $V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P^*),P^*}$ *is its unique fixed point, i.e., it fulfills the following Bellman optimality equation:*

$$V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} = T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}.$$

Proof. Let $f, g \in \mathscr{B}(\mathcal{S})$ and $s \in \mathcal{S}$, we have:

$$\begin{aligned}
&\left| \left(T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f\right)(s) - \left(T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} g\right)(s) \right| \\
&\qquad = \left| \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma f(s') \right) \right\} \right. \\
&\qquad\quad \left. - \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma g(s') \right) \right\} \right| \\
&\qquad \leqslant \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ \left| \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma f(s') \right) \right. \right.
\end{aligned}$$

$$
\begin{aligned}
&\left. - \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left(r_{\mathsf{Conf}}(s,a,s') + \gamma g(s') \right) \right| \Bigg\} \\
&\leqslant \gamma \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ \int_{\mathcal{A}} \int_{\mathcal{S}} (\beta_{\mathsf{Ag}}(P))(\mathrm{d}a|s) P(\mathrm{d}s'|s,a) \left| f(s') - g(s') \right| \right\} \\
&\leqslant \gamma \sup_{s' \in \mathcal{S}} \left\{ \left| f(s') - g(s') \right| \right\} \\
&\gamma \left\| f - g \right\|_{\infty} .
\end{aligned}
$$

By applying the supremum to the left hand side we get the result. By recalling that the conditions for the application of the Banach's fixed point theorem (Banach, 1922) are fulfilled, we conclude that $T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$ admits a unique fixed point. We now prove that $V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$ is a fixed point of $T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$. To this purpose, we show that for any bounded measurable function $f \in \mathscr{B}(\mathcal{S})$ if $f = T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f$ then $f = V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$. By combining this with the existence of the fixed point we get to the result. First, we prove that if $f \geqslant T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f$, then $f \geqslant V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$. Suppose that $f \geqslant T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f$, this means that for all $P \in \mathcal{P}^{\mathrm{SR}}$ we have $f \geqslant T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} f$. Consequently:

$$
\begin{aligned}
f - V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} &\geqslant T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} f - T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} \\
&= \gamma P^{\beta_{\mathsf{Ag}}(P)} \left(f - V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} \right) \geqslant 0.
\end{aligned}
$$

Since $f \geqslant V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P}$ holds for all $P \in \mathcal{P}^{\mathrm{SR}}$, we have that $f \geqslant \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \{ V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(P),P} \} = V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$. The reverse claim, i.e., if $f \leqslant T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f$, then $f \leqslant V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$ can be proved analogously. Consequently, if $f = T_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*} f$ then $f = V_{\mathsf{Conf}}^{\beta_{\mathsf{Ag}}(*),*}$. □

It is important to point out that, unfortunately, these properties are far from leading to practical algorithms, as the operator requires the explicit knowledge of the best response choice function. Although we believe that this non-cooperative view of Conf-MDPs is quite appealing and of interest for the real-world applications, our current understating of the problem is rather shallow. We still miss a study of the existence of the equilibria presented above for the Conf-MDPs as well as suitable algorithms to compute them.

Remark 5.2 (Connection with Markov Games)**.** *The reader might be tempted to reduce the non-cooperative Conf-MDP setting to a* Markov game *(Busoniu et al., 2008) having the agent and the configurator as players. We believe that this reduction is not straightforward. Indeed, while the agent takes its action based on the current state only (it plays a policy), the configurator bases its next-state decision on the current state but also the current action (it plays a transition model). This distinction is independent of the fact that the configurator* observes *the current agent's action. If it does, then we can map this setting to a* sequential game *in which the agent plays first and the configurator plays later, observing the agent's action. Instead, if the configurator does not observe the action the setting is that of a* simultaneous game*, but its strategy set is composed of transition models that provide a probability distribution depending on the action too, leading to a form of* partial information*.*

	Agent	Configurator
V-function	$V_{\mathsf{Ag}}^{*,P}(s) = \sup_{\pi \in \Pi^{\mathrm{SR}}} \left\{ V^{\pi,P}(s) \right\}$	$V_{\mathsf{Conf}}^{\pi,*}(s) = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ V^{\pi,P}(s) \right\}$
	$V_{\mathsf{Ag}}^{*,P}(s) = \sup_{a \in \mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s'\|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma V_{\mathsf{Ag}}^{*,P}(s') \right) \right\}$	$V_{\mathsf{Conf}}^{\pi,*}(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a\|s) \sup_{s' \in \mathcal{S}} \left\{ r_{\mathsf{Conf}}(s,a,s') + \gamma V_{\mathsf{Conf}}^{\pi,*}(s') \right\}$
	$\left(T_{\mathsf{Ag}}^{*,P} f \right)(s) = \sup_{a \in \mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s'\|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma f(s') \right) \right\}$	$\left(T_{\mathsf{Conf}}^{\pi,*} f \right)(s) = \int_{\mathcal{A}} \pi(\mathrm{d}a\|s) \sup_{s' \in \mathcal{S}} \left\{ r_{\mathsf{Conf}}(s,a,s') + \gamma f(s') \right\}$
Q-function	$Q_{\mathsf{Ag}}^{*,P}(s,a) = \sup_{\pi \in \Pi^{\mathrm{SR}}} \left\{ Q^{\pi,P}(s,a) \right\}$	$Q_{\mathsf{Conf}}^{\pi,*}(s,a) = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ Q^{\pi,P}(s,a) \right\}$
	$Q_{\mathsf{Ag}}^{*,P}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'\|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ Q_{\mathsf{Ag}}^{*,P}(s',a') \right\} \right)$	$Q_{\mathsf{Conf}}^{\pi,*}(s,a) = \sup_{s' \in \mathcal{S}} \left\{ r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'\|s') Q_{\mathsf{Conf}}^{\pi,*}(s',a') \right\}$
	$\left(T_{\mathsf{Ag}}^{*,P} f \right)(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'\|s,a) \left(r_{\mathsf{Ag}}(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ f(s',a') \right\} \right)$	$\left(T_{\mathsf{Conf}}^{\pi,*} f \right)(s,a) = \sup_{s' \in \mathcal{S}} \left\{ r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'\|s') f(s',a') \right\}$
U-function	$U_{\mathsf{Ag}}^{*,P}(s,a,s') = \sup_{\pi \in \Pi^{\mathrm{SR}}} \left\{ U^{\pi,P}(s,a,s') \right\}$	$U_{\mathsf{Conf}}^{\pi,*}(s,a,s') = \sup_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{ U^{\pi,P}(s,a,s') \right\}$
	$U_{\mathsf{Ag}}^{*,P}(s,a,s') = r_{\mathsf{Ag}}(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s''\|s',a') U_{\mathsf{Ag}}^{*,P}(s',a',s'') \right\}$	$U_{\mathsf{Conf}}^{\pi,*}(s,a,s') = r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'\|s') \sup_{s'' \in \mathcal{S}} \left\{ U_{\mathsf{Conf}}^{\pi,*}(s',a',s'') \right\}$
	$\left(T_{\mathsf{Ag}}^{*,P} f \right)(s,a,s') = r_{\mathsf{Ag}}(s,a,s') + \gamma \sup_{a' \in \mathcal{A}} \left\{ \int_{\mathcal{S}} P(\mathrm{d}s''\|s',a') f(s',a',s'') \right\}$	$\left(T_{\mathsf{Conf}}^{\pi,*} f \right)(s,a,s') = r_{\mathsf{Conf}}(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'\|s') \sup_{s'' \in \mathcal{S}} \left\{ f(s',a',s'') \right\}$

Table 5.2: *Summary of the best response value functions, Bellman operators and Bellman equations for non-cooperative Conf-MDPs.*

Part II

Learning in Cooperative Configurable Markov Decision Processes

CHAPTER 6

Learning in Finite Cooperative Configurable Markov Decision Processes

6.1 Introduction

In this chapter, we study the problem of solving a Conf-MDP in the cooperative setting when the state and action spaces are finite. Solving a Conf-MDP, as we introduced in Chapter 5, according to the optimality condition of Definition 5.6, means finding a policy and a transition model so that they jointly maximize the expected return:

$$(\pi^*, P^*) \in \underset{(\pi,P)\in\Pi^{\mathrm{SR}}\times\mathcal{P}^{\mathrm{SR}}}{\arg\max} \left\{J^{\pi,P}\right\}.$$

This general optimality condition, however, allows full control on both the policy and the transition model. However, typically, the search must be constrained because of the specific requirements that need to be guaranteed in the application of interest. This is particularly true for the transition model. Indeed, in several cases of interest, the transition model accounts for portions of the environment that are or immutable (e.g., physical laws). Thus, the computation of the optimal policy-model pair is typically carried out in a suitably tailored subspace $\Pi \times \mathcal{P} \subseteq \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$.

This problem can be addressed using numerous tools (e.g., gradient methods, entropy methods). In this chapter, we propose a method to *jointly* and *adaptively* optimize the policy and the transition model, named *Safe Policy-Model Iteration* (SPMI, Metelli et al., 2018a). The algorithm adopts a safe learning approach (García and Fernández, 2015)

based on the maximization of a lower bound on the guaranteed performance improvement, yielding a sequence of policy-transition model pairs with monotonically increasing performance. The safe learning perspective makes our approach suitable for critical applications where performance degradation during learning is not allowed (e.g., industrial scenarios where extensive exploration of the policy space might damage the machinery). In the standard RL framework, the usage of a lower bound to guide the choice of the policy has been first introduced by Conservative Policy Iteration (CPI, Kakade and Langford, 2002), improved by Safe Policy Iteration (SPI, Pirotta et al., 2013b) and subsequently exploited in (Ghavamzadeh et al., 2016; Abbasi-Yadkori et al., 2016; Papini et al., 2017, 2020; Vieillard et al., 2020). These methods revealed their potential thanks to the preference towards small policy updates, preventing from moving in a single step too far away from the current policy and avoiding premature convergence to suboptimal policies. A similar rationale is at the basis of Relative Entropy Policy Search (REPS, Peters et al., 2010), and, more recently, Trust Region Policy Optimization (TRPO, Schulman et al., 2015), Proximal Policy Optimization (PPO, Schulman et al., 2017), and Policy Optimization via Importance Sampling (POIS, Metelli et al., 2018b). In order to introduce our framework and highlight its benefits, we limit our analysis to the scenario in which the model space (and the policy space) is known. However, when the model space is unknown, we could resort to a sample-based version of SPMI, which could be derived by adapting those of SPI (Pirotta et al., 2013b).

Chapter Outline The chapter is organized as follows. We start in Section 6.2, in which we introduce the notion of relative advantage function that will be employed in the theoretical results and in the derivation of the algorithm. In Section 6.3 we first derive a bound on the divergence between the γ-discounted stationary distributions induced by different policy-transition model pairs. Then, we employ this result to obtain a performance improvement bound. Based on these theoretical results, we outline the main features of SPMI (Section 6.4) in comparison with the existing approaches, along with some theoretical results (Section 6.5). Then, we present the experimental evaluation (Section 6.6) in two explicative domains, simple abstractions of the motivational applications of Conf-MDPs, with the purpose of showing how configuring the transition model can be beneficial for the final policy performance. Finally, we present in Section 6.7 two examples of Conf-MDPs displaying some interesting behaviors when running SPMI.

6.2 Relative Advantage Functions

In this section, we introduce the notion of *relative advantage function* that will be extensively employed in the derivation of the performance improvement bounds. We already presented in Section 4.4 the notion of *advantage function*. Specifically, the policy, model, and coupled advantage functions, respectively, are defined for every $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ as:

$$A^{\pi,P}(s,a) = Q^{\pi,P}(s,a) - V^{\pi,P}(s),$$
$$A^{\pi,P}(s,a,s') = U^{\pi,P}(s,a,a') - Q^{\pi,P}(s),$$
$$\widetilde{A}^{\pi,P}(s,a,s') = U^{\pi,P}(s,a,a') - V^{\pi,P}(s).$$

These functions quantify the one-step gain in performance attained in state $s \in \mathcal{S}$ by either playing action $a \in \mathcal{A}$, for the policy advantage, selecting the next state $s' \in \mathcal{S}$ given that action $a \in \mathcal{A}$ was played, for the model advantage, or both for the coupled advantage, compared to playing policy π and employing transition model P. In order to evaluate the one-step improvement in performance attained by a new policy π' or model P' when the current policy is π and the current model is P, we introduce the *(uncoupled) relative advantage functions* (Kakade and Langford, 2002) defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$A_{\pi,P}^{\pi',P}(s) = \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) A^{\pi,P}(s,a),$$
$$A_{\pi,P}^{\pi,P'}(s,a) = \int_{\mathcal{S}} P'(\mathrm{d}s'|s,a) A^{\pi,P}(s,a,s'),$$

and the corresponding expected values under the γ-discounted distributions:

$$\mathbb{A}_{\pi,P,\mu_0}^{\pi',P} = \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) A_{\pi,P}^{\pi',P}(s)$$
$$\mathbb{A}_{\pi,P,\mu_0}^{\pi,P'} = \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_\gamma^{\pi,P}(\mathrm{d}s, \mathrm{d}a) A_{\pi,P}^{\pi,P'}(s,a).$$

To capture the combined effect of selecting the action with a new policy π' and the next state with the new transition model P', we introduce the *coupled relative advantage function* defined for every state $s \in \mathcal{S}$ as:

$$A_{\pi,P}^{\pi',P'}(s) = \int_{\mathcal{S}} \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) P'(\mathrm{d}s'|s,a) \widetilde{A}^{\pi,P}(s,a,s'),$$

Thus, $A_{\pi,P}^{\pi',P'}$ represents the one-step improvement attained by the new policy-transition model pair $(\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ over the current one $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$, i.e., the local gain in performance yielded by selecting an action with π' and the next state with P'. The corresponding expectation under the γ-discounted distribution is given by:

$$\mathbb{A}_{\pi,P,\mu_0}^{\pi',P'} = \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s).$$

To lighten the notation, we remove the subscript of the initial state distribution μ_0 whenever clear from the context. Thus, we simply write $\mathbb{A}_{\pi,P}^{\pi',P}$, $\mathbb{A}_{\pi,P}^{\pi,P'}$, and $\mathbb{A}_{\pi,P}^{\pi',P'}$. The following result relates the coupled relative advantage function with the corresponding (uncoupled) relative advantage functions.

Lemma 6.1. *Let $A_{\pi,P}^{\pi',P'}$ be the coupled relative advantage function, $A_{\pi,P}^{\pi',P}$ and $A_{\pi,P}^{\pi,P'}$ be the (uncoupled) policy and model relative advantage functions respectively. Then, for every state $s \in \mathcal{S}$ it holds that:*

$$A_{\pi,P}^{\pi',P'}(s) = A_{\pi,P}^{\pi',P}(s) + \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) A_{\pi,P}^{\pi,P'}(s,a).$$

Proof. Let $s \in \mathcal{S}$, let us consider the following derivation:

$$\begin{aligned}
A^{\pi',P'}_{\pi,P}(s) &= \int_{\mathcal{A}}\int_{\mathcal{S}} \pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)U^{\pi,P}(s,a,s') - V^{\pi,P}(s) \\
&= \int_{\mathcal{A}}\int_{\mathcal{S}} \pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)U^{\pi,P}(s,a,s') - V^{\pi,P}(s) \\
&\quad \pm \int_{\mathcal{A}}\int_{\mathcal{S}} \pi'(\mathrm{d}a|s)P(\mathrm{d}s'|s,a)U^{\pi,P}(s,a,s') \\
&= \int_{\mathcal{A}}\int_{\mathcal{S}} \pi'(\mathrm{d}a|s)P(\mathrm{d}s'|s,a)U^{\pi,P}(s,a,s') - V^{\pi,P}(s) \\
&\quad + \int_{\mathcal{A}}\int_{\mathcal{S}} \pi'(\mathrm{d}a|s)\left(P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a)\right)U^{\pi,P}(s,a,s') \\
&= \int_{\mathcal{A}} \pi'(\mathrm{d}a|s)Q^{\pi,P}(s,a)\mathrm{d}a - V^{\pi,P}(s) \qquad \text{(P.1)} \\
&\quad + \int_{\mathcal{A}} \pi'(\mathrm{d}a|s)\int_{\mathcal{S}}\left(P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a)\right)U^{\pi,P}(s,a,s') \\
&= A^{\pi',P}_{\pi,P}(s) + \int_{\mathcal{A}} \pi'(a|s)A^{\pi,P'}_{\pi,P}(s,a)\mathrm{d}a, \qquad \text{(P.2)}
\end{aligned}$$

where line (P.1) is obtained by recalling that $Q^{\pi,P}(s,a) = \int_{\mathcal{S}} P(\mathrm{d}s'|s,a)U^{\pi,P}(s,a,s')$, the first addendum of line (P.2) follows from observing that:

$$A^{\pi',P}_{\pi,P}(s) = \int_{\mathcal{A}} \pi'(\mathrm{d}a|s)A^{\pi,P}(s,a) = \int_{\mathcal{A}} \pi'(\mathrm{d}a|s)\left(Q^{\pi,P}(s,a) - V^{\pi,P}(s)\right),$$

and similarly the second addendum of line (P.2) comes from the identity:

$$\begin{aligned}
A^{\pi,P'}_{\pi,P}(s,a) &= \int_{\mathcal{S}} P'(\mathrm{d}s'|s,a)A^{\pi,P}(s,a,s') \\
&= \int_{\mathcal{S}} P'(\mathrm{d}s'|s,a)\left(U^{\pi,P}(s,a,s') - Q^{\pi,P}(s,a)\right).
\end{aligned}$$

□

6.3 Performance Improvement Bound

The goal of this section is to provide a lower bound to the performance improvement $J^{\pi',P'} - J^{\pi,P}$ obtained by moving from a policy-transition model pair $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ to another pair $(\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$. Since we aim at providing a *safe* learning algorithm, we look for a lower bound on the performance improvement that can be evaluated using samples collected with the current pair (π, P). We follow a path similar to that of (Kakade and Langford, 2002) and (Pirotta et al., 2013a). First, we derive an upper bound on the divergence between the γ-discounted stationary distributions induced by (π, P) and (π', P') (Section 6.3.1). Then, we employ this result to lower bound the performance improvement (Section 6.3.2). Finally, we compare the obtained result with the ones (mainly involving the policy only) already existing in the literature.

6.3.1 Bound on the γ-discounted Stationary Distribution

We start providing a bound for the total variation distance of γ-discounted stationary distributions under different policy-transition model pairs.

Proposition 6.2. *Let $(\pi, P), (\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ be two policy-transition model pairs for a Conf-MDP $\mathcal{C}$, the TV-norm of the difference between the γ-discounted state distributions can be upper bounded, for any $\gamma \in [0, 1)$ as:*

$$\left\|\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}\right\|_{\mathrm{TV}} \leqslant \frac{\gamma}{1-\gamma}\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}},$$

where:

$$\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} = \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s)\left\|P'^{\pi'}(\cdot|s) - P^{\pi}(\cdot|s)\right\|_{\mathrm{TV}}.$$

Proof. Exploiting the recursive equation of the γ-discounted state distribution (Section 2.3) we can write the distributions difference as follows in operator form:

$$\begin{aligned}
\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P} &= (1-\gamma)\mu_0 + \gamma\mu_\gamma^{\pi',P'}(P')^{\pi'} - (1-\gamma)\mu_0 - \gamma\mu_\gamma^{\pi,P}P^{\pi} \\
&= \gamma\mu_\gamma^{\pi',P'}(P')^{\pi'} - \gamma\mu_\gamma^{\pi,P}P^{\pi} \pm \mu_\gamma^{\pi,P}(P')^{\pi'} \\
&= \gamma\left(\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}\right)(P')^{\pi'} + \gamma\mu_\gamma^{\pi,P}\left((P')^{\pi'} - P^{\pi}\right) \\
&= \gamma\mu_\gamma^{\pi,P}\left((P')^{\pi'} - P^{\pi}\right)\left(\mathrm{Id}_{\mathcal{S}} - \gamma(P')^{\pi'}\right)^{-1},
\end{aligned}$$

where we exploited the recursive definition of $\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}$ and recalled that $\gamma < 1$. We proceed by applying the $\|\cdot\|_{\mathrm{TV}}$:

$$\begin{aligned}
\left\|\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}\right\|_{\mathrm{TV}} &= \gamma\left\|\mu_\gamma^{\pi,P}\left((P')^{\pi'} - P^{\pi}\right)\left(\mathrm{Id}_{\mathcal{S}} - \gamma(P')^{\pi'}\right)^{-1}\right\|_{\mathrm{TV}} \\
&\leqslant \frac{\gamma}{1-\gamma}\left\|\mu_\gamma^{\pi,P}\left((P')^{\pi'} - P^{\pi}\right)\right\|_{\mathrm{TV}} \qquad \text{(P.3)} \\
&= \frac{1}{2}\frac{\gamma}{1-\gamma}\left|\int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\int_{\mathcal{S}}\left((P')^{\pi'} - P^{\pi}\right)(\mathrm{d}s'|s)\right| \\
&\leqslant \frac{1}{2}\frac{\gamma}{1-\gamma}\int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\int_{\mathcal{S}}\left|(P')^{\pi'} - P^{\pi}\right|(\mathrm{d}s'|s) \\
&= \frac{\gamma}{1-\gamma}\int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\left\|\left((P')^{\pi'} - P^{\pi}\right)(\cdot|s)\right\|_{\mathrm{TV}} \qquad \text{(P.4)} \\
&= \frac{\gamma}{1-\gamma}\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}.
\end{aligned}$$

where line (P.3) derives Hölder's inequality and by observing that:

$$\left\|\left(\mathrm{Id}_{\mathcal{S}} - \gamma(P')^{\pi'}\right)^{-1}\right\|_\infty = \left\|\sum_{i=0}^{\infty}\gamma^i\left((P')^{\pi'}\right)^i\right\|_\infty \leqslant \sum_{i=0}^{\infty}\gamma^i\left\|\left((P')^{\pi'}\right)^i\right\|_\infty \leqslant \sum_{i=0}^{\infty}\gamma^i = \frac{1}{1-\gamma},$$

since $\left\|\left((P')^{\pi'}\right)^i\right\|_\infty = 1$ being a probability measure and via an application of Hölder's inequality. Line (P.4) follows from the definition of total variation norm. □

This proposition provides a way to upper bound the difference of the γ-discounted state distributions in terms of the state kernel dissimilarity. The state transition kernel couples the effects of the policy and the transition model, but it is convenient to keep their contribution separated, getting the following looser bound.

Corollary 6.3. *Let $(\pi, P), (\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ be two policy-transition model pairs for a Conf-MDP $\mathcal{C}$, the TV-norm of the difference between the γ-discounted state stationary distributions can be upper bounded, for any $\gamma \in [0, 1)$ as:*

$$\left\|\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}\right\|_{\mathrm{TV}} \leqslant \frac{\gamma}{1-\gamma}\left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right),$$

where:

$$\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} = \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s)\left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}},$$
$$\left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} = \int_{\mathcal{S}}\int_{\mathcal{A}} \mu_\gamma^{\pi,P}(\mathrm{d}s, \mathrm{d}a)\left\|P'(\cdot|s,a) - P(\cdot|s,a)\right\|_{\mathrm{TV}}.$$

Proof. We prove this corollary by bounding the expression $\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}$. Let us start with the decomposition for every $s, s' \in \mathcal{S}$:

$$\begin{aligned}(P')^{\pi'}(\mathrm{d}s'|s) - P^{\pi}(\mathrm{d}s'|s) &= (P')^{\pi'}(\mathrm{d}s'|s) - P^{\pi}(\mathrm{d}s'|s) \pm (P')^{\pi}(\mathrm{d}s'|s)\\ &= \int_{\mathcal{A}} P'(\mathrm{d}s'|s,a)\left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right)\\ &\quad + \int_{\mathcal{A}}\left(P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a)\right)\pi(\mathrm{d}a|s).\end{aligned}$$

We apply the total variation norm at the previous expression to get:

$$\begin{aligned}\left\|(P')^{\pi'}(\cdot|s) - P^{\pi}(\cdot|s)\right\|_{\mathrm{TV}} &\leqslant \left\|\int_{\mathcal{A}} P'(\cdot|s,a)\left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right)\right\|_{\mathrm{TV}}\\ &\quad + \left\|\int_{\mathcal{A}}\left(P'(\cdot|s,a) - P(\cdot|s,a)\right)\pi(\mathrm{d}a|s)\right\|_{\mathrm{TV}}\\ &= \frac{1}{2}\int_{\mathcal{S}}\left|\int_{\mathcal{A}} P'(\mathrm{d}s'|s,a)\left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right)\right|\\ &\quad + \frac{1}{2}\int_{\mathcal{S}}\left|\int_{\mathcal{A}}\left(P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a)\right)\pi(\mathrm{d}a|s)\right|\\ &\leqslant \frac{1}{2}\int_{\mathcal{A}}\left|\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right|\int_{\mathcal{S}} P'(\mathrm{d}s'|s,a)\\ &\quad + \frac{1}{2}\int_{\mathcal{A}}\pi(\mathrm{d}a|s)\int_{\mathcal{S}}\left|P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a)\right|\\ &= \left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}} + \int_{\mathcal{A}}\pi(\mathrm{d}a|s)\left\|P'(\cdot|s,a) - P(\cdot|s,a)\right\|_{\mathrm{TV}}.\end{aligned}$$

We now take the expectation w.r.t. $\mu_\gamma^{\pi,P}$ and exploit the monotonicity property of the expectation:

$$\begin{aligned}\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} &= \int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\left\|(P')^{\pi'}(\cdot|s) - P^{\pi}(\cdot|s)\right\|_{\mathrm{TV}}\\ &\leqslant \int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}}\\ &\quad + \int_{\mathcal{S}}\mu_\gamma^{\pi,P}(\mathrm{d}s)\int_{\mathcal{A}}\pi(\mathrm{d}a|s)\left\|P'(\cdot|s,a) - P(\cdot|s,a)\right\|_{\mathrm{TV}}\end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}} \\
&\quad + \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_\gamma^{\pi,P}(\mathrm{d}s, \mathrm{d}a) \left\|P'(\cdot|s,a) - P(\cdot|s,a)\right\|_{\mathrm{TV}} \qquad \text{(P.5)} \\
&= \left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}},
\end{aligned}
$$

where line (P.5) follows by recalling that $\mu_\gamma^{\pi,P}(\mathrm{d}s)\pi(\mathrm{d}a|s) = \mu_\gamma^{\pi,P}(\mathrm{d}s, \mathrm{d}a)$. □

It is worth noting that when $P = P'$ the bound resembles Corollary 3.2 of (Pirotta et al., 2013b), but it is tighter as:

$$
\begin{aligned}
\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} &= \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}} \\
&\leqslant \sup_{s \in \mathcal{S}} \left\{\left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}}\right\} = \left\|\pi' - \pi\right\|_{\mathrm{TV},\infty}.
\end{aligned}
$$

In particular, the bound of (Pirotta et al., 2013b) might yield a large bound value in case there exist states in which the policies are very dissimilar even if those states are rarely visited according to $\mu_\gamma^{\pi,P}$. In the context of policy learning, a lower bound employing the same dissimilarity index $\|\pi' - \pi\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}$ in the penalization term has been previously proposed in (Achiam et al., 2017). Looser bounds, but more convenient from the optimization standpoint, involving KL-divergence (Pirotta et al., 2013a; Schulman et al., 2015) or other distributional divergences, like Rényi divergences (Metelli et al., 2018b), are often employed in the literature.

6.3.2 Bound on the Performance Improvement

In this section, we exploit the previous results to obtain a lower bound on the performance improvement determined by chaining the policy and the transition model. We have all the elements to express the performance improvement in terms of the relative advantage functions and the γ-discounted distributions.

Theorem 6.4. *Let $\mathcal{C}$ be a Conf-MDP. The performance improvement of policy-transition model pair $(\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ over $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ is given by:*

$$
J^{\pi',P'} - J^{\pi,P} = \frac{1}{1-\gamma} \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s).
$$

Proof. Let us start from the definition of $J^{\pi',P'}$:

$$
\begin{aligned}
(1-\gamma)J^{\pi',P'} &= \int_{\mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)\pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)r(s,a,s') \\
&= \int_{\mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)\pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)r(s,a,s') \pm \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)V^{\pi,P}(s) \qquad \text{(P.6)} \\
&= \int_{\mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)\pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)r(s,a,s') \qquad \text{(P.7)} \\
&\quad + \int_{\mathcal{S}} \left((1-\gamma)\mu_0(\mathrm{d}s') + \gamma \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)\pi'(\mathrm{d}a|s)P'(\mathrm{d}s'|s,a)\right) V^{\pi,P}(s') \\
&\quad - \int_{\mathcal{S}} \mu_\gamma^{\pi',P'}(\mathrm{d}s)V^{\pi,P}(s)
\end{aligned}
$$

$$
\begin{aligned}
&= \int_{\mathcal{S}} \mu_{\gamma}^{\pi',P'}(\mathrm{d}s) \left(\int_{\mathcal{A}} \pi'(\mathrm{d}a|s) \int_{\mathcal{S}} P'(\mathrm{d}s'|s,a) \left(r(s,a,s') + \gamma V^{\pi,P}(s') \right) - V^{\pi,P}(s) \right) \\
&\quad + \int_{\mathcal{S}} \mu_0(\mathrm{d}s') V^{\pi,P}(s') = \\
&= \int_{\mathcal{S}} \mu_{\gamma}^{\pi',P'}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s) + (1-\gamma) J^{\pi,P}, \qquad \text{(P.8)}
\end{aligned}
$$

where we have exploited the recursive formulation of $\mu_{\gamma}^{\pi',P'}$ (Definition 2.3) to rewrite line (P.6) into line (P.7) and line (P.8) follows by observing that $\int_{\mathcal{S}} \mu_0(\mathrm{d}s') V^{\pi,P}(s') = J^{\pi,P}$ and using the definition $U^{\pi,P}(s,a,s') = r(s,a,s') + \gamma V^{\pi,P}(s')$. □

This theorem is the natural extension of the result proposed by Kakade and Langford (2002). It essentially highlights that to compute the performance improvement we need to average the coupled relative advantage function $A_{\pi,P}^{\pi',P'}$ by means of the γ-discounted stationary distribution $\mu_{\gamma}^{\pi',P'}$ induced by the candidate policy-model pair (π', P').

Coupled Bound Unfortunately, the expression of Theorem 6.4 cannot be directly exploited in an algorithm as the dependence of $\mu_{\gamma}^{\pi',P'}$ on the candidate policy-transition model pair (π', P') is nonlinear and difficult to treat. We aim to obtain, from this result, a lower bound on $J^{\pi',P'} - J^{\pi,P}$ that can be efficiently computed using the information on the current pair (π, P). Before moving to the main result, we introduce an auxiliary result due to (Haviv and Van der Heyden, 1984) that we report in our notation without proof.

Lemma 6.5 (Corollary 2.4 of Haviv and Van der Heyden (1984))**.** *Let $\mu, \nu \in \mathscr{P}(\mathcal{X})$ be two probability measures and let $f \in \mathscr{B}(\mathcal{X})$ be a measurable function. Then, it holds that:*

$$
\left| \int_{\mathcal{X}} (\mu(\mathrm{d}x) - \nu(\mathrm{d}x)) f(x) \right| \leqslant \|\mu - \nu\|_{\mathrm{TV}} \operatorname{sp}(f),
$$

where $\operatorname{sp}(f) = \sup_{x \in \mathcal{X}} \{f(x)\} - \inf_{x \in \mathcal{X}} \{f(x)\}$.

We are now ready to prove the main result.

Theorem 6.6 (Coupled Bound)**.** *Let $\mathcal{C}$ be a Conf-MDP. The performance improvement of policy-transition model pair $(\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ over $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ can be lower bounded as:*

$$
\underbrace{J^{\pi',P'} - J^{\pi,P}}_{\substack{\textit{performance} \\ \textit{improvement}}} \geqslant \underbrace{\frac{1}{1-\gamma} \mathbb{A}_{\pi,P}^{\pi',P'}}_{\textit{advantage}} - \underbrace{\frac{\gamma}{(1-\gamma)^2} \operatorname{sp}\left(A_{\pi,P}^{\pi',P'} \right) \left\| (P')^{\pi'} - P^{\pi} \right\|_{\mathrm{TV},\mu_{\gamma}^{\pi,P}}}_{\textit{dissimilarity penalization}}.
$$

Proof. Exploiting the bounds on the γ-discounted state distributions difference (Proposition 6.2) we can easily attain the performance improvement bound:

$$
\begin{aligned}
J^{\pi',P'} - J^{\pi,P} &= \frac{1}{1-\gamma} \int_{\mathcal{S}} \mu_{\gamma}^{\pi',P'}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s) \\
&= \frac{1}{1-\gamma} \int_{\mathcal{S}} \mu_{\gamma}^{\pi,P}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s) \\
&\quad + \frac{1}{1-\gamma} \int_{\mathcal{S}} \left(\mu_{\gamma}^{\pi',P'}(\mathrm{d}s) - \mu_{\gamma}^{\pi,P}(\mathrm{d}s) \right) A_{\pi,P}^{\pi',P'}(s) \qquad \text{(P.9)}
\end{aligned}
$$

$$\geqslant \frac{\mathbb{A}_{\pi,P}^{\pi',P'}}{1-\gamma} - \frac{1}{1-\gamma}\left|\int_{\mathcal{S}}\left(\mu_\gamma^{\pi',P'}(\mathrm{d}s) - \mu_\gamma^{\pi,P}(\mathrm{d}s)\right)A_{\pi,P}^{\pi',P'}(s)\right| \tag{P.10}$$

$$\geqslant \frac{\mathbb{A}_{\pi,P}^{\pi',P'}}{1-\gamma} - \frac{1}{1-\gamma}\left\|\mu_\gamma^{\pi',P'} - \mu_\gamma^{\pi,P}\right\|_{\mathrm{TV}} \mathrm{sp}\left(A_{\pi,P}^{\pi',P'}\right) \tag{P.11}$$

$$\geqslant \frac{\mathbb{A}_{\pi,P}^{\pi',P'}}{1-\gamma} - \frac{\gamma}{(1-\gamma)^2}\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \mathrm{sp}\left(A_{\pi,P}^{\pi',P'}\right), \tag{P.12}$$

where line (P.10) follows from line (P.9) by observing that $b \geqslant -|b|$ for any $b \in \mathbb{R}$, line (P.11) follows from (P.10) by applying Lemma 6.5 and line (P.12) is obtained by using Corollary 6.2. □

The bound is composed of two terms, like in (Kakade and Langford, 2002; Pirotta et al., 2013b): the first term, *advantage*, represents how much gain in performance can be locally obtained by moving from (π, P) to (π', P'), whereas the second term, *dissimilarity penalization*, discourages updates towards policy-model pairs that are too far away.

Decoupled Bound As we mentioned in Chapter 4, in several cases of interest, the possibility to act on the transition model is constrained, while the policy being under the complete control of the agent. In other cases, although less frequent, the control on the policy might be limited while the transition model can be changed arbitrarily. In both scenarios, however, it seems quite impractical to account for these limitations when the learning process is carried out on the state transition kernel P^π directly. This makes the *coupled bound* unsuitable in practice as it does not separate the contribution of the policy and that of the model. It is worth noting that the following derivations are slightly different compared to the ones presented in the original paper (Metelli et al., 2018a). This is because here we consider a reward function depending on the next state too $r(s, a, s')$ while in (Metelli et al., 2018a) only state-action rewards $r(s, a)$ were considered. We now present the uncoupled bound, whose complete derivation can be found in Appendix A.1.

Theorem 6.7 (Decoupled Bound)**.** *Let $\mathcal{C}$ be a Conf-MDP. The performance improvement of policy-transition model pair $(\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ over $(\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ can be lower bounded as:*

$$\underbrace{J^{\pi',P'} - J^{\pi,P}}_{\substack{\textit{performance}\\ \textit{improvement}}} \geqslant B(\pi', P') = \underbrace{\frac{1}{1-\gamma}\left(\mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'}\right)}_{\textit{advantage}} - \frac{2}{(1-\gamma)^2}$$

$$\times \underbrace{\left[\left\|P' - P\right\|_{\mathrm{TV},\infty}\left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \gamma\left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right)\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\mathrm{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right.}_{\textit{dissimilarity penalization}}$$

$$\underbrace{\left. + \gamma\left\|\pi' - \pi\right\|_{\mathrm{TV},\infty}\left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right)\sup_{s\in\mathcal{S}}\left\{\mathrm{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\right]}_{\textit{dissimilarity penalization}}.$$

Proof. We start from the coupled bound and we manage the three terms separately:

$$J^{\pi',P'} - J^{\pi,P} \geqslant \underbrace{\frac{\mathbb{A}_{\pi,P}^{\pi',P'}}{1-\gamma}}_{\text{(i)}} - \frac{\gamma}{(1-\gamma)^2}\underbrace{\left\|(P')^{\pi'} - P^{\pi}\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}_{\text{(ii)}}\underbrace{\mathrm{sp}\left(A_{\pi,P}^{\pi',P'}\right)}_{\text{(iii)}}.$$

(i) is bounded using Lemma A.1, (ii) is bounded using Lemma 6.3, and (iii) using Lemma A.2. Putting all together we have:

$$
\begin{aligned}
J^{\pi',P'} - J^{\pi,P} \geqslant & \frac{\mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'}}{1-\gamma} \\
& - \frac{2}{1-\gamma} \left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \left\|P' - P\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\} \\
& - \frac{\gamma}{(1-\gamma)^2} \left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right) \\
& \times 2\Bigg(\left\|\pi' - \pi\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\} \\
& + \left\|P' - P\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\}\Bigg).
\end{aligned}
$$

By rearranging the terms we get the result. □

Comparison with Existing Bounds We compare the bound of Theorem 6.7 with Theorem 3.3 of (Metelli et al., 2018a). In (Metelli et al., 2018a) only state-action reward functions $r(s,a)$ were considered. We claim that our bound reduces to that of (Metelli et al., 2018a) in such a case. Indeed, when the reward function is independent from the next state, we have $\mathrm{sp}\big(U^{\pi,P}(s,a,\cdot)\big) = \gamma \mathrm{sp}\big(V^{\pi,P}\big)$, by observing that $\mathrm{sp}\big(V^{\pi,P}\big) \leqslant \mathrm{sp}\big(Q^{\pi,P}\big)$ and $\sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\big(Q^{\pi,P}\big)(s,\cdot)\right\} \leqslant \mathrm{sp}\big(Q^{\pi,P}\big)$, we reduce exactly to the bound of Theorem 3.3 of (Metelli et al., 2018a):

$$
\begin{aligned}
J^{\pi',P'} - J^{\pi,P} \geqslant & \frac{1}{1-\gamma}\left(\mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'}\right) - \frac{2\gamma}{(1-\gamma)^2} \\
\times & \Bigg(\left\|P' - P\right\|_{\mathrm{TV},\infty} \left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \gamma \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right) \\
& + \left\|\pi' - \pi\right\|_{\mathrm{TV},\infty} \left(\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} + \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}\right)\Bigg) \mathrm{sp}\big(Q^{\pi,P}\big).
\end{aligned}
$$

It is also worthwhile to analyze the form of the bound when either $P' = P$ or $\pi' = \pi$, i.e., when we change alternatively either the policy or the transition model but not both. The following corollary provides the expression of the decoupled bound.

Corollary 6.8. *Let $\mathcal{C}$ be a Conf-MDP and let $(\pi', P'), (\pi, P) \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ two policy-transition models pairs. The performance improvement of policy π' over π, under transition model P, can be lower bounded as:*

$$
\begin{aligned}
J^{\pi',P} - J^{\pi,P} \geqslant & \frac{1}{1-\gamma}\mathbb{A}_{\pi,P}^{\pi',P} \\
& - \frac{2\gamma}{(1-\gamma)^2} \left\|\pi' - \pi\right\|_{\mathrm{TV},\infty} \left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\big(Q^{\pi,P}(s,\cdot)\big)\right\}.
\end{aligned}
$$

Furthermore, the performance improvement of transition model P' over P, under policy π, can be lower bounded as:

$$
J^{\pi,P'} - J^{\pi,P} \geqslant \frac{1}{1-\gamma}\mathbb{A}_{\pi,P}^{\pi,P'}
$$

$$- \frac{2\gamma}{(1-\gamma)^2} \left\|P' - P\right\|_{\mathrm{TV},\infty} \left\|P' - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\big(U^{\pi,P}(s,a,\cdot)\big)\right\}.$$

It is worth comparing the first bound of Corollary 6.8 with the bound of Corollary 3.6 of (Pirotta et al., 2013b). We observe that our bound is tighter for two reasons. First, it employs as policy dissimilarity the product $\|\pi' - \pi\|_{\mathrm{TV},\infty} \|\pi' - \pi\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \leqslant \|\pi' - \pi\|^2_{\mathrm{TV},\infty}$ used in (Pirotta et al., 2013b). Second, the term involving the Q-function is tighter since $\sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\big(Q^{\pi,P}(s,\cdot)\big)\right\} \leqslant \mathrm{sp}\big(Q^{\pi,P}\big)$.

6.4 Safe Policy Model Iteration

To deal with the learning problem in the Conf-MDP framework we could, in principle, learn the optimal policy by using a classical RL algorithm and adapt it to learn the optimal model, sequentially or in parallel. Alternatively, we could resort to general-purpose global optimization tools, like CEM (Rubinstein, 1999) or genetic algorithms (Holland and Goldberg, 1989), using as objective function the performance of the policy learned by a standard RL algorithm. Nonetheless, they may not correspond to the preferable, nor the safest, choices in this context as there exists an inherent connection between policy and model we could not overlook during the learning process. Indeed, a policy learned by interacting with a sub-optimal model could result in poor performance paired with a different, maybe optimal, model. At the same time, a policy far from the optimum could mislead the search of the optimal model. The goal of this section is to present an approach, *Safe Policy-Model Iteration* (SPMI), inspired to (Pirotta et al., 2013b), capable of learning the policy and the model simultaneously, possibly taking advantage of the inter-connection mentioned above.

Following the approach proposed in (Pirotta et al., 2013b), we define the policy and model improvement update rules:

$$\pi' = \alpha\overline{\pi} + (1-\alpha)\pi,$$
$$P' = \beta\overline{P} + (1-\beta)P,$$

where $\alpha, \beta \in [0,1]$, $\overline{\pi} \in \Pi^{\mathrm{SR}}$ and $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$ are the target policy and the target transition model respectively. Extending the rationale of (Pirotta et al., 2013b) to our context, we aim to determine the values of α and β which jointly maximize the *decoupled bound* (Theorem 6.7). In the following, for the sake of clarity, we will abbreviate the decoupled bound $B(\pi', P')$ with $B(\alpha, \beta)$. The following result states a notable condition for the optimization of the lower bound.

Theorem 6.9. *For any $\overline{\pi} \in \Pi^{\mathrm{SR}}$ and $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$, the decoupled bound is optimized for:*

$$(\alpha^*, \beta^*) \in \arg\max_{(\alpha,\beta)\in\mathcal{V}} \{B(\alpha,\beta)\},$$

where B is the bound in Theorem 6.7 and $\mathcal{V} = \{(\alpha_0^*, 0), (\alpha_1^*, 1), (0, \beta_0^*), (1, \beta_1^*)\}$ and:

$$\alpha_0^* = \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{4\gamma \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\right\} \|\overline{\pi} - \pi\|_{\mathrm{TV},\infty} \|\overline{\pi} - \pi\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}},$$

Algorithm 6.1: Safe Policy Model Iteration (SPMI).

Input: Conf-MDP $\mathcal{C}$, number of iterations T
Output: approximately optimal policy-transition model pair $(\pi^{(T)}, P^{(T)})$

Initialize π_0, P_0 arbitrarily
forall $i = 0, 1, \ldots, T-1$ **do**
  $\overline{\pi}^{(i)} = \textit{PolicyChooser}(\pi^{(i)})$
  $\overline{P}^{(i)} = \textit{ModelChooser}(P^{(i)})$
  $\mathcal{V}^{(i)} = \{(\alpha^*_{0,i}, 0), (\alpha^*_{1,i}, 1), (0, \beta^*_{0,i}), (1, \beta^*_{1,i})\}$
  $\alpha^*_i, \beta^*_i = \arg\max_{(\alpha,\beta)\in\mathcal{V}^{(i)}} \{B(\alpha, \beta)\}$
  $\pi^{(i+1)} = \alpha^*_i \overline{\pi}^{(i)} + (1 - \alpha^*_i)\pi^{(i)}$
  $P^{(i+1)} = \beta^*_i \overline{P}^{(i)} + (1 - \beta^*_i)P^{(i)}$
return $(\pi^{(T)}, P^{(T)})$

$$\alpha^*_1 = \alpha^*_0 - \frac{\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu^{\pi,P}_\gamma}}{2\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu^{\pi,P}_\gamma}} - \frac{\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\mathrm{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}}{2\gamma\sup_{s\in\mathcal{S}}\left\{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\right\}\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}},$$

$$\beta^*_0 = \frac{(1-\gamma)\mathbb{A}^{\pi,\overline{P}}_{\pi,P}}{4\gamma\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\right\}\left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu^{\pi,P}_\gamma}},$$

$$\beta^*_1 = \beta^*_0 - \frac{\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu^{\pi,P}_\gamma}}{2\gamma\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu^{\pi,P}_\gamma}} - \frac{\sup_{s\in\mathcal{S}}\left\{\mathrm{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}}{2\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\right\}\left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}},$$

to be clipped in the interval $[0, 1]$.

The proof of the theorem can be found in Appendix A.1. The theorem shows that the optimal (α, β) pair lies on the boundary of $[0, 1] \times [0, 1]$, i.e., either one between policy and model is moved and the other is kept unchanged or one is moved and the other is set to the target.

Algorithm 6.1 reports the pseudocode of SPMI. The procedures *PolicyChooser* and *ModelChooser* are designated for selecting the target policy and model (see Section 6.4.4). In the following subsections, we briefly discuss two simplifications of the SPMI algorithm in which we either keep the transition model fixed and update the policy, *Safe Policy Iteration* (SPI, Section 6.4.1) or we keep the policy fixed and update the transition model, *Safe Model Iteration* (SMI, Section 6.4.2).

6.4.1 Safe Policy Iteration

Safe Policy Iteration (SPI) is essentially the Unique-parameter SPI of (Pirotta et al., 2013b), with the only difference that we employ the bound of Corollary 6.8 that is tighter. The ultimate goal consists in finding an optimal policy under the fixed model $P \in \mathcal{P}^{\mathrm{SR}}$, i.e., $\pi^* \in \arg\max_{\pi\in\Pi^{\mathrm{SR}}}\left\{J^{\pi,P}\right\}$. The policy improvement rule is given by:

$$\pi' = \alpha\overline{\pi} + (1 - \alpha)\pi,$$

where $\alpha \in [0, 1]$ and $\overline{\pi} \in \Pi^{\mathrm{SR}}$ is the target policy chosen by a suitable *Policy Chooser* function. The following result provides the optimal value of the coefficient α and the

Algorithm 6.2: Safe Policy Iteration (SPI).

Input: Conf-MDP $\mathcal{C}$, number of iterations T
Output: approximately optimal policy $\pi^{(T)}$
Initialize π_0 arbitrarily
forall $i = 0, 1, \ldots, T-1$ **do**
  $\overline{\pi}^{(i)} = \textit{PolicyChooser}(\pi^{(i)})$
  $\pi^{(i+1)} = \alpha_i^* \overline{\pi}^{(i)} + (1 - \alpha_i^*)\pi^{(i)}$
return $\pi^{(T)}$

Algorithm 6.3: Safe Model Iteration (SMI).

Input: Conf-MDP $\mathcal{C}$, number of iterations T
Output: approximately optimal transition model $P^{(T)}$
Initialize P_0 arbitrarily
forall $i = 0, 1, \ldots, T-1$ **do**
  $\overline{P}^{(i)} = \textit{ModelChooser}(P^{(i)})$
  $P^{(i+1)} = \beta_i^* \overline{P}^{(i)} + (1 - \beta_i^*)P^{(i)}$
return $P^{(T)}$

corresponding performance improvement, while the pseudocode of SPI is reported in Algorithm 6.2.

Corollary 6.10. *For any $\overline{\pi} \in \Pi^{\mathrm{SR}}$ the first bound of Corollary 6.8 is optimized for:*

$$\alpha^* = \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{4\gamma \sup_{s \in \mathcal{S}} \{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}},$$

to be clipped in the interval $[0,1]$. In such a case, the performance improvement can be lower bounded as:

$$J^{\pi',P'} - J^{\pi,P} \geqslant \frac{\left(\mathbb{A}_{\pi,P}^{\overline{\pi},P}\right)^2}{8\gamma \sup_{s \in \mathcal{S}} \{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}.$$

Proof. The proof is obtained from Theorem 6.9, by simply setting $\beta = 0$ and substituting the optimal value α^* in the performance improvement bound. □

6.4.2 Safe Model Iteration

Analogously to SPI, we can devise a corresponding version for the transition model, named Safe Model Iteration (SMI). Here, the goal consists in finding an optimal transition model, under the fixed policy $\pi \in \Pi^{\mathrm{SR}}$, i.e., $P^* \in \arg\max_{P \in \mathcal{P}^{\mathrm{SR}}} \left\{J^{\pi,P}\right\}$. The update rule is still obtained by means of a convex combination:

$$P' = \beta \overline{P} + (1-\beta)P,$$

where $\beta \in [0,1]$ and $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$ is the target model. The optimal value of the coefficient β as well as the performance improvement are provided in the following result, whereas the pseudocode of SMI is reported in Algorithm 6.3.

Corollary 6.11. *For any $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$ the second bound of Corollary 6.8 is optimized for:*

$$\beta^* = \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\pi,\overline{P}}}{4\gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}} \{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\} \left\|\overline{P}-P\right\|_{\mathrm{TV},\infty} \left\|\overline{P}-P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}},$$

to be clipped in the interval $[0,1]$. In such a case, the performance improvement can be lower bounded as:

$$J^{\pi',P'} - J^{\pi,P} \geqslant \frac{\left(\mathbb{A}_{\pi,P}^{\pi,\overline{P}}\right)^2}{8\gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}} \{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\} \left\|\overline{P}-P\right\|_{\mathrm{TV},\infty} \left\|\overline{P}-P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}.$$

Proof. The proof is analogous to that of Corollary 6.10. □

6.4.3 Policy and Model Spaces

The selection of the target policy and model is a rather crucial component of the algorithm since the quality of the updates largely depends on it. To effectively adopt a target selection strategy we need to be aware of the degrees of freedom on the policy and model spaces. Focusing on the model space first, it is easy to discriminate two macro-classes.

Unconstrained In some cases, there are almost no constraints on the direction in which to update the model. In these scenarios, we can naturally design the first scenario as an *unconstrained* model space and choosing $\mathcal{P} = \mathcal{P}^{\mathrm{SR}}$ the space of all Markovian stationary transition models.

Parametric In other cases, only a limited model portion, typically a set of parameters inducing transition probabilities, can be accessed. To represent this case, we limit the model space to a parametric set $\mathcal{P}_\Omega = \{P_{\boldsymbol{\omega}} : \boldsymbol{\omega} \in \Omega \in \mathbb{R}^q\}$, as we have seen in Section 5.1.5. A particular choice, that turns out to be convenient for SPMI (especially in the analysis), is the *convex hull* of a set of vertex (or extreme) models (e.g., a set of deterministic models) $\mathcal{P}_{\mathrm{vtx}} = \{P_1, \dots, P_M\}$, with $M \in \mathbb{N}_{\geqslant 1}$:

$$\mathcal{P}_\Omega = \mathrm{co}(\mathcal{P}_{\mathrm{vtx}}) = \left\{ P_{\boldsymbol{\omega}} = \sum_{i=1}^{M} \omega_i P_i,\ \omega_i \geqslant 0,\ \forall i \in \{1,\dots,n\},\ \sum_{i=1}^{M} \omega_i = 1 \right\}.$$

It is worth noting that if we select as set of vertex models all the Markovian stationary deterministic transition models, i.e., $\mathcal{P}_{\mathrm{vtx}} = \mathcal{P}^{\mathrm{SD}}$, we have that $\mathrm{co}(\mathcal{P}_{\mathrm{vtx}}) = \mathcal{P}^{\mathrm{SR}}$.

It is noteworthy that we can symmetrically extend the dichotomy to the policy space, although the need for limiting the agent on the direction of policy updates is less relevant in our perspective.

6.4.4 Target Choice

Up to now we have not specified the form of the *PolicyChooser* and *ModelChooser* functions, in charge of outputting the target policy and the target model. To deal with unconstrained spaces, it is quite natural to adopt the target selection strategy presented in (Pirotta et al., 2013b), choosing the greedy model, defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$P^+(s, a) \in \arg\max_{s' \in \mathcal{S}} \{U^{\pi,P}(s, a, s')\},$$

that corresponds to the model that maximizes the relative advantage in each state-action pair. Extending this rationale to the policy space, we obtain an algorithm in which, at each step, the greedy policy and model w.r.t. the $Q^{\pi,P}$ and $U^{\pi,P}$ are selected as targets.

When we are not free to choose the greedy model, like in the parametric setting, because the greedy model might not belong to the space of representable transition models $\mathcal{P}_\Omega$, we can resort to a relaxed notion of greedy model, as the one maximizing the expected relative advantage function:

$$\overline{P} \in \arg\max_{\overline{P} \in \mathcal{P}^{\mathrm{SR}}} \left\{ \mathbb{A}^{\pi,\overline{P}}_{\pi,P} \right\}.$$

This *greedy choice* is based on local information and is not guaranteed to provide a policy-transition model pair maximizing the bound. Nevertheless, testing all the policy-transition model pairs is highly inefficient in the presence of large policy-transition model spaces. To mitigate this effect, a reasonable compromise is to select, as a target, the model that yields the maximum bound value between the greedy target and the previous target. This procedure, named *persistent choice*, effectively avoids the oscillating behavior, common with the greedy choice Wagner (2011).

6.5 Theoretical Analysis

In this section, we outline some relevant theoretical results related to SPMI. We start by analyzing the scenario in which the model/policy space is parametric and limited to the convex hull of a set of vertex models/policies, and then we provide some rationales for the target choices adopted. In most of the section, we restrict our attention to the transition model, as for the policy all results apply symmetrically. For this reason, we will remove the dependence on the policy from the relevant quantities, whenever not generating confusion.

6.5.1 Convex Hull Model Space

We consider the setting in which the transition model space is limited to the convex hull of a finite set of vertex models: $\mathcal{P}_\Omega = \mathrm{co}(\mathcal{P}_{\mathrm{vtx}})$, where $\mathcal{P}_{\mathrm{vtx}} = \{P_1, \ldots, P_M\}$. For the sake of brevity, we omit the dependency on π of all the quantities and we abbreviate $J^{\pi,P_{\boldsymbol{\omega}}}$ as $J(\boldsymbol{\omega})$. We define an optimal transition model $P_{\boldsymbol{\omega}*}$ as any model that maximizes the expected return, i.e., $J(\boldsymbol{\omega}^*) \geqslant J(\boldsymbol{\omega})$ for all $P_{\boldsymbol{\omega}} \in \mathrm{co}(\mathcal{P}_{\mathrm{vtx}})$. We start by stating some results on the expected relative advantage functions.

Lemma 6.12. *Let $P_{\boldsymbol{\omega}} \in \mathrm{co}(\mathcal{P}_{vtx})$ be a transition model, where $\mathcal{P}_{vtx} = \{P_1, \dots, P_M\}$. Then, for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ it holds that:*

$$\sum_{i=1}^{M} \omega_i A^{P_i}_{P_{\boldsymbol{\omega}}}(s,a) = 0.$$

Proof. Let us rewrite the expected relative advantage by decomposing $P_{\boldsymbol{\omega}}$:

$$\begin{aligned} A^{P_i}_{P_{\boldsymbol{\omega}}}(s,a) &= \int_{\mathcal{S}} \left(P_i(\mathrm{d}s'|s,a) - P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a)\right) U^{P_{\boldsymbol{\omega}}}(s,a,s') \\ &= \int_{\mathcal{S}} \left(P_i(\mathrm{d}s'|s,a) - \sum_{j=1}^{M} \omega_j P_j(\mathrm{d}s'|s,a)\right) U^{P_{\boldsymbol{\omega}}}(s,a,s'). \end{aligned}$$

Now we take the weighted sum of the previous equation:

$$\begin{aligned} \sum_{i=1}^{M} \omega_i A^{P_i}_{P_{\boldsymbol{\omega}}}(s,a) &= \sum_{i=1}^{M} \omega_i \int_{\mathcal{S}} \left(P_i(\mathrm{d}s'|s,a) - \sum_{j=1}^{M} \omega_j P_j(\mathrm{d}s'|s,a)\right) U^{P_{\boldsymbol{\omega}}}(s,a,s') \\ &= \int_{\mathcal{S}} \left(\sum_{i=1}^{M} \omega_i P_i(\mathrm{d}s'|s,a) - \sum_{j=1}^{M} \omega_j P_j(\mathrm{d}s'|s,a)\right) U^{P_{\boldsymbol{\omega}}}(s,a,s') = 0, \end{aligned}$$

where we just observed that $\sum_{i=1}^{M} \omega_i P_i(\mathrm{d}s'|s,a) - \sum_{j=1}^{M} \omega_j P_j(\mathrm{d}s'|s,a) = 0$. □

As a consequence, we observe that also the expected relative advantage functions $\mathbb{A}^{P_i}_{P_{\boldsymbol{\omega}}}$ sum up to zero when weighted by the coefficients $\boldsymbol{\omega}$. An analogous statement holds when the policy is defined as a convex combination of vertex policies. The following theorem establishes an essential property of the optimal transition model.

Theorem 6.13. *Let $P_{\boldsymbol{\omega}} \in \mathrm{co}(\mathcal{P}_{vtx})$ be a transition model, where $\mathcal{P}_{vtx} = \{P_1, \dots, P_M\}$. Then, it holds that $\mathbb{A}^{P_{\boldsymbol{\omega}}}_{P_{\boldsymbol{\omega}*}} \leqslant 0$. Moreover, for all $P_{\boldsymbol{\omega}} \in \mathrm{co}\left(\{P_i \in \mathcal{P}_{vtx} : \omega_i^* > 0\}\right)$, it holds that $\mathbb{A}^{P_{\boldsymbol{\omega}}}_{P_{\boldsymbol{\omega}*}} = 0$.*

Proof. We first prove that the expected relative advantage w.r.t. the vertex models is non-positive and then we extend it to all the models. By contradiction, suppose there exists a vertex model $P_i \in \mathcal{P}_{\text{vtx}}$ having a positive expected relative advantage. Then, we can perform a step of model update with SPMI starting from $P_{\boldsymbol{\omega}*}$ and getting the new model $P_{\overline{\boldsymbol{\omega}}}$ with a performance improvement of at least (Corollary 6.11):

$$\begin{aligned} &J(\boldsymbol{\omega}') - J(\boldsymbol{\omega}^*) \\ &\geqslant \frac{\left(\mathbb{A}^{P_i}_{P_{\boldsymbol{\omega}*}}\right)^2}{8\gamma \sup_{s\in\mathcal{S}, a\in\mathcal{A}} \left\{\mathrm{sp}\left(U^{P_{\boldsymbol{\omega}*}}(s,a,\cdot)\right)\right\} \left\|P_i - P_{\boldsymbol{\omega}*}\right\|_{\mathrm{TV},\infty} \left\|P_i - P_{\boldsymbol{\omega}*}\right\|_{\mathrm{TV},\mu^{\pi,P}_{\gamma}}} > 0, \end{aligned}$$

which is impossible as $P_{\boldsymbol{\omega}*}$ is the optimal model. Let us consider a generic model $P_{\boldsymbol{\omega}}$, its advantage decomposes linearly in the vertex models:

$$\mathbb{A}^{P_{\boldsymbol{\omega}}}_{P_{\boldsymbol{\omega}*}} = \sum_{i=1}^{M} \omega_i \mathbb{A}^{P_i}_{P_{\boldsymbol{\omega}*}} \leqslant 0.$$

Let us now consider the subset of vertex models having non-zero coefficient for the optimal model $\{P_i \in \mathcal{P}_{\text{vtx}} : \omega_i^* > 0\}$. From Lemma 6.12 we have:

$$\sum_{i=1}^{M} \omega_i^* \mathbb{A}_{P_{\boldsymbol{\omega}}*}^{P_i} = \sum_{i:\omega_i^*>0} \omega_i^* \mathbb{A}_{P_{\boldsymbol{\omega}}*}^{P_i} = 0. \tag{P.13}$$

Since $\mathbb{A}_{P_{\boldsymbol{\omega}}*}^{P_i} \leqslant 0$ from the first part of the theorem, it must be that all $\mathbb{A}_{P_{\boldsymbol{\omega}}*}^{P_i} = 0$. As an immediate consequence, all transition models in $\text{co}\left(\{P_i \in \mathcal{P}_{\text{vtx}} : \omega_i^* > 0\}\right)$ must have zero expected relative advantage, due to the linear decomposition of the advantage. □

The theorem provides a necessary condition for a transition model to be optimal, i.e., all the expected relative advantages must be non-positive and, moreover, those of the vertex transition models associated with non-zero coefficients must be zero. It is worth noting that the expected relative advantage $\mathbb{A}_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}'}}$ represents only a *local* index of the performance improvement, as it is defined by taking the expectation of the relative advantage $A_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}'}}(s,a)$ w.r.t. the current $\mu_\gamma^{P_{\boldsymbol{\omega}}}$. On the other hand, the actual performance improvement $J(\boldsymbol{\omega}') - J(\boldsymbol{\omega})$ is a *global* index, being obtained by averaging the relative advantage $A_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}'}}(s,a)$ w.r.t. the new $\mu_\gamma^{P_{\boldsymbol{\omega}'}}$ (Theorem 6.4). This is intimately related to the *measure mismatch* claim provided in (Kakade, 2003) as the model expected relative advantage $\mathbb{A}_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}}*}$ might be null even if $J(\boldsymbol{\omega}^*) > J(\boldsymbol{\omega})$, making SPMI, just like CPI and SPI, stop into locally optimal models. Furthermore, it is simple to see that asking for a guaranteed performance improvement may prevent from finding the global optimum, as this may require visiting a lower performance region (see Section 6.7.1 for an example). Nevertheless, we can provide a bound for the performance gap between a locally optimal model and the global optimal model.

Proposition 6.14. *Let $P_{\overline{\boldsymbol{\omega}}} \in \text{co}(\mathcal{P}_{vtx})$ be a transition model, where $\mathcal{P}_{vtx} = \{P_1, \ldots, P_M\}$. If for all $P_i \in \mathcal{P}_{vtx}$ it holds that the expected relative advantage function is non-positive, i.e., $\mathbb{A}_{P_i}^{P_{\boldsymbol{\omega}}*} \leqslant 0$, then it holds that:*

$$J(\boldsymbol{\omega}^*) - J(\overline{\boldsymbol{\omega}}) \leqslant \frac{1}{1-\gamma} \sup_{s\in\mathcal{S}, a\in\mathcal{A}} \left\{ \max_{i\in\{1,\ldots,M\}} \left\{ A_{P_{\overline{\boldsymbol{\omega}}}}^{P_i}(s,a) \right\} \right\}.$$

Proof. Using Theorem 6.4 and Lemma 6.1 we can write:

$$\begin{aligned} J(\boldsymbol{\omega}^*) - J(\overline{\boldsymbol{\omega}}) &= \frac{1}{1-\gamma} \int_{\mathcal{S}} \mu^{P_{\boldsymbol{\omega}}*}(\mathrm{d}s) \int_{\mathcal{A}} \pi(\mathrm{d}a|s) A_{P_{\overline{\boldsymbol{\omega}}}}^{P_{\boldsymbol{\omega}}*}(s,a) \\ &\leqslant \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu^{P_{\boldsymbol{\omega}}*}(\mathrm{d}s, \mathrm{d}a) \sup_{s\in\mathcal{S}, a\in\mathcal{A}} \left\{ A_{P_{\overline{\boldsymbol{\omega}}}}^{P_{\boldsymbol{\omega}}*}(s,a) \right\} \\ &\leqslant \frac{1}{1-\gamma} \sup_{s\in\mathcal{S}, a\in\mathcal{A}} \left\{ A_{P_{\overline{\boldsymbol{\omega}}}}^{P_{\boldsymbol{\omega}}*}(s,a) \right\}, \end{aligned}$$

Now we observe that the relative advantage decomposes linearly in the target models:

$$A_{P_{\overline{\boldsymbol{\omega}}}}^{P_{\boldsymbol{\omega}}*}(s,a) = \sum_{i=1}^{M} \omega_i^* A_{P_{\overline{\boldsymbol{\omega}}}}^{P_i}(s,a) \leqslant \max_{i\in\{1,\ldots,M\}} \left\{ A_{P_{\overline{\boldsymbol{\omega}}}}^{P_i}(s,a) \right\},$$

from which the theorem follows. □

From this result, we notice that a sufficient condition for a model to be optimal is that $A^{P_i}_{P_{\overline{\omega}}}(s,a) = 0$ for all state-action pairs. This is a stronger requirement than the maximization of $J^{P_{\overline{\omega}}}$ as it asks the model to be optimal in *every* state-action pair independently of the initial state distribution μ_0;[1] such a model might not exist when considering a model space $\mathcal{P}$ that does not include all the possible transition models (see Section 6.7 for an example).

6.5.2 P-Gradient Theorem

In this section, we elucidate the relationship between the relative advantage function and the gradient of the expected return. Let us start by stating the expression of the gradient of the expected return w.r.t. a parametric transition model. This is the equivalent of the Policy Gradient Theorem (Sutton et al., 1999a) for the transition model.

Theorem 6.15 (P-Gradient Theorem)**.** *Let $\mathcal{P}_\Omega = \{P_{\boldsymbol{\omega}} : \boldsymbol{\omega} \in \Omega \in \mathbb{R}^q\}$ be a set of parametric stochastic transition models differentiable in $\boldsymbol{\omega} \in \Omega$. Then, the gradient of the expected return $J(\boldsymbol{\omega})$ w.r.t. $\boldsymbol{\omega}$ is given by:*

$$\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}) = \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_\gamma^{P_{\boldsymbol{\omega}}}(\mathrm{d}s, \mathrm{d}a) \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) U^{P_{\boldsymbol{\omega}}}(s,a,s').$$

Proof. We just rephrase the proof of the Policy Gradient Theorem (Sutton et al., 1999a). Let us compute the gradient of the Q-function for any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
\nabla_{\boldsymbol{\omega}} Q^{P_{\boldsymbol{\omega}}}(s,a) &= \nabla_{\boldsymbol{\omega}} \int_{\mathcal{S}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) U^{P_{\boldsymbol{\omega}}}(s,a,s') \\
&= \int_{\mathcal{S}} \Big(\nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) U^{P_{\boldsymbol{\omega}}}(s,a,s') + P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) \nabla_{\boldsymbol{\omega}} U^{P_{\boldsymbol{\omega}}}(s,a,s') \Big) \quad \text{(P.14)} \\
&= \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) U^{P_{\boldsymbol{\omega}}}(s,a,s') \\
&\quad + \int_{\mathcal{S}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) \nabla_{\boldsymbol{\omega}} \left(r(s,a,s') + \gamma \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') Q^{P_{\boldsymbol{\omega}}}(s',a') \right) \quad \text{(P.15)} \\
&= \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) U^{P_{\boldsymbol{\omega}}}(s,a,s') \\
&\quad + \gamma \int_{\mathcal{S}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s,a) \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') \nabla_{\boldsymbol{\omega}} Q^{P_{\boldsymbol{\omega}}}(s',a'), \quad \text{(P.16)}
\end{aligned}$$

where (P.15) follows from (P.14) by expressing the U-function with the corresponding Bellman equation. After unfolding (P.16) we get:

$$\nabla_{\boldsymbol{\omega}} Q^{P_{\boldsymbol{\omega}}}(s,a) = \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu^{P_{\boldsymbol{\omega}}}_{\delta_{(s,a)},\gamma}(\mathrm{d}s'', \mathrm{d}a'') \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(\mathrm{d}s'|s'',a'') U^{P_{\boldsymbol{\omega}}}(s'',a',s'),$$

where $\mu^{P_{\boldsymbol{\omega}}}_{\delta_{(s,a)},\gamma}$ is the γ-discounted state-action distribution when forcing the first state to be s and the first action to be a. We obtain the gradient of the expected return by observing that $J(\boldsymbol{\omega}) = \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_0(\mathrm{d}s)\pi(\mathrm{d}a|s) Q^{P_{\boldsymbol{\omega}}}(s,a)$ and therefore:

$$\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}) = \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_0(\mathrm{d}s)\pi(\mathrm{d}a|s) \nabla_{\boldsymbol{\omega}} Q^{P_{\boldsymbol{\omega}}}(s,a)$$

[1] This is the same difference between a policy that maximizes the value function V^π in all states and a policy that maximizes the expected return J^π.

$$= \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_{\gamma}^{P_{\boldsymbol{\omega}}}(\mathrm{d}s'', \mathrm{d}a'') \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(s'|s'', a'') U^{P_{\boldsymbol{\omega}}}(s'', a'', s'),$$

by observing that $\int_{\mathcal{S}} \int_{\mathcal{A}} \mu_0(\mathrm{d}s)\pi(\mathrm{d}a|s)\mu^{P_{\boldsymbol{\omega}}}_{\delta_{(s,a)},\gamma}(\mathrm{d}s'', \mathrm{d}a'') = \mu^{P_{\boldsymbol{\omega}}}_{\mu_0,\gamma}(\mathrm{d}s'', \mathrm{d}a'')$ and we agreed to omit the subscript μ_0. By remanding the integration variables we get the result. □

Let us now show the connection between $\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega})$ and the expected relative advantage functions. This result extends that of Kakade (2003) to the case of the transition model.

Proposition 6.16. *Let $P \in \mathcal{P}^{\mathrm{SR}}$ be the current transition model and $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$ be the target transition model. Let us consider the update rule:*

$$P' = \beta \overline{P} + (1-\beta)P,$$

with $\beta \in [0, 1]$. Then, the derivative of the expected return of P' w.r.t. the β coefficients evaluated in P is given by:

$$\left.\frac{\partial J^{P'}}{\partial \beta}\right|_{\beta=0} = \frac{1}{1-\gamma} \mathbb{A}^{\overline{P}}_{P}.$$

Proof. Exploiting Theorem 6.15 and the definition of P' we can write the expression of the gradient:

$$\begin{aligned}\frac{\partial J^{P'}}{\partial \beta} &= \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_{\gamma}^{P'}(\mathrm{d}s, \mathrm{d}a) \int_{\mathcal{S}} \frac{\partial}{\partial \beta} P'(\mathrm{d}s'|s, a) U^{P'}(s, a, s') \\ &= \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \mu_{\gamma}^{P'}(\mathrm{d}s, \mathrm{d}a) \int_{\mathcal{S}} \left(\overline{P}(\mathrm{d}s'|s, a) - P(\mathrm{d}s'|s, a)\right) U^{P'}(s, a, s').\end{aligned}$$

The result immediately follows by observing that $P'|_{\beta=0} = P$. □

The proposition provides an interesting interpretation of the expected relative advantage function. Suppose that $P_{\boldsymbol{\omega}}$ is the current model and we have to choose which update direction (target model) to follow. If we consider the target model as a convex combination of a set of vertex models $\mathcal{P}_{\mathrm{vtx}}$, i.e., $\overline{P} = \sum_{i=1}^{M} \eta_i P_i$, the local performance improvement, at the first order, is given by $J^{P'} - J^{P} \simeq \left.\frac{\partial J^{P'}}{\partial \beta}\right|_{\beta=0} \beta = \frac{1}{1-\gamma} \beta \sum_{i=1}^{M} \eta_i \mathbb{A}^{P_i}_{P}$. Given that β will be determined later by maximizing the bound, the local performance improvement is maximized by assigning one to the coefficient of the model yielding the maximal advantage. Therefore, the choice of the direction to follow, when considering the greedy target choice, is based on local information only (gradient), while the step size β is obtained by maximizing the bound on the guaranteed performance improvement (safe), as done in (Pirotta et al., 2013a).

6.6 Experimental Evaluation

The goal of this section is to show the benefits of configuring the environment while the policy learning proceeds. The experiments are conducted on two explicative domains: the Student-Teacher domain (unconstrained model space) the Racetrack Simulator (parametric model space). We compare different target choices (greedy and persistent) and different *update strategies*. Specifically, SPMI, that adaptively updates policy and model, is compared with some alternative model learning approaches: SPMI-alt(ernated) in which model and policy updates are forced to be alternated, SPMI-sup that uses a looser bound, obtained from Theorem 6.7 by replacing $\|\cdot\|_{\mathrm{TV},\mu_{\gamma}^{\pi},P}$ with $\|\cdot\|_{\mathrm{TV},\infty}$,[2] SPI+SMI that optimizes

[2] When considering only policy updates, this is equivalent to the bound used in SPI (Pirotta et al., 2013b).

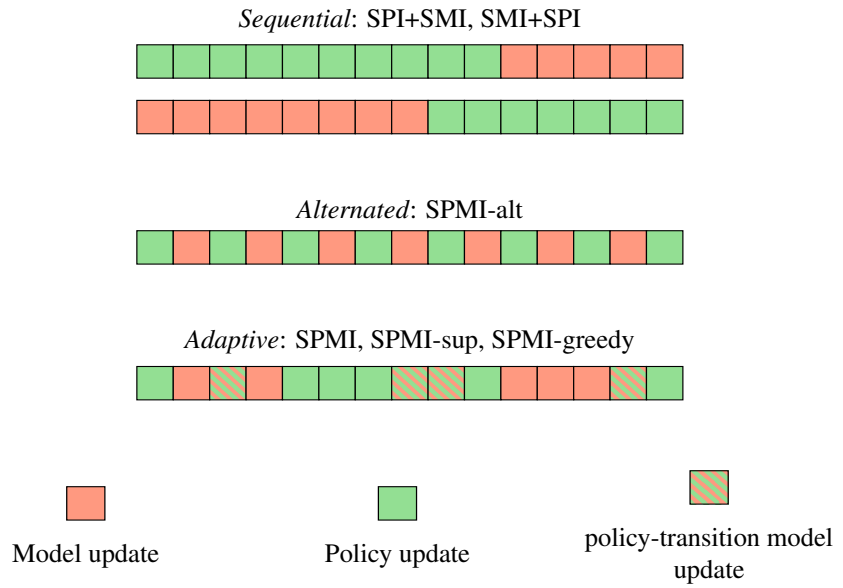


Figure 6.1: *Graphical representation of the update sequence performed by the algorithms compared in the experiments.*

policy and model in sequence and SMI+SPI that does the opposite. A graphical representation of the behavior in terms of policy and model updates of the compared algorithms is reported in Figure 6.1. For the implementation details and additional experiments, refer to Appendix E of the original paper (Metelli et al., 2018a).

6.6.1 Student-Teacher domain

The Student-Teacher domain is a simple model of concept learning, inspired to (Rafferty et al., 2011), involving two entities: the teacher and the student. We assume both entities share the same goal, i.e., maximizing the knowledge the student acquires. The teaching model, however, should be suited for the specific learning policy of the student. For instance, not all students have the same skills and are able to capture the information provided by the teacher with the same speed and effectiveness. Thus, the teaching model should be tailored in order to meet the student's needs. Given the goal of maximizing learning, a teaching model induces an optimal learning policy (within the space of the policies that a certain student can play). Symmetrically, a learning policy determines an optimal teaching model (within the space of models available to the teacher). The question we want to answer in this experiment is: "can we dynamically adapt the teaching model to the learning policy and the learning policy to the teaching model, so to maximize the learning?"

Environment Description We formalize the teaching/learning process as an MDP in which the student is the agent and the teacher is the environment. To fit our framework to this context, we can think of the teacher as an online learning platform that can be

configured by the student in order to improve the learning experience. As in Rafferty et al. (2011) we test the model on the "alphabet arithmetic" a concept-learning task in which literals are mapped to numbers.

We consider n literals $L_1, \ldots, L_n$, to which the student can assign the values $\{0, \ldots, m\}$. The teacher, at each time step, provides an "example", i.e., an equation where a number (from 2 to $p \leqslant n$) of distinct literals sum to a numerical answer (e.g., `A+C=3`). The set of all possible examples is given by:

$$\mathcal{E} = \left\{ \sum_{i \in I} L_i = l \ : I \subseteq \{1, \ldots, n\}, 2 \leqslant |I| \leqslant p, l \in \{0, \ldots, |I|m\} \right\}.$$

The student reacts to an example by performing an action, i.e., an assignment of literals (e.g., `A=1`, `C=3`). The set of all assignments, i.e., actions, is given by:

$$\mathcal{A} = \{L_1 = l_1, L_2 = l_2, \ldots, L_n = l_n \ : l_i \in \{0, ..., m\}, i \in \{1, \ldots, n\}\},$$

thus $|\mathcal{A}| = (m+1)^n$. In order to model the student policy space we assume that a student can modify an arbitrary number of literals under the assumption that two consecutive assignments satisfy $\sum_{i=1}^{n} |l'_i - l_i| \leqslant k$, i.e., the literal values can change by not more than a total value of k. This models the learning limitations of the student, in particular how hard is for the student to capture the teacher information. We assume that the teacher can provide any example. The set of states is the Cartesian product between examples and assignments, i.e., $\mathcal{S} = \mathcal{E} \times \mathcal{A}$. A problem setting is defined by the 4-tuple *number of literals - maximum literal value - maximum update allowed - maximum number of literals in the statement* (e.g., 2-1-1-2).

The goal of the student is to perform assignments that are consistent with the teacher's examples (within its limitations on the possible assignments). So, while the student is learning the optimal policy it can configure the teacher to provide more suitable examples. The reward is 1 when the assignment is consistent, 0, when it is not. Notice that we do not have a goal state, differently from (Rafferty et al., 2011). We assume that, in the beginning, both policy and model are uniform distribution on the allowed actions/states. Figure 6.2 reports a portion of the MDP corresponding to the 2-1-1-2 problem.

Experiments We start considering the illustrative example in which there are two binary literals, and the student can change only one literal at a time (2-1-1-2). This example aims to illustrate the benefits of SPMI over other update strategies and target choices.

In Figure 6.3, we show the behavior of the different update strategies starting from a uniform initialization. We can see that both SPMI and SPMI-sup perform the policy updates and the model updates in sequence. This is a consequence of the fact that, by looking only at the local advantage function, it is more convenient for the student to learn an almost optimal policy with no intervention on the teacher and then refining the teacher model to gain further reward. The joint and adaptive strategy of SPMI outperforms both SPMI-sup and SPMI-alt. The alternated policy-transition model update (SPMI-alt) is not convenient since, with an initial poor-performing policy, updating the model does not yield a significant performance improvement. It is worth noting that all the methods converge in a finite number of steps and the learning rates α and β exhibit an exponential growth trend. The bound value is not plotted for SPMI-alt since the algorithm keeps alternating between

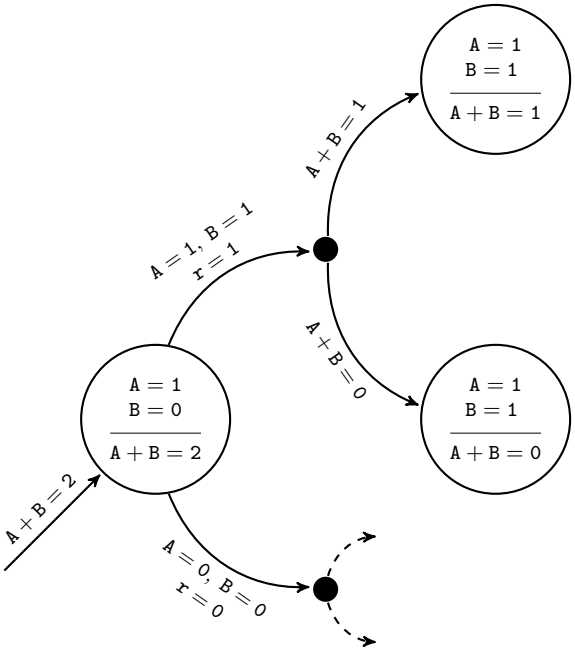


Figure 6.2: *Portion of the MDP corresponding to the problem 2-1-1-2.*

policy and model updates that are performed using different bounds. Thus, its behavior is not really meaningful.

In Figure 6.4, we compare the greedy target selection with the persistent target selection. The former, while being the best local choice maximizing the advantage, might result in an unstable behavior that slows down the convergence of the algorithm. This is confirmed since the number of times the target policy changes is significantly larger compared to the persistent choice.

In Figure 6.5 (left), we compare SPMI, where both the policy and the transition model are learned simultaneously, with the sequential approaches SPI+SMI and SMI+SPI. We immediately notice that learning both policy and model is convenient since the performance of SPMI at convergence is higher than that of SPI (only policy learned) and SMI (only model learned), corresponding to the markers in Figure 6.5. Furthermore, we observe that SPMI outperforms SPI+SMI but displays a slower convergence compared to SMI+SPI. This behavior can be explained based on the peculiar properties of the problem, in combination with the local nature of our bound. Indeed, at the beginning, it is convenient to learn the policy (the slope of the dotted line is larger w.r.t. that of the dash-dotted line in the first iterations). However, it turns out that by sacrificing some performance improvement at the beginning it is possible to reach faster convergence. Nevertheless, if we are interested in the online performance of the learning process, we clearly see that SPMI reveals to be the best strategy. Figure 6.5 (right) proposes another interesting case in which SPMI-sup, SPMI-alt, and SMI+SPI all converge faster than SPMI. From these examples, we can conclude that although SPMI adopts the tightest bound, its update strategy is not guaranteed to yield globally the fastest convergence as it is based on local information,

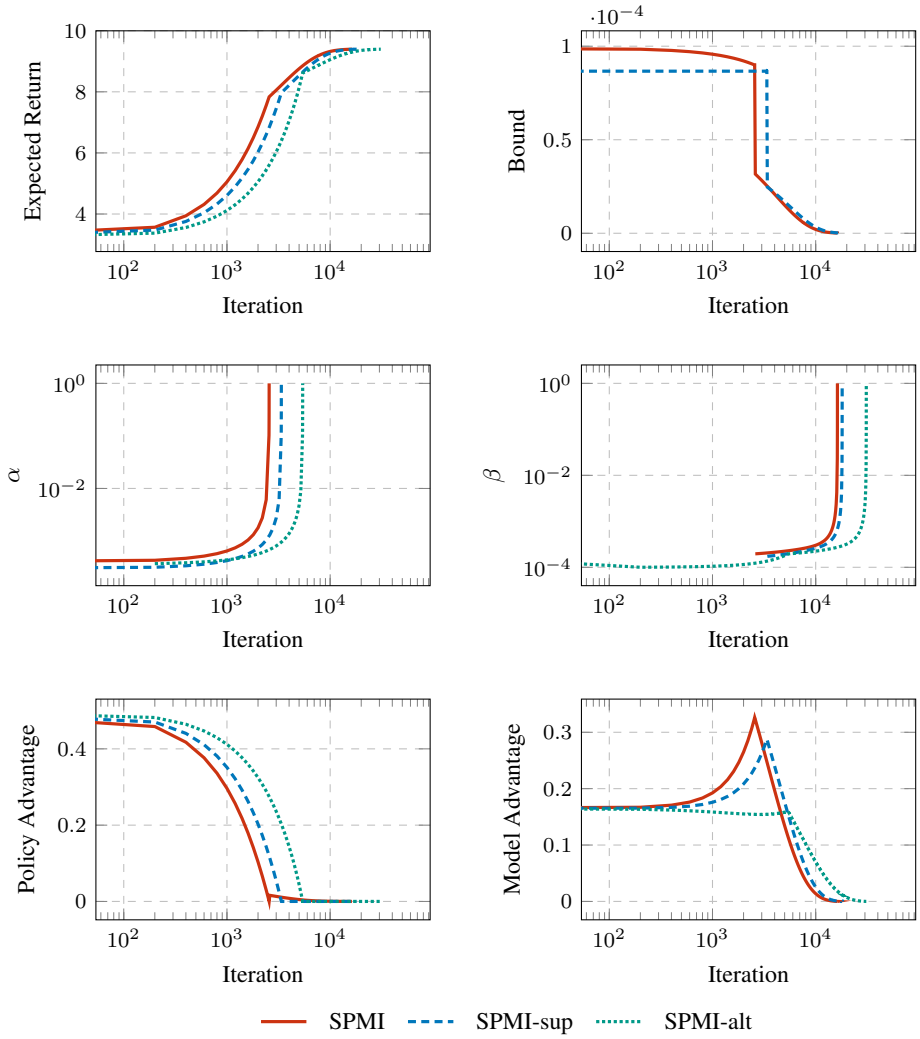


Figure 6.3: *Expected return, bound value, α and β coefficients, policy and model advantages for the Student-Teacher domain 2-1-1-2 for different update strategies.*

i.e., expected relative advantage.

In Table 6.1 we report the number of iterations to convergence for the different problem settings we considered. We can see that SPMI is the first or the second algorithm to converge in most of the cases.

6.6.2 Racetrack Simulator

The Racetrack simulator is an abstract representation of a car driving problem. The autonomous driver (agent) has to optimize a driving policy to run the vehicle on the track,

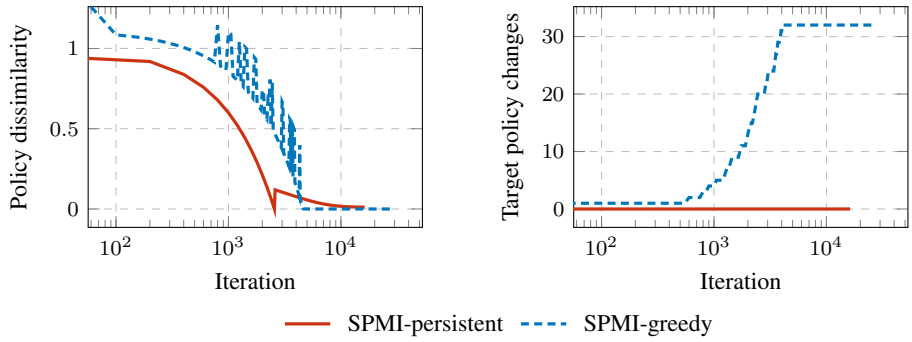


Figure 6.4: *Policy dissimilarity and number of target policy changes for greedy and persistent target choices in the 2-1-1-2 case.*

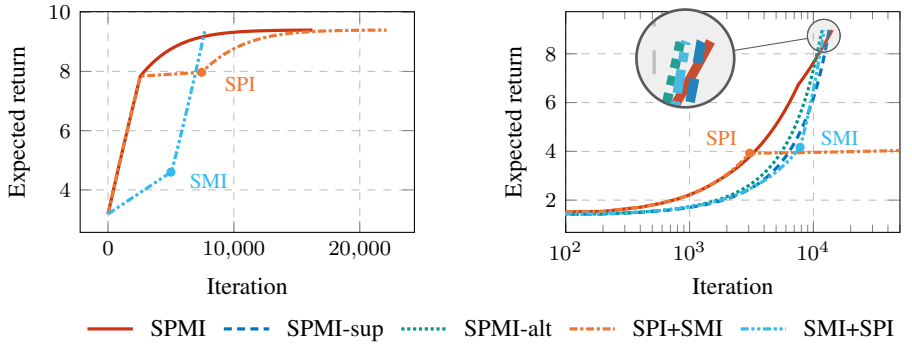


Figure 6.5: *Expected return for the Student-Teacher domains 2-1-1-2 (left) and 2-3-1-2 (right) for different update strategies.*

reaching the finish line as fast as possible. During the process, the agent can configure two vehicle settings to improve their driving performance: the *vehicle stability* and the *engine boost*.

Environment Description The autonomous driver, the learning agent, has to optimize a driving policy in order to run the vehicle to the track finish line as fast as possible. The vehicle and the track naturally compose the model of the learning process, however, there is the possibility to tune a set of vehicle parameters, such as aerodynamic profile (to affect the vehicle stability) and engine setting. Therefore, to maximize the performance, the driving policy of the agent and the model configuration has to be jointly considered. It is noteworthy that a specific model parametrization (vehicle setting) induces an optimal driving policy and, on the other hand, a driving policy determines an optimal model parametrization. Moreover, a policy-transition model pair that results to be optimal for a specific track may not be optimal for a (morphologically) different track. Then, the question we aim to answer with this experiment is the following: "can we learn the optimal policy-transition model pair for a given track by dynamically adapt the vehicle parametrization to the driv-

Problem	SPMI	SPMI-sup	SPMI-alt	SPI+SMI	SPI+SMI
2-1-1-2	<u>16234</u>	18054	30923	22130	**7705**
2-1-2-2	**2839**	3194	5678	**2839**	12973
2-2-1-2	20345	<u>18287</u>	>50000	39722	**10904**
2-2-2-2	**12025**	<u>14315</u>	>50000	>50000	15257
2-3-1-2	14187	13391	**11772**	>50000	<u>12183</u>
3-1-1-2	<u>15410</u>	17929	22707	31122	**14257**
3-1-2-2	**3313**	3313	8434	**3313**	22846
3-1-3-2	**2945**	3435	5891	**2945**	18090

Table 6.1: *Number of steps for convergence for the update strategies in different problem settings of the Student-Teacher domain. In **bold** the best algorithm and <u>underlined</u> the second best. The runs were stopped after 50000 iterations.*

ing policy and, conversely, the driving policy to the vehicle parametrization during the learning process?"

We formalize the learning process as an MDP in which the driver is the agent and the environment is composed by the track and the vehicle. The track is represented by a grid of positions, each grid point is either of type $roadway$, $wall$, $initial\ position$, $goal\ position$. A state in the learning process belongs to the set:

$$\mathcal{S} = \Big\{(x, y, v_x, v_y)\ : x \in \{0, \dots, x_{\max}\}, y \in \{0, \dots, y_{\max}\}, \\ v_x \in \{v_{\min}, \dots, v_{\max}\}, v_y \in \{v_{\min}, \dots, v_{\max}\}\Big\},$$

where (x, y) corresponds to a grid position and (v_x, v_y) are the speed along the coordinate axes. At each step, the agent can increment or decrement the speed along a coordinate direction or do nothing. Then, the action space is represented by the following:

$$\mathcal{A} = \Big\{\text{keep, increment } v_x\text{, increment } v_y\text{, decrement } v_x\text{, decrement } v_y\Big\}.$$

The learning process starts at the state corresponding to the initial position with zero velocities; the agent collects reward 1 when it reaches a state corresponding to the goal position he collects 0 reward in any other case.

The transition model induces a success probability to any action, a failed action causes a random action to occur instead of the one selected by the agent. This probability aims to model the stability of the vehicle, the more the vehicle is unstable, the more is hard for the agent to drive it (or select an action). The model also induces a failure probability to every action: a failure represents a break of the vehicle, thus it directly cause the end of the episode. This feature represents the pressure on the vehicle engine, the more performance the driver asks for, the more it may break down. We formalize the transition model as a convex combination between a set of vertex models: these correspond to vehicle configuration pushed towards the limit in terms of the aspects described above. For our purpose, we define a model dichotomy related to vehicle stability: $P\_highspeed$ (P_hs) trades stability at lower speed to have more stability (or high action success probability) in

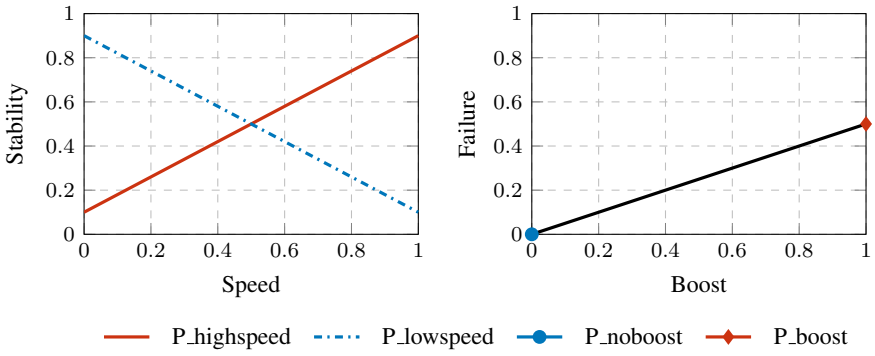


Figure 6.6: *Graphical representation of the racetrack extreme models.*

Figure 6.7: *Graphical representation of the tracks used in the Racetrack Simulator. From left to right: T1, T3, T4 and T2 just below. Each position has a type label: red for initial states, green for goal states, gray for walls, and white for roadtracks.*

high speed situations, $P\_lowspeed$ (P_ls), instead, provides more stability in low speed situation and poor stability at higher speed. We define also a model dichotomy related to engine boost: P_boost (P_b) guarantees higher engine performance and a lower reliability (or higher failure probability), at the opposite $P\_noboost$ (P_nb) provides higher reliability but poor engine performance. In Figure 6.6 we propose a graphical representation of the features of these extreme models.

Considering any possible combination of stability and engine setting, we define the model set (set of vertex models) $\mathcal{P}_{\text{vtx}} = \{P\_hs\_b, P\_hs\_nb, P\_ls\_b, P\_ls\_nb\}$. Each model in this set is obtained by taking, for each state-action pair, the product of the transition probabilities of the components (e.g., $P\_hs\_b(\cdot|s,a) = P\_hs(\cdot|s,a) \cdot P\_b(\cdot|s,a)$). Then, we derive the model space as the convex hull of the vertices in the model set:

$$P_{\boldsymbol{\omega}} = \omega_1 \cdot P\_hs\_b + \omega_2 \cdot P\_hs\_nb + \omega_3 \cdot P\_ls\_b + \omega_4 \cdot P\_ls\_nb,$$

where $\sum_{i=1}^{4} \omega_i = 1$ and $\omega_i \geqslant 0$ for all $i \in \{1,2,3,4\}$. While the agent is learning the

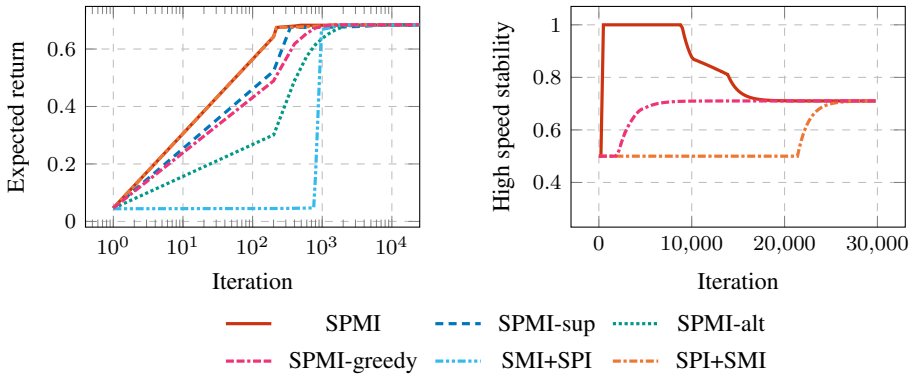


Figure 6.8: *Expected return and coefficient of the high speed stability vertex model for different update strategies in track T1.*

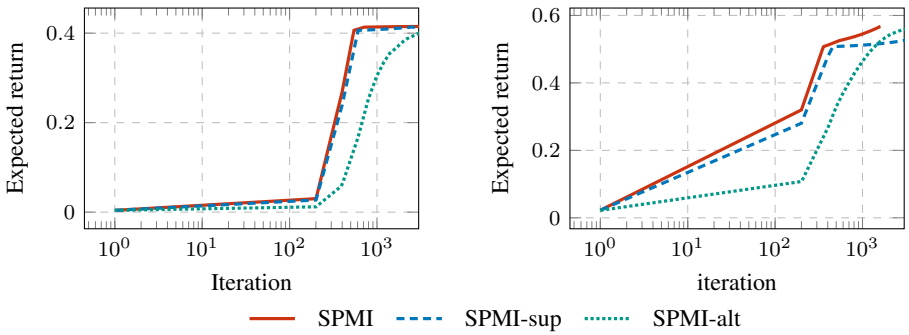


Figure 6.9: *Expected return of the Racetrack Simulator in the T3 and T4 for different update strategies and considering vehicle stability configuration only.*

optimal driving policy, the model parametrization can be configured (selecting a vector $\boldsymbol{\omega}$) trying to fit the vehicle settings to the driving policy and simultaneously trying to fit the policy-settings pair to the morphology of the track. At the beginning of the learning process, we assume the policy to be a uniform distribution on the action space and the model to be $(0, 0.5, 0, 0.5)$, that we can consider the most conservative parametrization in our context. We also report in Figure 6.7 an illustrative representation of the tracks used in the experiments.

Two Vertex Models Experiment We first present an introductory example on a simple track (T1) in which only the vehicle stability can be configured. In Figure 6.8 left, we highlight the effectiveness of SPMI updates over SPMI-sup and SPMI-alt and sequential executions of SMI and SPI on track T1. Furthermore, the SPMI-greedy, which selects the target greedily in each iteration, results in lower performance w.r.t. SPMI. Comparing SPMI with the sequential approaches, we can easily deduce that is not valuable to configure the vehicle stability, i.e., updating the model, while the driving policy is still really

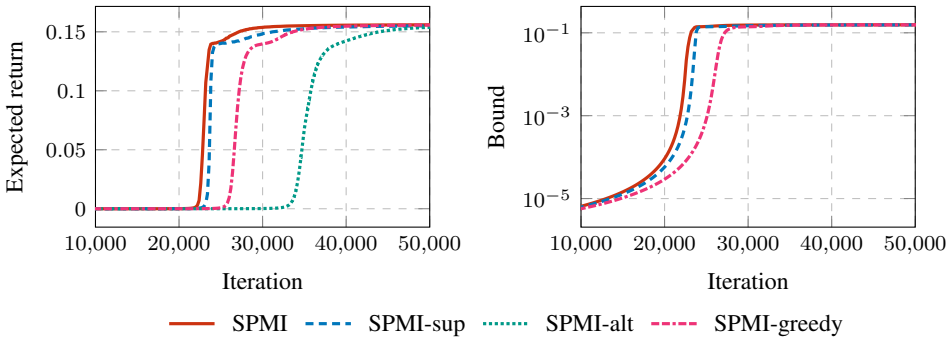


Figure 6.10: *Expected return in track T2 with 4 vertex models for different update strategies.*

rough. Although in the showed example the difference between SPMI and SPI+SMI is way less significant in terms of expected return, their learning paths are quite peculiar. In Figure 6.8 right, we show the trend of the model coefficient related to high-speed stability. While the optimal configuration results in a mixed model for vehicle stability, SPMI exploits the maximal high-speed stability to learn the driving policy efficiently in an early stage, SPI+SMI, instead, executes all the policy updates and then directly leads the model to the optimal configuration. SPMI-greedy prefers avoiding the maximal high-speed stability region as well. It is worthwhile to underline that SPMI could temporarily drive the process aside from the optimum if it leads to higher performance from a local perspective. We consider this behavior quite valuable, especially in scenarios where performance degradations during learning are unacceptable.

In Figure 6.9, we propose additional experiments on different tracks (T3 and T4). We can notice that SPMI displays a better learning curve compared to the other strategies. Moreover, we observe that, while the online performance is comparable with the previous example, the convergence speed is significantly faster. This can be explained by the fact that, in these tracks, the optimal environment configuration corresponds to a vertex model (and not a mixed configuration). For this reason, when such a vertex model is selected as target, it is kept fixed for the whole learning process.

Four Vertex Models Experiment Figure 6.10 shows how the previous considerations generalize to an example on a morphologically different track (T4), in which also the engine boost can be configured. The learning process is characterized by a long exploration phase, both in the model and the policy space, in which the driver cannot lead the vehicle to the finish line to collect any reward. Then, we observe a fast growth in expected return when the agent has acquired enough information to reach the finish line consistently. SPMI displays a more efficient exploration phase compared to other update strategies and target choices, leading the process to a quicker convergence to the optimal model. In Figure 6.11, we show the behavior of the convex combination coefficients associated with the four vertex models. We can clearly see that the learning process prefers high speed stability and an intermediate engine boost configuration. Nevertheless, it interesting to observe that during the learning process the importance low speed stability is increased.

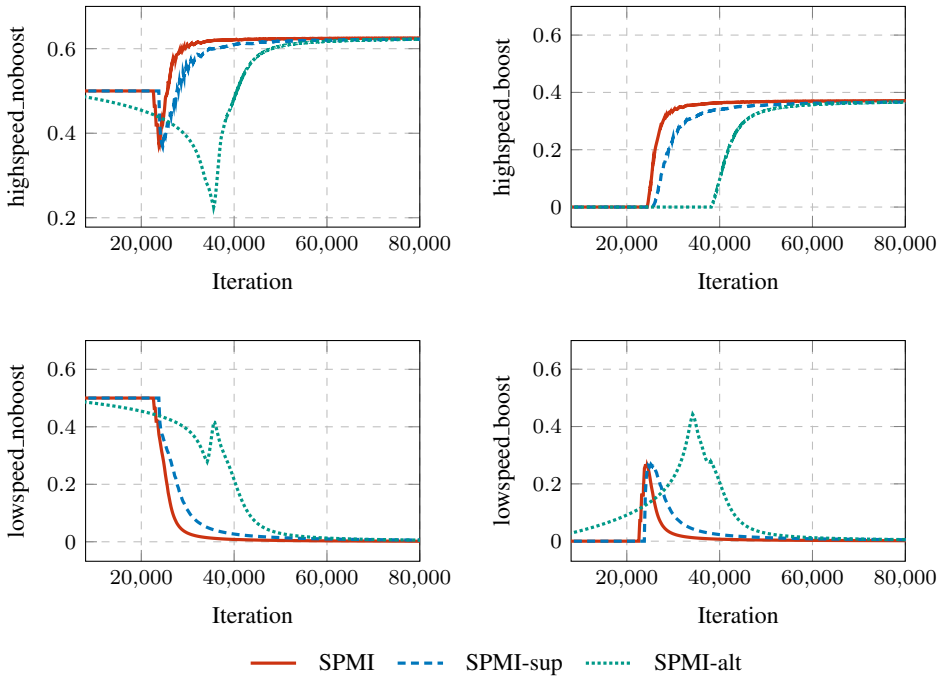


Figure 6.11: *Coefficients of the different vertex models for different update strategies in track T2 with 4 vertex models.*

Finally, we observe that all the algorithms, although with different paths, reach the same final configuration.

6.6.3 Summary of the Experiments

We provided an experimental evaluation in simple discrete domains, inspired by the examples motivating the introduction of the Conf-MDP framework. The evaluation allowed to highlight essentially two points. First, we have seen that learning the configuration, together with the agent's policy, allows reaching higher performances overall. This behavior is particularly visible in Student-Teacher domain (Section 6.6.2) and emerges in the Racetrack simulator, in the particular choice of the coefficients (Section 6.6.1). This empirically motivates the introduction of the Conf-MDP framework, regardless of the employed learning algorithm. Second, we illustrated that *jointly and adaptively* learning the policy and the environment configuration is, most of the times, the preferable option, compared to sequential approaches in terms of learning curve. This is a property of SPMI and it can be observed in both the domains we tested.

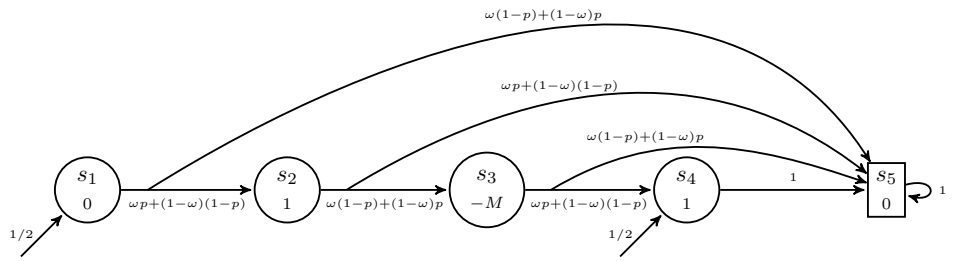


Figure 6.12: *An example of Conf-MDP with local maxima. The transition probabilities are reported on the arrows and the reward function inside the circles.*

6.7 Examples of Conf-MDPs

In this section, we report two examples of Conf-MDPs that display interesting behaviors when running SPMI. First, we show an example in which SPMI can be trapped into a local optima solution (Section 6.7.1). This is a phenomenon that SPMI shares with SPI. Second, we provide an example in which the optimal configuration parameters do not lie in the border of the domain, even when considering a parametrization made of a convex combination of vertex models (Section 6.7.2).

6.7.1 An example of Conf-MDP with local optima

Let us consider the Conf-MDP represented in Figure 6.12 where $\omega \in [0, 1]$ is the parameter, $p \in [0, 1]$ is a small fixed probability and $M > 0$ is a large positive number. In each state, there is only one action available (i.e., all policies are optimal).[3] The vertex models are obtained for $\omega \in \{0, 1\}$. For both target models, there is a small probability to get the punishment $-M$ since for $\omega = 0$ the probability to reach state s_3 from s_2 is p and for $\omega = 1$ state s_2 is reachable from s_1 with probability p. We expect that by mixing the two target models we can only worsen the performance. It is simple to realize that the expected return is a cubic function of ω. We report the expression for $p = 0.1$ and $\gamma = 1$:

$$J(\omega) = \frac{1}{2}\left(0.512\omega^3 + (0.64M - 1.088)\omega^2 - (0.64M + 0.296)\omega + 1.981 - 0.09M\right).$$

We can find the stationary points by looking at the derivative:

$$\frac{\partial J^{P_\omega}}{\partial \omega} = 0.768\omega^2 + (0.64M - 1.088)\omega - 0.32M - 0.148.$$

For M sufficiently large the derivative has one sign variation thus it has two solutions of opposite sign, having expression:

$$\omega_{1,2} = \frac{1}{24}\left(17 - 10M \pm 10\sqrt{M^2 - M - 4}\right).$$

[3] This is a simplification to focus the attention to the optimization of the transition model. These examples can be generalized for the more realistic case of multiple actions.

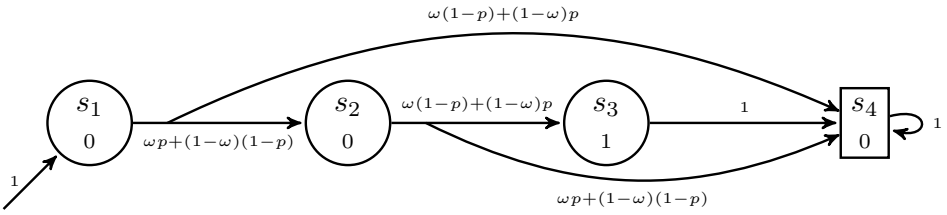


Figure 6.13: *An example of Conf-MDP with mixed optimal model. The transition probabilities are reported on the arrows and the reward function inside the circles.*

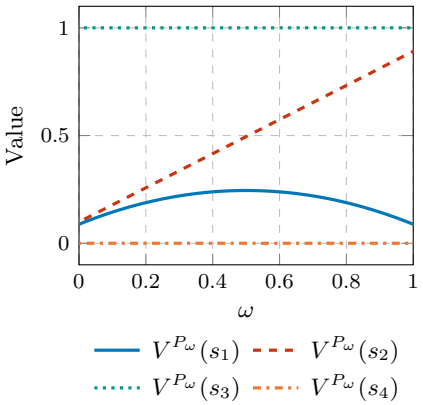


Figure 6.14: *The state value function of the Conf-MDP in Figure 6.13 as a function of the parameter.*

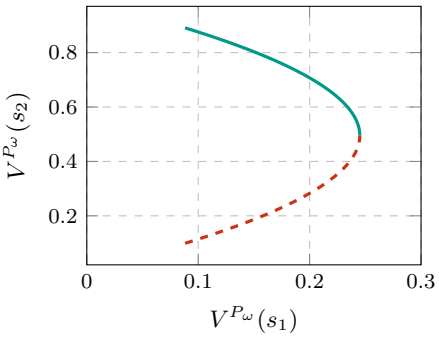


Figure 6.15: *The state value function of states A and B (the only ones varying with the parameter) of the Conf-MDP in Figure 6.13. The green continuous line is the Pareto frontier.*

Clearly, we are interested only in the solutions within $[0, 1]$ thus we discard the negative one. It is simple to see that the positive solution is approximately $\frac{1}{2}$ for M sufficiently large, as:

$$\lim_{M \to +\infty} \frac{1}{24} \left(17 - 10M + 10\sqrt{M^2 - M - 4}\right) = \frac{1}{2}.$$

However, having a look at the second derivative we realize that this is a point of minimum, since

$$\frac{\partial^2 J(\omega)}{\partial \omega^2} = 1.536\omega + 0.64M - 1.088\big|_{\omega=\frac{1}{2}} > 0.$$

Notice that in the unfortunate case in which SPMI is initialized at this value of ω the expected relative advantage (which is the same as the gradient) is zero for both the vertex models and therefore there would be no update. Therefore, the maximum must lie on the border, specifically either for $\omega = 0$ or $\omega = 1$. It is simple to see that $J(1) > J(0)$. Moreover, if we compute the value of the gradient for $\omega = 0$ and $\omega = 1$ we realize that in both cases the value is negative. Having a negative advantage, SPMI would never make any step even when the model is initialized at the lower performance vertex $\omega = 0$.

6.7.2 An example of Conf-MDP with a mixed optimal model

We consider the Conf-MDP as represented in Figure 6.13. As in the previous case, the parameter is $\omega \in [0,1]$ and $p \in [0,1]$ is a fixed probability. We want to show that there exists no value of ω such that P_ω maximizes the value function in all states, while there exists one value of ω maximizing the expected return. It is simple to compute the value function in each state:

$$
\begin{aligned}
V^{P_\omega}(s_1) &= \gamma^2 \left(\omega p + (1-\omega)(1-p)\right)\left(\omega(1-p) + (1-\omega)p\right), \\
V^{P_\omega}(s_2) &= \gamma \left(\omega(1-p) + (1-\omega)p\right), \\
V^{P_\omega}(s_3) &= 1, \\
V^{P_\omega}(s_4) &= 0.
\end{aligned}
$$

Since the initial state is s_1 we have that $J(\omega) = V^{P_\omega}(s_1)$ which is maximized for $\omega = \frac{1}{2}$. However, there is no value of ω for which the value function of each state is maximized. As shown in Figure 6.14, while $V^{P_\omega}(s_1)$ is maximal in $\omega = \frac{1}{2}$, $V^{P_\omega}(s_2)$ is maximal for $\omega = 1$. All values of $\omega \in \left[\frac{1}{2}, 1\right]$ are indeed Pareto optimal (Figure 6.15). With some calculations we can determine the expression of the expected relative advantage functions:

$$
\begin{aligned}
\mathbb{A}_{P_\omega}^{P_1} &= \gamma^2(1-\omega)(1-2\omega)(1-2p)^2 \\
\mathbb{A}_{P_\omega}^{P_2} &= -\gamma^2\omega(1-2\omega)(1-2p)^2.
\end{aligned}
$$

We clearly see that they both vanish for $\omega = \frac{1}{2}$.

CHAPTER 7

Learning in Continuous Configurable Markov Decision Processes

7.1 Introduction

In Chapter 6, we introduced a safe-learning algorithm, *Safe Policy Model Iteration* (SPMI), to solve the learning problem in the Conf-MDP framework, based on the optimization of a lower bound of the performance improvement to ensure a monotonic increase of the long-term reward (Kakade and Langford, 2002; Pirotta et al., 2013b). Although this approach succeeded in showing the benefits of configuring the environment in some illustrative examples, it is quite far from being applicable to real-world scenarios. SPMI is affected by two main limitations. First of all, it is only applicable to problems with a finite state-action space, while the most interesting Conf-MDP examples have, at least, a continuous state space (e.g., the car configuration problem). Second, it requires full knowledge of the environment dynamics. This latter limitation is the most relevant as, in reality, we almost never know the true environment dynamics, and even if a model is available it could be too approximate or too complex and computationally expensive (e.g., the fluid-dynamic model of a car).

In this chapter, we propose a new learning algorithm for the Conf-MDP problem that overcomes the main limitations of SPMI. *Relative Entropy Model Policy Search* (REMPS) belongs to the *trust-region* class of methods (Schulman et al., 2015) and takes inspiration from REPS (Peters et al., 2010). REMPS operates with parametric policies $\pi_{\boldsymbol{\theta}}$ and configurations $P_{\boldsymbol{\omega}}$ and can be endowed with an approximate configuration model $\widehat{P}_{\boldsymbol{\omega}}$ that can

be estimated from interaction with the environment. At each iteration, REMPS performs two phases: *optimization* and *projection*. In the optimization phase, we aim at identifying a new stationary distribution for the Conf-MDP that maximizes the long-term reward in a neighborhood of the current stationary distribution. This notion of neighborhood is encoded in our approach as a KL–divergence constraint. However, this distribution may fall outside the space of representable distributions, given the parametrization of the policy and that of the configuration. Thus, the second phase performs a moment projection in order to find an approximation of this stationary distribution in terms of representable policies and configurations.

In principle, the learning process in a parametric Conf-MDP can be carried out by a standard stochastic gradient method (Sutton et al., 1999a; Peters and Schaal, 2008). We can easily adapt the classic REINFORCE (Williams, 1992) and G(PO)MDP (Baxter and Bartlett, 2001) estimators for learning the configuration parameters. However, we believe that a first-order method does not scale to relevant situations that are of motivating interest in the Conf-MDP framework. For instance, it may be convenient to select a new configuration that makes the performance of the current policy worse because, in this new configuration, we have a much better chance of learning high-performing policies. We argue that this behavior is impossible by using a gradient method, as the gradient update direction attempts to improve performance for all parameters, including those in the transition model. This example justifies the choice of our trust-region method that allows a closed-form optimization in a controlled region. It has been proved empirically that these methods, also in the policy search framework, are able to overcome local maxima (Levine and Koltun, 2013).

Chapter Outline The chapter is organized as follows. We start in Section 7.2, by recalling the optimality conditions for solving a parametric Conf-MDP and presenting the straightforward extensions of REINFORCE and G(PO)MDP for model learning. Section 7.3 introduces our algorithm, REMPS, and its two phases: optimization and projection. The theoretical analysis of REMPS is provided in Section 7.4, including a finite-sample analysis of the single step of REMPS. Section 7.5 shows how to equip REMPS with an approximation of the environment dynamics. Finally, Section 7.6 presents the experimental evaluation on both discrete and continuous tasks. To simplify the mathematical treatment, we will assume that all relevant distributions admit a probability density function w.r.t. the Lebesgue measure.

7.2 Solving Parametric Conf-MDPs

In Chapter 5, we introduced the optimality conditions for Conf-MDPs assuming that the search of the transition model and the policy is extended to the whole space of Markovian stationary models $\mathcal{P}^{\mathrm{SR}}$ and policies Π^{SR}. As we already mentioned, this general setting, although being extremely convenient from a theoretical standpoint, it might result quite unrealistic, especially from the configuration standpoint. We recall that in a *parametric* Conf-MDP we restrict the search of he transition model and the policy to appropriate spaces:

$$\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \mathscr{P}(\mathcal{A}) : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\},$$

$$\mathcal{P}_\Omega = \{P_{\boldsymbol{\omega}} : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S}) : \boldsymbol{\omega} \in \Omega \subseteq \mathbb{R}^q\}.$$

Given a pair of parameters $(\boldsymbol{\theta}, \boldsymbol{\omega}) \in \Theta \times \Omega$, the corresponding policy $\pi_{\boldsymbol{\theta}}$ and transition model $P_{\boldsymbol{\omega}}$, induce for every $\gamma \in [0, 1]$ a γ-discounted stationary distribution (Definition 2.3) $\mu_\gamma^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}$. We denote with $\mathcal{D}_{\Theta,\Omega} = \{\mu_\gamma^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}} : (\boldsymbol{\theta}, \boldsymbol{\omega}) \in \Theta \times \Omega\}$ the set of γ-discounted stationary distributions induced by the parameter spaces Θ and Ω. In a parametric setting, the goal consists in finding the best policy parameter $\boldsymbol{\theta}^*$ and best environment configuration parameter $\boldsymbol{\omega}^*$ so that they maximize the expected return:

$$(\boldsymbol{\theta}^*, \boldsymbol{\omega}^*) \in \underset{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega}{\arg\max} \{J(\boldsymbol{\theta}, \boldsymbol{\omega})\}, \tag{7.1}$$

where $J(\boldsymbol{\theta}, \boldsymbol{\omega})$ is an abbreviation of $J^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}} = J^{\mu_\gamma^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}}$ that makes more explicit the dependence on the parameters.

7.2.1 Gradient Estimators for Parametric Configuration Learning

When Π_Θ and $\mathcal{P}_\Omega$ are parametric spaces made of stochastic and differentiable policies and transition models respectively, we can address the optimization problem in Equation (7.1) via gradient ascent. In this section, we provide the straightforward extensions of REINFORCE (Williams, 1992) and G(PO)MDP (Baxter and Bartlett, 2001) gradient estimators that can be used to adapt policy gradient methods to the problem of learning parametric environment configurations. We have already provided in Chapter 6, the P-Gradient Theorem, introduced in Metelli et al. (2018a), which is the natural adaptation of the Policy Gradient Theorem of Sutton et al. (1999a). We can also directly derive the trajectory-based expression of the gradient w.r.t. the environment configuration parameters.

Proposition 7.1. *Let $\mathcal{P}_\Omega$ be a class of parametric stochastic transition models differentiable in $\boldsymbol{\omega} \in \Omega$, let $\pi \in \Pi^{\mathrm{SR}}$ be a policy (non necessarily parametric). Then, the gradient of the expected return w.r.t. $\boldsymbol{\omega}$ is given by:*

$$\begin{aligned}\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}) &= \mathbb{E}^{\pi, P_{\boldsymbol{\omega}}}\left[\nabla_{\boldsymbol{\omega}} \log \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) G_\gamma(\tau)\right] \\ &= \mathbb{E}^{\pi, P_{\boldsymbol{\omega}}}\left[\sum_{t=0}^{\infty} \nabla_{\boldsymbol{\omega}} \log p_{\boldsymbol{\omega}}(S_{t+1}|S_t, A_t) G_\gamma(\tau)\right],\end{aligned}$$

where $\mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}$ is the trajectory density function and $G_\gamma(\tau) = \sum_{t=0}^{\infty} \gamma^t R_{t+1}$ is the trajectory return.

Proof. The result derives from the linearity of the gradient and expectation and using the log-trick:

$$\begin{aligned}\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}) &= \nabla_{\boldsymbol{\omega}} \int_{\mathcal{T}} \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) G_\gamma(\tau) \mathrm{d}\tau \\ &= \int_{\mathcal{T}} \nabla_{\boldsymbol{\omega}} \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) G_\gamma(\tau) \mathrm{d}\tau \\ &= \int_{\mathcal{T}} \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) \nabla_{\boldsymbol{\omega}} \log \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) G_\gamma(\tau) \mathrm{d}\tau.\end{aligned}$$

By rewriting the log density and exploiting the properties of the logarithm and observing that the terms depending on $\boldsymbol{\omega}$ are those of the transition model only, we obtain:

$$\nabla_{\boldsymbol{\omega}} \log \mathbb{p}^{\pi, P_{\boldsymbol{\omega}}}(\tau) = \nabla_{\boldsymbol{\omega}} \log \left(\mu_0(S_0) \prod_{t=0}^{\infty} \pi(A_t|S_t) p_{\boldsymbol{\omega}}(S_{t+1}|S_t, A_t) r(R_{t+1}|S_t, A_t, S_{t+1})\right)$$

$$= \sum_{t=1}^{\infty} \nabla_{\boldsymbol{\omega}} p_{\boldsymbol{\omega}}(S_{t+1}|S_t, A_t).$$

□

We can now derive the REINFORCE and G(PO)MDP estimators for the gradient and the corresponding optimal baselines.

REINFORCE The REINFORCE estimator is simply obtained by writing the sample-based version of the second expression in Proposition 7.1. Let $\{\tau_i\}_{i=1}^n$ be a set of trajectories, the estimator can be expressed for every $k \in \{1, \dots, q\}$ as:

$$\widehat{\nabla}^{\mathrm{RF}}_{\boldsymbol{\omega}_k} J(\boldsymbol{\omega}) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{T(\tau_i)-1} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{\tau_i,t+1}|S_{\tau_i,t}, A_{\tau_i,t}) \right) \times \left(\sum_{t=0}^{T(\tau_i)-1} \gamma^t R_{\tau_i,t+1} - \mathbf{b}_k \right),$$

where $\mathbf{b} \in \mathbb{R}^q$ is the *baseline*. The estimator $\widehat{\nabla}^{\mathrm{RF}}_{\boldsymbol{\omega}_k} J(\boldsymbol{\omega})$ is unbiased for every choice of $\mathbf{b}$, but its variance is minimized for the following baseline, defined for every $k \in \{1, \dots, q\}$ as:

$$\mathbf{b}_k^{\mathrm{RF}*} = \frac{\mathbb{E}^{\pi, P_{\boldsymbol{\omega}}} \left[\left(\sum_{t=0}^{\infty} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{t+1}|S_t, A_t) \right)^2 G_\gamma(\tau) \right]}{\mathbb{E}^{\pi, P_{\boldsymbol{\omega}}} \left[\left(\sum_{t=0}^{\infty} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{t+1}|S_t, A_t) \right)^2 \right]}$$

The derivation of the baseline is here omitted since it is analogous to that employed for deriving the baseline in traditional REINFORCE.

G(PO)MDP The derivation of the G(PO)MDP estimator can be performed analogously to that for the policy parameters, by observing that the reward is independent of the future states and actions given the current and past ones. Indeed, we can simplify the second expression of the gradient derived in Proposition 7.1 as follows:

$$\nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}) = \mathbb{E}^{\pi, P_{\boldsymbol{\omega}}} \left[\sum_{t=0}^{\infty} \sum_{l=0}^{t} \nabla_{\boldsymbol{\omega}} \log p_{\boldsymbol{\omega}}(S_{l+1}|S_l, A_l) \gamma^t R_{t+1} \right].$$

The estimator is obtained by simply replacing the expectation with the sample mean, obtained with a set of trajectories $\{\tau_i\}_{i=1}^n$ and defined for every $k \in \{1, \dots, q\}$ as:

$$\widehat{\nabla}^{\mathrm{G(PO)MDP}}_{\boldsymbol{\omega}_k} J(\boldsymbol{\omega}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T(\tau_i)-1} \left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{\tau_i,l+1}|S_{\tau_i,l}, A_{\tau_i,l}) \right) \times \left(\gamma^t R_{\tau_i,t+1} - \mathbf{b}_{t,k} \right),$$

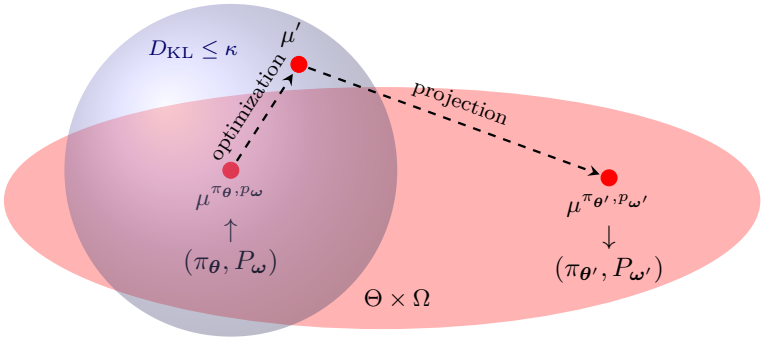


Figure 7.1: *Graphical representation of the two phases of REMPS, optimization and projection.*

where for every $t \in \{0, \dots, T-1\}$, we have that $\mathbf{b}_t \in \mathbb{R}^q$ is a *step-dependent baseline*. The expression of the baseline minimizing the variance is provided below for every $t \in \{0, \dots, T-1\}$ and $k \in \{1, \dots, q\}$:

$$\mathbf{b}_{t,k}^{\mathrm{G(PO)MDP*}} = \frac{\mathbb{E}^{\pi, P_{\boldsymbol{\omega}}}\left[\left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{l+1}|S_l, A_l)\right)^2 \gamma^t R_{t+1}\right]}{\mathbb{E}^{\pi, P_{\boldsymbol{\omega}}}\left[\left(\sum_{l=0}^{t} \nabla_{\boldsymbol{\omega}_k} \log p_{\boldsymbol{\omega}}(S_{l+1}|S_l, A_l)\right)^2\right]}.$$

7.3 Relative Entropy Model Policy Search

In this section, we introduce an algorithm to solve the learning problem in the Conf-MDP framework that can be effectively applied to continuous state-action spaces and overcomes the *local* nature of the previously presented gradient methods. *Relative Entropy Model Policy Search* (REMPS), imports several ideas from the classic REPS (Peters et al., 2010); in particular, the use of a constraint to ensure that the resulting new stationary distribution is sufficiently close to the current one. REMPS consists of two subsequent phases: *optimization* and *projection*. In the optimization phase (Section 7.3.1) we look for the stationary distribution μ' (discounted or not) that optimizes the expected return as in Equation (7.1). This search is limited to the space of distributions that are not too dissimilar from the current stationary distribution $\mu_\gamma^{\pi,P}$. The notion of dissimilarity is formalized in terms of a threshold $\kappa > 0$ on the KL-divergence. However, the resulting distribution μ' may not fall within the space of the representable stationary distributions given our parametrization $\mathcal{D}_{\Theta,\Omega}$. Therefore, similarly to Daniel et al. (2012), in the projection phase (Section 7.3.2) we need to retrieve a policy $\pi_{\boldsymbol{\theta}}$ and a configuration $P_{\boldsymbol{\omega}}$ inducing a stationary distribution $\mu_\gamma^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}} \in \mathcal{D}_{\Theta,\Omega}$ as close as possible to μ'. Refer to Figure 7.1 for a graphical representation of these two phases.

7.3.1 Optimization

The optimization problem can be stated in terms of stationary distributions only. Given a stationary distribution $\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ (e.g., the one used to collect samples $\mu_\gamma^{\pi,P}$) and a KL–divergence threshold $\kappa > 0$, we look for a new stationary distribution $\mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ that is the solution of the following optimization problem PRIMAL$_\kappa$:[1]

$$\max_{\mu' \in \mathscr{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})} J^{\mu'} = \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \mu'(s,a,s') r(s,a,s') \mathrm{d}s\mathrm{d}a\mathrm{d}s'$$
$$\text{s.t.} \quad D_{\mathrm{KL}}(\mu' \| \mu) = \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \mu'(s,a,s') \log \frac{\mu'(s,a,s')}{\mu(s,a,s')} \mathrm{d}s\mathrm{d}a\mathrm{d}s' \leqslant \kappa.$$

It is worth noting that, unlike REPS, we do not enforce a constraint on the validity of the stationary distribution w.r.t. the transition model (see Section 3.3.2), as in a Conf-MDP we have the possibility to change the transition model, determining an effect on the stationary distribution. With similar mathematical tools, we can solve PRIMAL$_\kappa$ in closed form.

Theorem 7.2. *Let $\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be a probability measure and $\kappa > 0$ a KL-divergence threshold. The solution $\mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ of the problem PRIMAL$_\kappa$, for $\kappa > 0$, satisfies for every $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$:*

$$\mu'(s,a,s') \propto \mu(s,a,s') \exp\left(\frac{1}{\eta} r(s,a,s')\right), \tag{7.2}$$

where η is the unique solution of the dual problem DUAL$_\kappa$:

$$\min_{\eta \in [0,+\infty)} g(\eta) = \eta \log \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \mu(s,a,s') \exp\left(\frac{1}{\eta} r(s,a,s') + \kappa\right) \mathrm{d}s\mathrm{d}a\mathrm{d}s'.$$

Proof. For the sake of brevity, we define $\mathcal{X} = \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $(s,a,s') = x \in \mathcal{X}$. We restate the PRIMAL$_\kappa$ problem in a more explicit form:

$$\max_{\mu'} \int_{\mathcal{X}} \mu'(x) r(x) \mathrm{d}x \tag{P.1}$$
$$\text{s.t.} \int_{\mathcal{X}} \mu'(x) \log \frac{\mu'(x)}{\mu(x)} \mathrm{d}x \leqslant \kappa \tag{P.2}$$
$$\int_{\mathcal{X}} \mu'(x) \mathrm{d}x = 1, \tag{P.3}$$

where we simply made explicit the constraint guaranteeing that μ' must sum up to one. Note that we do not need to ensure that $\mu'(x) \geqslant 0$ for all $x \in \mathcal{X}$ since this is guaranteed by the KL-divergence constraint. We solve the optimization problem using the Lagrange multipliers. We denote with $\eta \geqslant 0$ the Lagrange multiplier associated with the KL constraint (P.2) and with λ the multiplier associated with the constraint (P.3). The Lagrangian function becomes:

$$\mathcal{L}(\mu', \eta, \lambda) = \int_{\mathcal{X}} \mu'(x) r(x) \mathrm{d}x + \eta \left(\kappa - \int_{\mathcal{X}} \mu'(x) \log \frac{\mu'(x)}{\mu(x)} \mathrm{d}x\right) \tag{P.4}$$
$$+ \lambda \left(1 - \int_{\mathcal{X}} \mu'(x) \mathrm{d}x\right) \tag{P.5}$$

[1] The KL-divergence allows solving the optimization problem in a particularly convenient way. In principle, other divergences could be employed, like total variation or Rényi divergence (Rényi, 1961).

$$= \int_{\mathcal{X}} \mu'(x) \left(r(x) - \eta \log \frac{\mu'(x)}{\mu(x)} - \lambda \right) \mathrm{d}x + \eta\kappa + \lambda.$$

Taking the functional derivative of $\mathcal{L}$ w.r.t. μ' and applying a simple form of the Euler-Lagrange equation (Gelfand and Silverman, 2000), we get:

$$\frac{\delta \mathcal{L}}{\delta \mu'(x)} = r(x) - \eta \log \frac{\mu'(x)}{\mu(x)} - \eta - \lambda = 0$$
$$\implies \mu'(x) = \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \exp\left(-1 - \frac{\lambda}{\eta}\right). \tag{P.6}$$

We can derive an expression for μ' by enforcing the constraint (P.3):

$$\exp\left(-1 - \frac{\lambda}{\eta}\right)^{-1} = \int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \mathrm{d}x$$
$$\implies \mu'(x) = \frac{\mu(x) \exp\left(\frac{r(x)}{\eta}\right)}{\int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \mathrm{d}x}. \tag{P.7}$$

Substituting (P.6) into the Lagrangian function (P.4) and recalling (P.7), we obtain the dual function:

$$\begin{aligned}
g(\eta, \lambda) &= \exp\left(-1 - \frac{\lambda}{\eta}\right) \int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \Big\{ r(x) \\
&\quad - \eta \log \left[\exp\left(\frac{r(x)}{\eta}\right) \exp\left(-1 - \frac{\lambda}{\eta}\right) \right] - \lambda \Big\} \mathrm{d}x + \eta\kappa + \lambda \\
&= \eta \exp\left(-1 - \frac{\lambda}{\eta}\right) \int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \mathrm{d}x + \eta\kappa + \lambda \\
&= \eta + \eta\kappa + \lambda \\
&= \eta \log \left[\exp\left(-1 - \frac{\lambda}{\eta}\right)^{-1} \right] + \eta\kappa \\
&= \eta \log \int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta}\right) \mathrm{d}x + \eta\kappa \\
&= \eta \log \int_{\mathcal{X}} \mu(x) \exp\left(\frac{r(x)}{\eta} + \kappa\right) \mathrm{d}x.
\end{aligned}$$

Making the change of variable $\overline{\eta} = 1/\eta$, we have that $\frac{1}{\overline{\eta}} \log \int_{\mathcal{X}} \mu(x) \exp\left(\overline{\eta} r(x)\right) \mathrm{d}x$ is convex (Boyd et al., 2004). Moreover, $\frac{\kappa}{\overline{\eta}}$ is strictly convex (as $\frac{\partial^2}{\partial \overline{\eta}^2} \frac{\kappa}{\overline{\eta}} = \frac{2\kappa}{\overline{\eta}^3} > 0$ for $\kappa > 0$), therefore their sum is strictly convex. Furthermore, function g is proper as it admits at least one feasible point (e.g., $\eta = 1$). Thus, being g strictly convex and proper, the optimization problem admits a unique solution (Boyd et al., 2004). □

Thus, to find the optimal solution of PRIMAL$_\kappa$ we must first determine η, by solving DUAL$_\kappa$. It can be proved, as done in REPS, that with a change of variable $\overline{\eta} = \frac{1}{\eta}$, we have that $g(\overline{\eta})$ is a convex function (Boyd et al., 2004), and therefore DUAL$_\kappa$ can be easily solved using standard optimization tools. Given a value of η, the new stationary distribution μ' is defined by the exponential reweighting of each $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ triple with its reward $r(s, a, s')$. Moreover, given a stationary distribution μ', we can derive a representation of a policy π' and a configuration P' inducing μ'.

Corollary 7.3. *Let $\mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be a probability measure (e.g., the solution of PRIMAL$_\kappa$). Then μ' is induced by the transition model $P' \in \mathcal{P}^{\mathrm{SR}}$ and the policy $\pi' \in \Pi^{\mathrm{SR}}$ defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:*

$$p'(s'|s,a) \propto p(s'|s,a) \exp\left(\frac{1}{\eta} r(s,a,s')\right),$$
$$\pi'(a|s) \propto \pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{1}{\eta} r(s,a,s')\right) \mathrm{d}s'.$$

Proof. Recall the factorization of $\mu'(s,a,s')$ as $\mu'(s,a,s') = \mu'(s)\pi'(a|s)p'(s'|s,a)$ for every $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Therefore, we have:

$$p'(s'|s,a) = \frac{\mu'(s,a,s')}{\mu'(s)\pi'(a|s)} = \frac{\mu'(s,a,s')}{\mu'(s,a)} = \frac{\mu'(s,a,s')}{\int_{\mathcal{S}} \mu'(s,a,s')\mathrm{d}s'}.$$

Now, we substitute the expression of μ':

$$\begin{aligned} p'(s'|s,a) &= \frac{\mu(s,a,s') \exp\left(\frac{r(s,a,s')}{\eta}\right)}{\int_{\mathcal{S}} \mu(s,a,s') \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'} \\ &= \frac{\mu(s)\pi(a|s)p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right)}{\mu(s)\pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'} \\ &= \frac{p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right)}{\int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'}. \end{aligned}$$

In a similar way for the policy, recall that $\mu'(s,a) = \mu'(s)\pi'(a|s)$, we have:

$$\pi'(a|s) = \frac{\mu'(s,a)}{\mu'(s)} = \frac{\int_{\mathcal{S}} \mu'(s,a,s')\mathrm{d}s'}{\int_{\mathcal{A}} \int_{\mathcal{S}} \mu'(s,a,s')\mathrm{d}s'\mathrm{d}a}.$$

Now, we substitute the expression of μ' again:

$$\begin{aligned} \pi'(a|s) &= \frac{\int_{\mathcal{S}} \mu(s,a,s') \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'}{\int_{\mathcal{A}} \int_{\mathcal{S}} \mu(s,a,s') \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'\mathrm{d}a} \\ &= \frac{\mu(s)\pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'}{\mu(s) \int_{\mathcal{A}} \pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'\mathrm{d}a} \\ &= \frac{\pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'}{\int_{\mathcal{A}} \pi(a|s) \int_{\mathcal{S}} p(s'|s,a) \exp\left(\frac{r(s,a,s')}{\eta}\right) \mathrm{d}s'\mathrm{d}a}. \end{aligned}$$

□

Sample-based Optimization In practice, we do not have access to the functional form of the sampling distribution $\mu^{\pi,P}$, so we cannot compute the exact solution of the dual problem DUAL$_\kappa$. As in REPS, all expectations must be estimated from samples. Given

a dataset $\mathcal{D} = \{(S_i, A_i, S'_i, R_i)\}_{i=1}^n$ of n samples drawn from $\mu^{\pi,P}$, the empirical dual problem $\widetilde{\text{DUAL}}_\kappa$ becomes:

$$\min_{\widetilde{\eta}\in[0,+\infty)} \widetilde{g}(\widetilde{\eta}) = \widetilde{\eta}\log\frac{1}{n}\sum_{i=1}^{n}\exp\left(\frac{1}{\widetilde{\eta}}R_i + \kappa\right),$$

which yields the solution $\widetilde{\eta}$ inducing the distribution $\widetilde{\mu}'$ as in Equation (7.2), defined for every $(s, a, s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ as:

$$\widetilde{\mu}(s,a,s') \propto \mu(s,a,s')\exp\left(\frac{1}{\widetilde{\eta}}r(s,a,s')\right).$$

We discuss the effect of using a finite number of samples in Section 7.4.3.

7.3.2 Projection

The solution μ' of the PRIMAL$_\kappa$ problem does not belong, in general, to the class of stationary distributions $\mathcal{D}_{\Theta,\Omega}$ induced by Π_Θ and $\mathcal{P}_\Omega$. For this reason, we look for a parametric policy $\pi_{\boldsymbol{\theta}}$ and a parametric configuration $P_{\boldsymbol{\omega}}$ that induce a stationary distribution $\mu^{\pi_{\boldsymbol{\theta}},P_{\boldsymbol{\omega}}}$ as close as possible to μ', by performing a moment projection (PROJ$_\mu$):[2]

$$\begin{aligned}\boldsymbol{\theta}', \boldsymbol{\omega}' &\in \arg\min_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\omega}\in\Omega}\left\{D_{\text{KL}}\left(\mu'\|\mu^{\pi_{\boldsymbol{\theta}},p_{\boldsymbol{\omega}}}\right)\right\}\\ &= \arg\max_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\omega}\in\Omega}\left\{\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log\mu^{\pi_{\boldsymbol{\theta}},P_{\boldsymbol{\omega}}}(S,A,S')\right]\right\}.\end{aligned}$$

However, this problem is hard to solve as computing the functional form of $\mu^{\pi_{\boldsymbol{\theta}},P_{\boldsymbol{\omega}}}$ is complex and cannot be performed in closed form for most of the cases of interest. If the state space and the action space are finite, we can formulate the problem as follows, recalling the definition of γ-discounted stationary distribution (Definition 2.3):

$$\begin{aligned}\max_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\omega}\in\Omega}\quad & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu(s,a,s')\log\mu'(s')\pi_{\boldsymbol{\theta}}(a|s)p_{\boldsymbol{\omega}}(s'|s,a)\\ \text{s.t.}\quad & \mu'(s) = (1-\gamma)\mu_0(s) + \gamma\sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu'(s')\pi_{\boldsymbol{\theta}}(a|s')p_{\boldsymbol{\omega}}(s|s',a)\quad \forall s\in\mathcal{S}\\ & \mu'(s)\geqslant 0\quad \forall s\in\mathcal{S}.\end{aligned}$$

Nevertheless, in most of the relevant cases, the problem remains intractable as the state space could be very large. Therefore, we consider more convenient projection approaches that we will justify from a theoretical standpoint in Section 7.4.1. A first relaxation consists in finding an approximation of the transition kernel $(P')^{\pi'}$ induced by μ' (PROJ$_{P^\pi}$):

$$\begin{aligned}\boldsymbol{\theta}', \boldsymbol{\omega}' &\in \arg\min_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\omega}\in\Omega}\left\{\mathop{\mathbb{E}}_{S\sim\mu'}\left[D_{\text{KL}}\left((P')^{\pi'}(\cdot|S)\|P^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\omega}}(\cdot|S)\right)\right]\right\}\\ &= \arg\max_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\omega}\in\Omega}\left\{\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log p^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\omega}}(S'|S)\right]\right\}.\end{aligned}$$

[2] When using samples, the moment projection is equivalent to the maximum likelihood estimation.

$$\begin{array}{llllll} \text{REMPS}_\kappa & \mu & \xrightarrow{\text{PRIMAL}_\kappa} & \mu' & \xrightarrow{\text{PROJ}} & (\boldsymbol{\theta}', \boldsymbol{\omega}') \\ \widetilde{\text{REMPS}}_\kappa & \mu & \xrightarrow{\widetilde{\text{PRIMAL}}_\kappa} & \widetilde{\mu}' & \xrightarrow{\widetilde{\text{PROJ}}} & (\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \end{array}$$

Figure 7.2: *Summary of the symbols employed for two phases of REMPS and the corresponding outputs.*

Clearly, we need to be able to compute the functional form of the state transition kernel $P_{\boldsymbol{\omega}}^{\pi_{\boldsymbol{\theta}}}$, which is only possible when considering finite action spaces. Indeed, in such case, we just have to marginalize over the (finite) action space as, for every $s, s' \in \mathcal{S}$:

$$p_{\boldsymbol{\omega}}^{\pi_{\boldsymbol{\theta}}}(s'|s) = \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) p_{\boldsymbol{\omega}}(s'|s, a).$$

When also the action space is infinite, we resort to separate projections for the policy and the transition model ($\text{PROJ}_{\pi,P}$):

$$\begin{aligned} \boldsymbol{\theta}' &\in \arg\min_{\boldsymbol{\theta} \in \Theta} \left\{ \mathop{\mathbb{E}}_{S \sim \mu'} \left[D_{\mathrm{KL}} \left(\pi'(\cdot|S) \| \pi_{\boldsymbol{\theta}}(\cdot|S) \right) \right] \right\} \\ &= \arg\max_{\boldsymbol{\theta} \in \Theta} \left\{ \mathop{\mathbb{E}}_{S,A,S' \sim \mu} \left[\log \pi_{\boldsymbol{\theta}}(A|S) \right] \right\}, \\ \boldsymbol{\omega}' &\in \arg\min_{\boldsymbol{\omega} \in \Omega} \left\{ \mathop{\mathbb{E}}_{S,A \sim \mu'} \left[D_{\mathrm{KL}} \left(P'(\cdot|S, A) \| P_{\boldsymbol{\omega}}(\cdot|S, A) \right) \right] \right\} \\ &= \arg\max_{\boldsymbol{\omega} \in \Omega} \left\{ \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[\log p_{\boldsymbol{\omega}}(S'|S, A) \right] \right\}. \end{aligned}$$

Sample-based Projection Similarly to what happens during the optimization phase, we only have access to a finite dataset of n samples to perform the projection. Moreover, we face an additional challenge, i.e., we need to compute expectations w.r.t. μ', but our samples are collected with μ. This can be cast as an off-distribution estimation problem and therefore we resort to importance weighting (Owen, 2013). In the importance weighting estimation, each sample (S_i, A_i, S_i') is reweighted by the likelihood of being generated by μ', i.e., by:

$$w_i = \frac{\mu'(S_i, A_i, S_i')}{\mu(S_i, A_i, S_i')} \propto \exp\left(\frac{R_i}{\widetilde{\eta}} \right),$$

In the following, we will denote the approximate projections with $\widetilde{\text{PROJ}}$ and with $(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \in \Theta \times \Omega$ the corresponding recovered policy and model parameters. A summary of the objective functions for the different projection approaches, their applicability, and the corresponding estimators are reported in Table 7.1.

Therefore, the full REMPS problem can be stated as the composition of optimization and projection, i.e., $\text{REMPS}_\kappa = \text{PROJ} \circ \text{PRIMAL}_\kappa$, and the corresponding problem from samples as $\widetilde{\text{REMPS}}_\kappa = \widetilde{\text{PROJ}} \circ \widetilde{\text{PRIMAL}}_\kappa$ (Figure 7.2). Refer to Algorithm 7.1 for a high-level pseudocode of REMPS.

Projection	$\lvert\mathcal{S}\rvert = \infty$	$\lvert\mathcal{A}\rvert = \infty$	Exact objective	Estimated objective
PROJ_{μ}	✗	✗	$\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log \mu^{\pi_{\boldsymbol{\theta}},p_{\boldsymbol{\omega}}}(S,A,S')\right]$	$\frac{1}{N}\sum_{i=1}^{N} w_i \log \mu^{\pi_{\boldsymbol{\theta}},p_{\boldsymbol{\omega}}}(S_i,A_i,S_i')$
$\text{PROJ}_{P^{\pi}}$	✓	✗	$\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log p^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\omega}}(S'\lvert S)\right]$	$\frac{1}{N}\sum_{i=1}^{N} w_i \log p^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\omega}}(S_i'\lvert S_i)$
$\text{PROJ}_{\pi,P}$	✓	✓	$\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log \pi_{\boldsymbol{\theta}}(A\lvert S)\right]$ $\mathop{\mathbb{E}}_{S,A,S'\sim\mu'}\left[\log p_{\boldsymbol{\omega}}(S'\lvert S,A)\right]$	$\frac{1}{N}\sum_{i=1}^{N} w_i \log \pi_{\boldsymbol{\theta}}(A_i\lvert S_i)$ $\frac{1}{N}\sum_{i=1}^{N} w_i \log p_{\boldsymbol{\omega}}(S_i'\lvert S_i,A_i)$

Table 7.1: *Applicability, exact objective function and corresponding estimator for the three projections presented.* w_i *is the (non-normalized) importance weight defined as* $w_i = \exp\left(\frac{R_i}{\widetilde{\eta}}\right)$.

Algorithm 7.1: Relative Entropy Model Policy Search (REMPS).

Input: Conf-MDP $\mathcal{C}$, number of iterations T
Output: approximately optimal policy-transition model pair $(\pi_{\boldsymbol{\theta}^{(T)}}, P_{\boldsymbol{\omega}^{(T)}})$
Initialize $\boldsymbol{\theta}^{(0)}, \boldsymbol{\omega}^{(0)}$ arbitrarily
forall $t = 0, 1, \ldots, T-1$ **do**
  Collect n samples $\{(S_i, A_i, S_i', R_i)\}_{i=1}^{n}$ with $\mu^{\pi_{\boldsymbol{\theta}^{(t)}}, P_{\boldsymbol{\omega}^{(t)}}}$
  (*Optimization*) Compute $\widetilde{\eta}$ and $\widetilde{\mu}'$ solving the $\widetilde{\text{DUAL}}_{\kappa}$
  (*Projection*) Perform the projection of $\widetilde{\mu}'$ and obtain $\boldsymbol{\theta}^{(t+1)}$ and $\boldsymbol{\omega}^{(t+1)}$
return $\pi_{\boldsymbol{\theta}^{(T)}}, P_{\boldsymbol{\omega}^{(T)}}$

7.4 Theoretical Analysis

In this section, we elaborate on three theoretical aspects of REMPS. First of all, we provide three inequalities that bound the difference of performance when changing the policy and the model in terms of distributional divergences between stationary distributions, policies, and models (Section 7.4.1). Second, we present a sensitivity study of the hyper-parameter κ (i.e., the KL-divergence threshold) of REMPS (Section 7.4.2). Finally, we discuss a finite-sample analysis of the single step of REMPS (Section 7.4.3). Furthermore, we will consider the following assumption on the regularity of the MDP induced by policies and configurations.

Assumption 7.1. *(Ergodicity) Let* $\pi \in \Pi^{\text{SR}}$ *and* $P \in \mathcal{P}^{\text{SR}}$*, the* ergodicity coefficient *of the Markov chain induced by* π *and* P *is defined as (Seneta, 1988):*

$$\tau\left(P^{\pi}\right) = \sup_{s,s'\in\mathcal{S}} \left\{ \left\| P^{\pi}(\cdot\lvert s) - P^{\pi}(\cdot\lvert s') \right\|_{\text{TV}} \right\}.$$

If $\gamma = 1$*, for every* $(\boldsymbol{\theta}, \boldsymbol{\omega}) \in \Theta \times \Omega$ *we assume* $\tau\left(P^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\omega}}\right) \leqslant \tau_{\max} < 1$.[3]

7.4.1 Performance Bounds

We start with the following result that bounds the absolute difference of expected return with a dissimilarity index between the stationary distributions. The results we provide

[3]Note that $\tau\left(P^{\pi}\right) = 1$ in the case of deterministic transition models.

are stated in terms of the α-Rényi divergence (Rényi, 1961), that we have introduced in Section 3.3.2, and extend those presented in Metelli et al. (2019a) which were formulated in terms of the KL-divergence.

Proposition 7.4. *Let $\mu, \mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be two stationary distributions, then for any $\alpha \in [0, 1]$ it holds that:*

$$\left|J^{\mu'} - J^{\mu}\right| \leqslant 2R_{\max} \left\|\mu' - \mu\right\|_{\text{TV}} \leqslant R_{\max}\sqrt{\frac{2}{\alpha} D_{\alpha}\left(\mu' \| \mu\right)}.$$

Proof. The first inequality is obtained with the following simple derivation:

$$\begin{aligned}\left|J^{\mu} - J^{\mu'}\right| &= \left|\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \left(\mu(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s') - \mu'(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s')\right) r(s, a, s')\right| \\ &\leqslant R_{\max} \int \left|\mu(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s') - \mu'(\mathrm{d}s, \mathrm{d}a, \mathrm{d}s')\right| \\ &= 2R_{\max} \left\|\mu' - \mu\right\|_{\text{TV}}.\end{aligned}$$

The second inequality is a straightforward application of the extension of Pinsker's inequality presented in (van Erven and Harremoës, 2014, Equation (8)). □

This result justifies the projection PROJ_{μ}, since minimizing the KL-divergence between the stationary distributions allows controlling the performance difference. The statement is presented for $\alpha \in [0, 1]$ (the order of the Rényi divergence). Since the Rényi divergence is monotonic in α,we obtain analogous expression also for $\alpha > 1$ (recall that for $\alpha = 1$, we reduce to the KL-divergence). As we have seen in Section 7.3.2, the PROJ_{μ} is typically intractable. Therefore, we now prove that performing the projection of the state transition kernel ($\text{PROJ}_{P^{\pi}}$) still allows controlling the performance difference.

Corollary 7.5. *Let P^{π} and $(P')^{\pi'}$ two transition kernels, inducing the stationary distributions μ and μ' respectively, then, under Assumption 7.1, it holds that for every $\alpha \in [0, 1]$:*

$$\left|J^{\mu'} - J^{\mu}\right| \leqslant R_{\max}\rho\sqrt{\frac{2}{\alpha}\int_{\mathcal{S}} \mu'(\mathrm{d}s) D_{\alpha}\left((P')^{\pi'}(\cdot|s) \| P^{\pi}(\cdot|s)\right)},$$

where $\rho = \frac{\gamma}{1-\gamma}$ if $\gamma < 1$ or $\rho = \frac{1}{1-\tau_{\max}}$ if $\gamma = 1$.

Proof. If $\gamma < 1$, the statement is obtained starting from Proposition 6.2 and bounding $\left\|\mu' - \mu\right\|_{\text{TV}}$ as in Proposition 3.1 of Metelli et al. (2018a):

$$\left\|\mu' - \mu\right\|_{\text{TV}} = \frac{\gamma}{1-\gamma} \int_{\mathcal{S}} \mu'(\mathrm{d}s) \left\|(P')^{\pi'}(\cdot|s) - P^{\pi}(\cdot|s)\right\|_{\text{TV}}.$$

For the case $\gamma = 1$, we start from the following inequality provided in Seneta (1988) (Section 2, taking $p = \infty$) that we rewrite in our notation:

$$\begin{aligned}2\left\|\mu' - \mu\right\|_{\text{TV}} &\leqslant \frac{1}{2}\sum_{k=0}^{\infty} \tau\left(P^{\pi}\right)^k \int_{\mathcal{S}} \left|\int_{\mathcal{S}} \mu'(\mathrm{d}s) \left((P')^{\pi'}(\mathrm{d}s'|s) - p^{\pi}(\mathrm{d}s'|s)\right)\right| \\ &\leqslant \frac{1}{2}\int_{\mathcal{S}} \mu'(\mathrm{d}s) \int_{\mathcal{S}} \left|(P')^{\pi'}(\mathrm{d}s'|s) - P^{\pi}(\mathrm{d}s'|s)\right| \sum_{k=0}^{\infty} \tau_{\max}^k\end{aligned}$$

$$\leqslant \frac{1}{1-\tau_{\max}} \int_{\mathcal{S}} \mu'(\mathrm{d}s) \left\| P^{\pi}(\cdot|s) - (P')^{\pi'}(\cdot|s) \right\|_{\mathrm{TV}},$$

where we exploited Assumption 7.1 for the bound $\tau\left(P^{\pi}\right) \leqslant \tau_{\max} < 1$. An application of the extension of Pinsker's inequality (van Erven and Harremoës, 2014) concludes the proof. □

Finally, the following result provides a justification for the separate projections of policy and model (PROJ$_{\pi,P}$) .

Lemma 7.6. *Let $(\pi, P), (\pi', P') \in \Pi^{\mathrm{SR}} \times \mathcal{P}^{\mathrm{SR}}$ be two policy-transition model pairs, then, under Assumption 7.1, it holds that for every $\alpha \in [0,1]$:*

$$\left| J^{\mu'} - J^{\mu} \right| \leqslant R_{\max}\rho \sqrt{2 \int_{\mathcal{S}\times\mathcal{A}} \mu'(\mathrm{d}s, \mathrm{d}a) \left(D_{\alpha}\left(\pi'(\cdot|s) \| \pi(\cdot|s)\right) + D_{\alpha}\left(P'(\cdot|s,a) \| P(\cdot|s,a)\right)\right)},$$

where $\rho = \frac{\gamma}{1-\gamma}$ if $\gamma < 1$ or $\rho = \frac{1}{1-\tau_{\max}}$ if $\gamma = 1$.

Proof. To prove the result, we refer to the proof of Corollary 6.3 and we employ the inequality:

$$\left\| (P')^{\pi'} - P^{\pi} \right\|_{\mathrm{TV},\mu'} \leqslant \left\| \pi' - \pi \right\|_{\mathrm{TV},\mu'} + \left\| P' - P \right\|_{\mathrm{TV},\mu'}.$$

Then, we bound each of the terms by using the extension of Pinsker's inequality van Erven and Harremoës (2014) to get the Rényi divergence. □

7.4.2 Sensitivity to the KL threshold

We analyze how the performance of the solution of PRIMAL$_{\kappa}$ changes when the KL-divergence threshold κ varies. Suppose that $\kappa' \leqslant \kappa$, then the KL constraint is more restrictive, thus, we expect $J^{\mu'} \leqslant J^{\mu}$. To analyze this setting, let us consider a new class distributions $\mu_{\alpha} = \alpha\mu + (1-\alpha)\mu_0$, with $\alpha \in [0,1]$ and μ_0 be the sampling distribution. Ideally, we could increase α until we saturate the constraint κ', getting a form of projection of μ onto the region that satisfies the constraint induced by κ'. The following result provides a characterization of the value of α in this circumstance.

Lemma 7.7. *Let $\mu, \mu' \in \mathscr{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ be the solutions of the problems PRIMAL$_{\kappa}$ and PRIMAL$_{\kappa'}$ with $\kappa' \leqslant \kappa$ and μ_0 as sampling distribution. Let $\mu_{\alpha} = \alpha\mu + (1-\alpha)\mu_0$ with $\alpha \in [0,1]$. If $D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0) = \kappa'$, then $\alpha \geqslant \frac{\kappa'}{\kappa}$.*

Proof. We use the convexity of the KL divergence: $D_{\mathrm{KL}}(\alpha\eta_1 + (1-\alpha)\eta_2 \| \alpha\nu_1 + (1-\alpha)\nu_2) \leqslant \alpha D_{\mathrm{KL}}(\eta_1\|\nu_1) + (1-\alpha) D_{\mathrm{KL}}(\eta_2\|\nu_2)$ for $\alpha \in [0,1]$. Take $\eta_1 = \mu$, $\eta_2 = \nu_1 = \nu_2 = \mu_0$:

$$\begin{aligned}\kappa' = D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0) = D_{\mathrm{KL}}(\alpha\mu + (1-\alpha)\mu_0 \| \alpha\mu_0 + (1-\alpha)\mu_0) \leqslant \\ \leqslant \alpha D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0) + (1-\alpha) D_{\mathrm{KL}}(\mu_0\|\mu_0) = \alpha D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0).\end{aligned}$$

Therefore, observing that $D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0) \leqslant \kappa$:

$$\alpha \geqslant \frac{\kappa'}{D_{\mathrm{KL}}(\mu_{\alpha}\|\mu_0)} \geqslant \frac{\kappa'}{\kappa}. \tag{P.8}$$

□

The following result upper bounds the reduction in performance between the optimal solution μ of PRIMAL$_{\kappa}$ and the optimal solution μ' of PRIMAL$_{\kappa'}$ when $\kappa' \leqslant \kappa$.

Proposition 7.8. *Let $\mu, \mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be the solutions of PRIMAL$_\kappa$ and PRIMAL$_{\kappa'}$ respectively with $\kappa' \leqslant \kappa$, having μ_0 as sampling distribution. Then, it holds that:*

$$J^{\mu} - J^{\mu'} \leqslant 2R_{\max} \left\| \mu - \mu_0 \right\|_{\text{TV}} \left(1 - \frac{\kappa'}{\kappa} \right). \tag{7.3}$$

Proof. Consider the $\alpha' \in [0, 1]$, as defined in Lemma 7.7, such that $D_{\text{KL}}(\mu_{\alpha'} \| \mu_0) = \kappa'$. We start observing that being μ' the optimal solution with constraint κ' and since $\mu_{\alpha'}$ fulfills the constraint, we surely have $J^{\mu'} \geqslant J^{\mu_{\alpha'}}$. Consider the following sequence of inequalities:

$$\begin{aligned} J^{\mu} - J^{\mu'} &\leqslant J^{\mu} - J^{\mu_{\alpha'}} \\ &\leqslant 2R_{\max} \left\| \mu - \mu_{\alpha'} \right\|_{\text{TV}} \\ &\leqslant 2R_{\max} \left\| (1 - \alpha')(\mu - \mu_0) \right\|_{\text{TV}} \\ &= 2R_{\max}(1 - \alpha') \left\| (\mu - \mu_0) \right\|_{\text{TV}}. \end{aligned}$$

Applying Lemma 7.7 we get $1 - \alpha' \leqslant 1 - \frac{\kappa'}{\kappa}$, from which the result follows. □

This result is general and can be applied broadly to the class of trust-region methods, when using the KL-divergence as a constraint to define the trust-region.

7.4.3 Finite-sample Analysis

We present a finite-sample analysis of the single step of REMPS. In particular, our goal is to upper bound the difference $J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}')$ between the performance of $\mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$, the solution of the exact problem PRIMAL$_\kappa$, and $(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \in \Theta \times \Omega$ obtained after solving the whole $\widetilde{\text{REMPS}}_\kappa$ problem through samples. Thus, starting with $\mu^{\pi,P} \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$, the initial γ-discounted stationary distribution, $\widetilde{\text{REMPS}}_\kappa$ provides the solution $\mu^{\pi_{\widetilde{\theta}'}, P_{\widetilde{\omega}'}}$ which is in terms derived from the $\widetilde{\text{PRIMAL}}_\kappa$ problem yielding $\widetilde{\mu}' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ and the $\widetilde{\text{PROJ}}$ problem. There are two sources of error in this process. First of all, $\widetilde{\mu}'$ is obtained from finite samples and thus it may differ from μ (estimation error). Secondly, we limit to a hypothesis space $\mathcal{D}_{\Theta,\Omega}$ that may not be able to represent $\widetilde{\mu}'$ (approximation error). Furthermore, the projection is performed from samples as well, generating another source of estimation error.

For a given probability measure $\mu \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$, we will denote the following set of the possible solution of the PRIMAL$_\kappa$ problem, ignoring the KL-divergence constraint:

$$\mathcal{D}_{\mu} = \left\{ \mu' \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S}) : \frac{\mu'}{\mu} \propto \exp\left(\frac{r}{\eta} \right) : \eta \in [0, +\infty) \right\}.$$

In order to derive a meaningful bound, we consider the following additional assumptions.

Assumption 7.2. *(Finite pseudo-dimension) Let $\mu^{\pi,P}$ be the sampling distribution, the pseudo-dimensions of the hypothesis spaces $\{ \frac{\mu}{\mu^{\pi,P}} : \mu \in \mathcal{D}_{\mu^{\pi,P}} \}$, $\{ \frac{\mu}{\mu^{\pi,P}} r : \mu \in \mathcal{D}_{\mu^{\pi,P}} \}$, $\{ \frac{\mu}{\mu^{\pi,P}} \log \frac{\mu}{\mu^{\pi,P}} : \mu \in \mathcal{D}_{\mu^{\pi,P}} \}$ and $\{ \frac{\mu}{\mu^{\pi,P}} \log \mu' : \mu \in \mathcal{D}_{\mu^{\pi,P}}, \mu' \in \mathcal{D}_{\Theta,\Omega} \}$ are bounded by $v < +\infty$.*

Assumption 7.3. *(Finite β–moments) There exist $\beta \in (1,2)$, such that*

$$\mathop{\mathbb{E}}_{S,A,S' \sim \mu^{\pi,P}} \left[\left| \frac{\mu(s,a,s')}{\mu^{\pi,P}(s,a,s')} \right|^{\beta} \right]^{1/\beta} \quad \text{and}$$

$$\mathop{\mathbb{E}}_{S,A,S' \sim \mu^{\pi,P}} \left[\left| \frac{\mu(S,A,S')}{\mu^{\pi,P}(S,A,S')} \log \mu'(S,A,S') \right|^{\beta} \right]^{1/\beta}$$

are bounded for all $\mu \in \mathcal{D}_{\mu^{\pi,P}}$ and $\mu' \in \mathcal{D}_{\Theta,\Omega}$.

Assumption 7.2 requires that all the involved hypothesis spaces (for the solution of the PRIMAL$_\kappa$ and PROJ) are characterized by a finite pseudo-dimension. This assumption is necessary to state learning theory guarantees. Assumption 7.3 is more critical as it requires that the involved loss functions (used to solve the PRIMAL$_\kappa$ and PROJ) have a uniformly bounded (over the hypothesis space) moment of order $\beta \in (1,2)$. In particular, the first line states that the exponentiated β-Rényi divergence (Rényi, 1961; Cortes et al., 2010) between μ and $\mu^{\pi,P}$ is finite for some $\beta \in (1,2)$. This requirement allows an analysis based on Cortes et al. (2019) for unbounded loss function with bounded moments. A more straightforward analysis can be made by assuming that the involved loss functions are uniformly bounded and using more traditional tools (Mohri et al., 2012) (see Appendix A.4.4 of Metelli et al. (2019a)). We report below the finite-sample result, under Assumption 7.3, whose derivation is reported in Appendix A.2.

Theorem 7.9. *(Finite–Sample Bound) Let $\mu^{\pi,P} \in \mathscr{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{S})$ be the sampling distribution, $\kappa > 0$ be the KL–divergence threshold, $\mu' \in \mathcal{D}_\mu$ be the solution of the PRIMAL$_\kappa$ problem and $(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \in \Theta \times \Omega$ be the solution of the $\widetilde{\text{REMPS}}_\kappa$ problem with PROJ$_\mu$ computed with $n > 0$ samples collected with μ. Then, under Assumptions 4.1, 7.2 and 7.3, for any $\alpha \in (1,\beta)$, there exist two constants χ, ξ and a function $\zeta(n) = \mathcal{O}(\log n)$ depending on α, and on the samples, such that for any $\delta \in (0,1)$, with probability at least $1 - 4\delta$ it holds that:*

$$J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \leqslant \underbrace{\sqrt{2} R_{\max} \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \inf_{\overline{\mu} \in \mathcal{D}_{\Theta,\Omega}} \left\{ \sqrt{D_{\mathrm{KL}}(\mu \| \overline{\mu})} \right\}}_{\textit{approximation error}}$$
$$+ \underbrace{R_{\max} \chi \sqrt{\epsilon} + R_{\max} \zeta(n) \epsilon + R_{\max} \xi \epsilon^2}_{\textit{estimation error}},$$

where $\epsilon = 2^{\frac{\alpha+2}{2\alpha}} \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}} \Gamma\left(\alpha, \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right)$, which depend on the pseudo-dimension bound $v < +\infty$ and $\Gamma(\alpha,\tau) = \frac{\alpha-1}{\alpha} + \frac{1}{\alpha} \left(\frac{\alpha}{\alpha-1}\right)^{\alpha-1} \left(1 + \left(\frac{\alpha-1}{\alpha}\right)^{\alpha-1} \log \frac{1}{\tau}\right)^{\frac{\alpha-1}{\alpha}}$.

The estimation error is dominated by $\sqrt{\epsilon}$. Ignoring logarithmic terms, we have that $J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') = \widetilde{\mathcal{O}}\big(n^{-\frac{2(\alpha-1)}{4\alpha}}\big)$. In this analysis, we considered the case in which the projection is performed over the stationary distribution (PROJ$_\mu$).[4] The result can be easily extended to the case in which we resort to PROJ$_{P^\pi}$ or PROJ$_{\pi,P}$ (Corollary A.6).

[4]Note that Assumption 7.3 ensures that the approximation error is finite, since the KL-divergence is the 1-Rényi divergence and the Rényi divergence is non-decreasing in the order β (van Erven and Harremoës, 2014).

7.5 Approximation of the Transition Model

The formulation of REMPS we presented above, requires access to a representation of the environment model $P_{\boldsymbol{\omega}}$, depending on a vector of parameters $\boldsymbol{\omega}$. Although the parameters that can be configured are usually known; the environment dynamics is unknown in a model-free scenario. Even when an environment model is available it may be too imprecise or too complex to be used effectively. In principle, we could resort to a general model-based RL approach to effectively approximate the transition model (Deisenroth and Rasmussen, 2011; Nagabandi et al., 2018). However, in our scenario, we need to learn a mapping from state-action-configuration triples to a new state. Our approach is based on a simple maximum likelihood estimation. Given a dataset of experience $\{(S_i, A_i, S'_i, \boldsymbol{\omega}_i)\}_{i=1}^n$ (possibly collected with different policies π_i and different configurations $\boldsymbol{\omega}_i$) and given an approximation space $\widehat{\mathcal{P}}_\Omega \subset \{\widehat{P} : \mathcal{S} \times \mathcal{A} \times \Omega \to \mathscr{P}(\mathcal{S})\}$ we solve the maximum likelihood problem:

$$\max_{\widehat{p} \in \widehat{\mathcal{P}}_\Omega} \frac{1}{n} \sum_{i=1}^{n} \log \widehat{p}(S'_i | S_i, A_i, \boldsymbol{\omega}_i), \tag{7.4}$$

where we made explicit that the distribution of the next state S'_i depends also on the configuration.[5] Given the model approximation, we can run REMPS by replacing P with $\widehat{P} \in \widehat{\mathcal{P}}_\Omega$. We do not impose any restriction on the specific model class $\widehat{\mathcal{P}}_\Omega$ (e.g., neural network, Gaussian process) and on the moment in which the fitting phase has to be performed (e.g., at the beginning of the training or every m iterations).

7.6 Experiments

In this section, we provide the experimental evaluation of REMPS on three domains: a simple chain domain (Section 7.6.1), the classical Cartpole (Section 7.6.2), and a more challenging car-configuration task based on TORCS (Section 7.6.3). In the first two experiments, we compare REMPS with the extension of G(PO)MDP to the policy-configuration learning, whereas in the last experiment we evaluate REMPS against REPS, the latter used for policy learning only. For the implementation details and additional experimental results, refer to Appendix D and E of (Metelli et al., 2019a).

7.6.1 Chain Domain

We start the experimental evaluation with an illustrative example of Conf-MDP, the Chain domain, to show the main features of REMPS compared with other algorithms for learning in Conf-MDPs.

Environment Description In the Chain Domain (Figure 7.3) there are two states s_1 and s_2 and the agent can perform two actions a (forward) and b (backward). The agent is forced to play every action with the same probability in both states, i.e., $\pi_\theta(a|s) = \theta$ and $\pi_\theta(b|s) = 1 - \theta$ for all $s \in \{s_1, s_2\}$ and $\theta \in [0, 1]$. The environment can be configured via the parameter $\omega \in [0, 1]$, that is the probability of action failure. Action a, if successful, takes the agent to state s_2, whereas action b, if successful, takes the agent to state s_1.

[5] Notice that the configuration parameters $\boldsymbol{\omega}$ are an input of the approximate model.

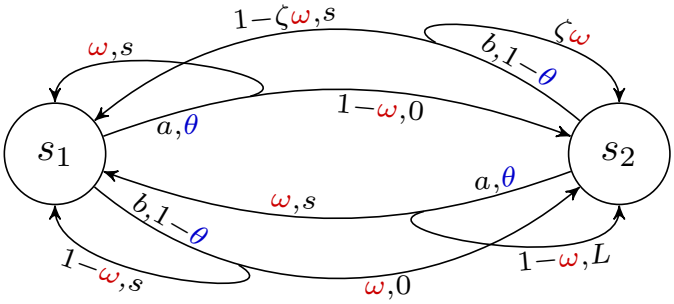


Figure 7.3: *The Chain Domain. On the edges outgoing each state s the pair $(\star, \pi(\star|s))$ where $\star \in \{a, b\}$, while on the arrows incoming to each state s' the pair $(p(s'|s, \star), r(s, \star, s'))$.*

Parameter	Value
ζ	0.2
L	10
l	8
s	2
ω_0	0.8
θ_0	0.2

Table 7.2: *Parameter values used in the experiments on the Chain domain, including the initialization values for θ and ω.*

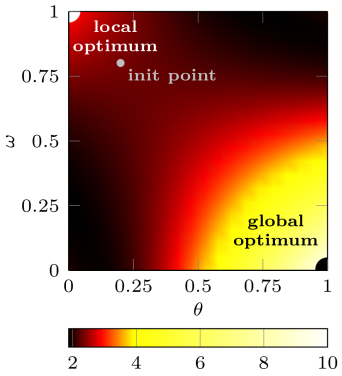


Figure 7.4: *Return surface of the Chain domain.*

When one action fails, the other is executed. The agent gets a high reward, $L > 0$, if, starting from state s_1, it successfully executes action a, while it gets a smaller reward, l $(0 < l < L)$ if it lands in state s_2 starting from 1 but by performing action b. The agent gets an even smaller reward, s $(0 < s < l)$, when it lands in state s_1. The parameter $\zeta \in [0, 1]$ is not configurable and has been added to avoid symmetries in the return surface. The values of the parameters is reported in Table 7.2.

Learning Experiments The main goal of this experiment is to show the benefits of REMPS compared to a simple gradient method, assuming to know the exact environment model. The return surface is characterized by two local maxima (Figure 7.4). If the system is initialized in a suitable region (as in Figure 7.4), to reach the global maximum we need to change the model in order to worsen the current policy performance. In Figure 7.5, we compare our algorithm REMPS using PROJ_{P^π} with different values of κ, against G(PO)MDP adapted to model learning. We can see that G(PO)MDP, besides the slow convergence, moves in the direction of the local maximum. Instead, for some ap-

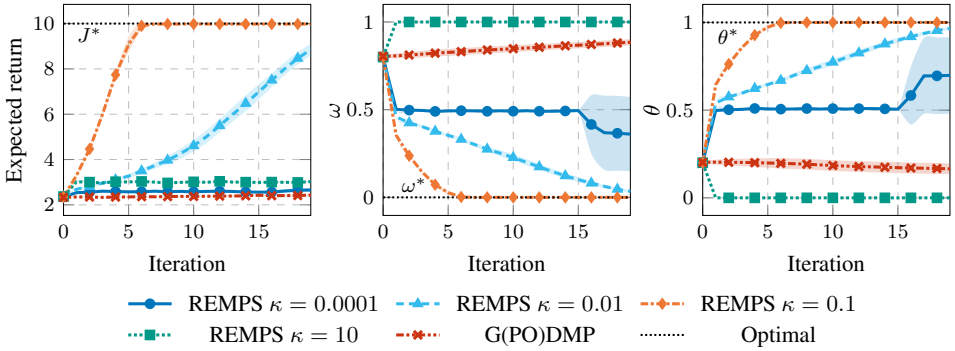


Figure 7.5: *Expected return, configuration parameter ω, and policy parameter θ, as a function of the number of iterations for REMPS with different values of κ and G(PO)MDP. 20 runs, 95% c.i.*

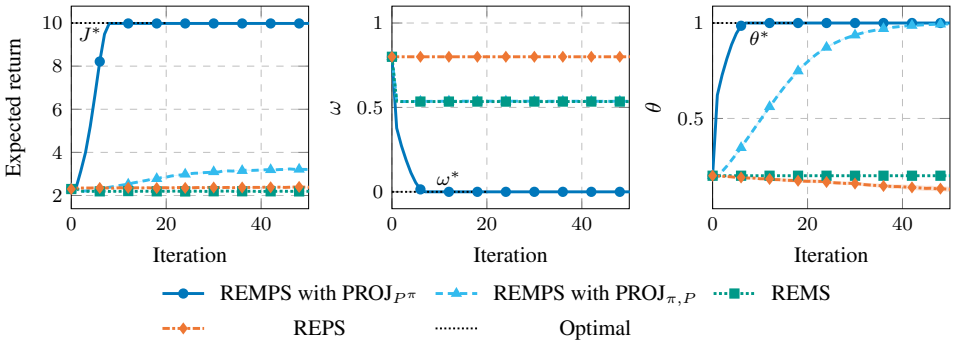


Figure 7.6: *Expected return, configuration parameter (ω) and policy parameter (θ) in the Chain domain with different projection strategies, only-policy (REPS) and only-configuration (REMS) learning as a function of the number of iterations. 20 runs 95% c.i.*

propriate values of the hyperparameter (e.g., $\kappa \in \{0.1, 0.01\}$) REMPS is able to reach the global optimum. It is worth noting that too small a value of κ (e.g., $\kappa = 0.0001$) prevents escaping the basin of attraction of the local maximum. Likewise, for too large κ (e.g., $\kappa = 10$) the estimated quantities are too uncertain and therefore we are not able to reach the global optimum as well.

Comparison of Projection Strategies In Figure 7.6, we compare the different projection strategies together with the no-configuration cases. We can see that the best learning curve is attained by the PROJ$_{P^\pi}$ that reaches the global optimum quickly. REMPS with PROJ$_{\pi,P}$ is unable to reach the global optimum, indeed the configuration parameter gets stuck to a suboptimal value (around 0.55), thus the performance is significantly worse w.r.t. PROJ$_{P^\pi}$. The same behavior, limited to the configuration parameter value, is displayed by the only-configuration (REMS, Relative Entropy Model Search) learning case. Finally, the

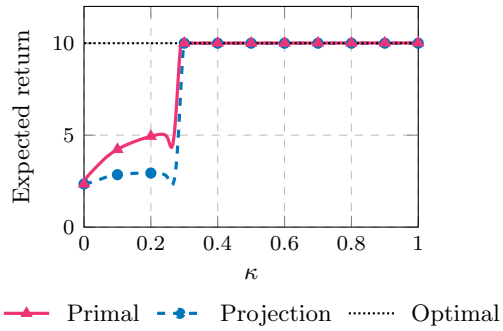


Figure 7.7: *Expected return after PRIMAL$_\kappa$ (primal) and after PROJ$_\mu$ (projection) compared with the optimal performance, as a function of the KL-threshold κ.*

only-policy (REPS, Relative Entropy Policy Search) learning moves the policy parameter towards zero, approaching the local optimum.

Effect of the Policy and Model Spaces The optimization phase (PRIMAL$_\kappa$) in REMPS is able to find in closed-form a new stationary distribution μ' that optimizes our performance index subject to a trust-region constraint. As we have seen, this distribution is not typically representable in space $\mathcal{D}_{\Theta,\Omega}$ and, thus, we need to perform a projection. We analyze how the limited representation power of $\mathcal{D}_{\Theta,\Omega}$ affects performance. Figure 7.7 shows the performance of the best model-policy found as a function of κ and the value of PRIMAL$_\kappa$ which is the expected return obtained by evaluating μ' after solving the primal. We can see that the value of the primal is always larger than the performance after the projection, i.e., the performance of the new policy-configuration pair. As expected, the projection yields a degradation of performance. Notice that for $\kappa \geqslant 0.3$, the primal optimization provides as solution the optimal stationary distribution, i.e., the one we would find without the KL-divergence constraint. This distribution is representable exactly with our policy and model parametrization and, thus, the error is null.

Sensitivity to Parameter Initialization REMPS behaves consistently with respect to a random initialization of model and policy parameters. In Figure 7.8, we can see that REMPS updates the model and policy parameters towards the global maximum while G(PO)MDP updates vary across the different initializations. In the G(PO)MDP learning curves it is possible to see clearly the two attractors.

7.6.2 Cartpole

The Cartpole domain (Widrow and Smith, 1964; Barto et al., 1983) is a continuous-state and finite-action environment. We add to the standard Cartpole domain the possibility to configure the cart force, via the parameter ω.

Environment Description The Cartpole domain (Widrow and Smith, 1964; Barto et al., 1983) is a standard RL benchmark. The environment consists of a cart that moves along

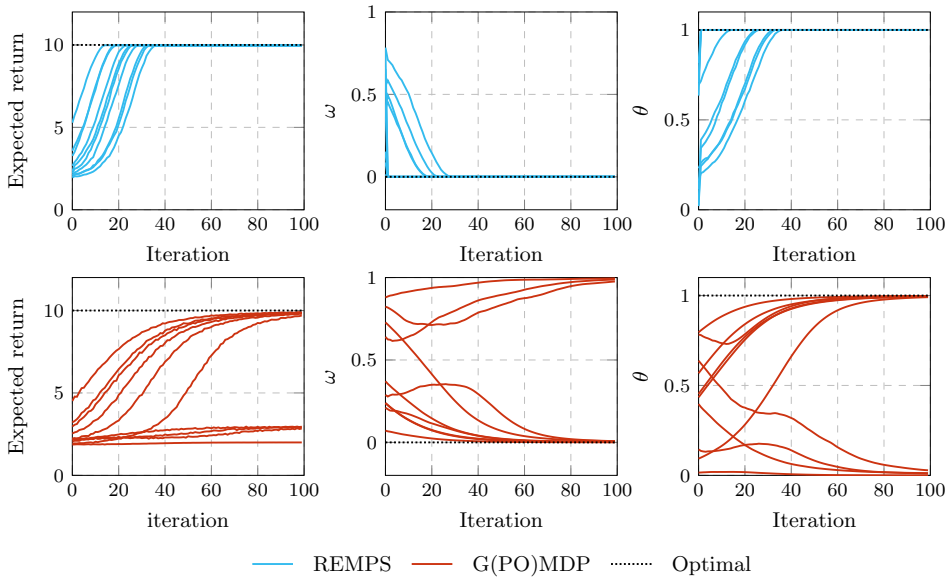


Figure 7.8: *Expected return, configuration parameter (ω), policy parameter (θ) and in the Chain domain with random initialization of model and policy parameter. Comparison between G(PO)MDP and REMPS.*

the horizontal axis and a pole that is anchored on the cart. The state space is continuous and is represented by the position of the cart x, by the cart velocity $\dot{x}$, by the pole angle γ with respect to the vertical, and by the pole angular velocity $\dot{\gamma}$. The action space is discrete and consists of two actions: left L and right R. The model parameter is represented by the force ω to be applied to the cart, which is the same for both actions, thus the resulting force is $\pm\omega$ based on the action. The parameter space is $\Omega = [0, 30]$. Each action, when performed, is affected by a noise term proportional to the applied force and independent for each state component. The goal is to keep the pole in a vertical position ($\gamma = 0$) as long as possible. The episode ends when the pole reaches a certain angle ($|\gamma| > \bar{\gamma}$) or after a predefined number of steps. We want to encourage smaller forces, to this end we use the following reward function:

$$r(s, a, s') = 10 - \frac{\omega^2}{20} - 20 \cdot (1 - \cos(\gamma)).$$

The first part of the reward function is a fixed bonus for each time step the pole is up and the pole angle is within the range $[-\bar{\gamma}, \bar{\gamma}]$. The second part of the reward is a penalty proportional to the force. The third part is a penalty proportional to the pole angle. Ideally, the agent should learn to balance the pole with the smallest force possible, keeping it fixed in a vertical position.

Policy and Model Approximators We evaluate the performance of our algorithm in the exact case (known model) and in the approximate case. In the exact case, we know the

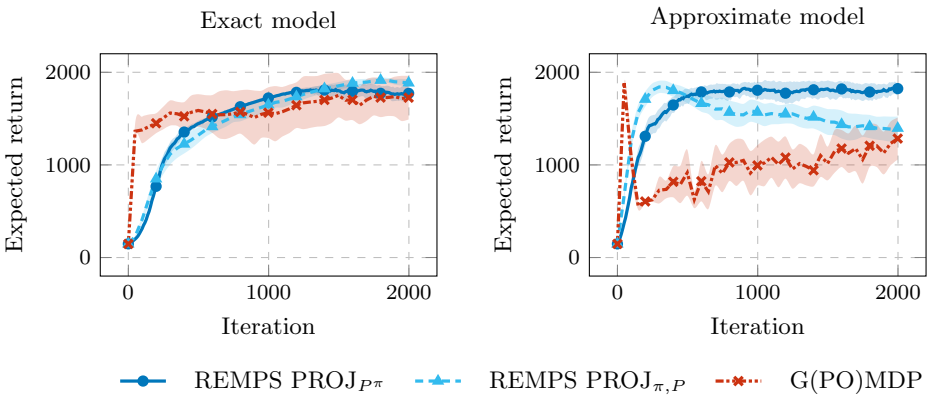


Figure 7.9: *Expected return as a function of the number of iterations for the Cartpole experiment when the environment model is exact (left) or approximated from samples (right) comparing REMPS with PROJ$_{P^\pi}$, PROJ$_{\pi,P}$ and G(PO)MDP. 20 runs, 95% c.i.*

effect of the model parameters on the transition function, i.e., we know $p_\omega(\cdot|s,a)$. The policy $\pi_{\boldsymbol{\theta}}$ is softmax policy with a linear mapping in the state space $\mathbf{s} = (x, \dot{x}, \gamma, \dot{\gamma}, 1)$.

For the approximate case, we assume the distribution over the next states can be approximated by a Gaussian distribution with diagonal covariance. We model the mean and the variance using two independent neural networks with the same input $(\mathbf{s}, a, \omega)$ and the same architecture, i.e., one hidden layer made of 10 neurons with tanh activation. The training is performed just once at the beginning of training, using a dataset made of 10^5 samples collected with different configuration parameters ω (randomly generated).

Experiment The goal of this experiment is to test the ability of REMPS to learn jointly the policy and the environment configuration in a continuous state environment, as well as the effect of replacing the exact environment model with an approximator, trained just at the beginning of the learning process. In Figure 7.9, we compare the performance of REMPS, with the two projection strategies PROJ$_{P^\pi}$ and PROJ$_{\pi,P}$, and G(PO)MDP, starting from a fixed value of the model parameter ($\omega_0 = 8$), both for the case of exact model and approximate model. In the exact case, the performance of REMPS is similar to that of G(PO)MDP. The latter is even faster to achieve a good performance, although it shows a larger variance across the runs. No significant difference can be found between PROJ$_{P^\pi}$ and PROJ$_{\pi,P}$ in this case. Instead, in the approximated scenario, REMPS notably outperforms G(PO)MDP, which shows a very unstable curve. Indeed, constraining the search in a trust-region, as REMPS does by means of κ, is even more important in the approximate case, since the estimated quantities are affected by further uncertainty (injected by the approximated model of the environment). It is worth noting that, in this case, the difference between PROJ$_{P^\pi}$ and PROJ$_{\pi,P}$ is more visible. Indeed, PROJ$_{\pi,P}$ is less precise than PROJ$_{P^\pi}$ (being a relaxation) and thus, when projecting μ', it trusts the approximate model moving towards a suboptimal configuration.

Parameter	Description
α	Angle between the car direction and the direction of the track axis.
rpm	Number of rotation per minute of the car engine.
v_x	Speed of the car along the longitudinal axis of the car.
v_y	Speed of the car along the transverse axis of the car.
v_z	Speed of the car along the Z axis of the car.
track	Vector of 19 range finder sensors: each sensor returns the distance between the track edge and the car within a range of 200 meters.
trackPos	Distance between the car and the track axis.
wheelSpinVel	Vector of 4 sensors representing the rotation speed of wheels.

Table 7.3: *State space of the TORCS experiment.*

7.6.3 Driving and Configuring with TORCS

The Open Racing Car Simulator TORCS (Wymann et al., 2000) is a simulation tool for driving racing. TORCS has been used several times in RL (Loiacono et al., 2010; Koutník et al., 2013; Lillicrap et al., 2016; Mnih et al., 2016). We modified TORCS adding the possibility to configure the car parameters taking inspiration from the "Car Setup Competition" (Loiacono et al., 2013). The agent's observation is a low-dimensional representation of the car's sensors (including speed, focus and wheel speeds), while the action space is composed of steering and acceleration/braking (continuous).

Environment Description The state space of the TORCS environment is composed by 29 dimensions, $\mathcal{S} \subseteq \mathbb{R}^{29}$. The action space is composed by 2 dimensions, $\mathcal{A} \subseteq \mathbb{R}^2$: acceleration/brake action, where $+1$ indicates full acceleration and -1 full brake and steering angle, where -1 indicates maximum left steer and $+1$ maximum right steer. Among all possible parameters, in our experiments, we focused on configuring the Rear and Front Wings and the Front-Rare Brake Repartition. All configuration parameters are normalized in the range $[0, 1]$. The state space space is summarized in Table 7.3 and the configuration parameters in Table 7.4. We consider the following reward function:

$$r(s, a, s') = v'_x \cdot \cos(\alpha'), \tag{7.5}$$

where v'_x is the velocity on the longitudinal direction of the car in state s' and α' is the angle between the car direction and the direction of the track axis. We give a penalty of -1000 if the agent runs backward, if it goes out of track or if the progress in the race is too small. The rationale behind this reward is to encourage the agent to go at high speed and to stay centered with respect to the track.

Policy and Model Approximators The policy we used in the TORCS experiments is a Gaussian Policy parameterized by a fully connected neural network with one hidden layer with 64 neurons with tanh activations. The activation of the last layer is tanh since actions are limited in $[-1, 1]$. The covariance matrix is diagonal and independent of the state. We initialize the policy fitting, via maximum likelihood, a scripted policy (snakeoil) using 45000 samples collected with 30 randomly generated values of the configurable parameters.

Parameter	Description
Rear Wing	Angle of the rear wing.
Front Wing	Angle of the front wing.
Front-Rear Brake Repartition	Repartition of the brake between the front and rear.
Front Anti-Roll Bar	Front Spring.
Rear Anti-Roll Bar	Rear Spring.
Front Left-Right Brake	Brake disk diameter of the front wheels.
Rear Left-Right Brake	Brake disk diameter of the rear wheels.

Table 7.4: *Configuration space of the TORCS experiment. Underlined the parameters we configure in the experiment.*

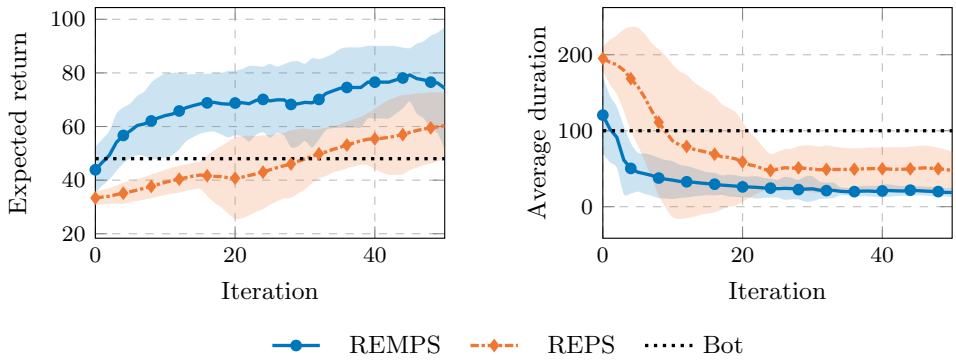


Figure 7.10: *Expected return and episode duration as a function of the number of iterations for the TORCS experiment comparing REMPS, REPS and the bot. 10 runs, 80% c.i.*

We considered a Gaussian model to approximate the dynamics of the task. The mean network is composed of two hidden layers of 64 neurons each with tanh activation. The covariance matrix is diagonal and independent of the state, action and, configurable parameters. The model fitting is performed at the beginning of learning using the same samples employed for fitting the policy.

Experiment The goal of this experiment is to show the ability of REMPS to learn policy and configuration in a continuous state-action space, like a car racing scenario. We consider a configuration space made of three parameters: rear and front wing orientation and brake repartition between front and rear. We start with a policy pretrained via behavioral cloning, using samples collected with a driving bot (snakeoil). Using the same bot, we collect a dataset of episodes with different parameter values, used to train an approximation of the environment. In Figure 7.10, we compare the Expected return and the average lap time for REMPS (with $\text{PROJ}_{\pi,P}$), in which we act on both the policy and the model, and REPS, in which only policy learning is enabled. We can notice that REMPS is able to reach performances larger than those achievable without configuring the environment. In this experiment, we can appreciate another remarkable benefit of environment con-

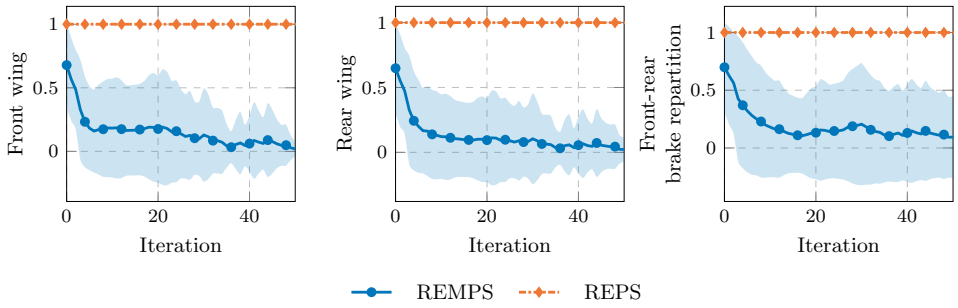


Figure 7.11: *Configurable parameters values and episode duration as a function of the number of iterations for the TORCS experiment comparing REMPS and REPS. 10 runs, 80% c.i.*

figurability: configuring the environment can also speed up the learning process (online performance), as clearly visible in Figure 7.10. In Figure 7.11, we report the behavior of the configurable parameters. We can notice that all parameters tend to be moved towards zero. Indeed, a good behavior in the considered track consists of increasing the speed as much as possible. Therefore, the orientation of the wing tends to be reduced to increase the speed. A similar behavior is visible for the Front-Rear Brake Repartition.

7.6.4 Summary of the Experiments

The experimental evaluation confirmed the benefits of the Conf-MDP in terms of the final performance, which can be achieved by acting on the environment configuration, in addition to the improvement of the agent's policy. The take-home message of this evaluation is that learning in continuous Conf-MDP poses new challenges related to the knowledge of the transition model space and the need for parametric representations. We have shown that REMPS is able to learn in this setting, with the limitations in performance due to the approximation error introduced when resorting to a limited parametric representation. Moreover, REMPS overcomes some limitations of purely gradient-based methods that tend to be trapped in local optima. These two aspects were extensively analyzed in the Chain Domain experiment (Section 7.6.1). Moreover, we illustrated that learning the effect of the configuration parameters on the transition probabilities can be performed during the learning process, with acceptable degradation of the performance. This issue is examined in the Cartpole experiment (Section 7.6.2). Finally, in the TORCS experiment, we observed that configuring the environment can have the side remarkable effect of speeding up the learning process, in addition to increasing the final performance (Section 7.6.3).

Part III

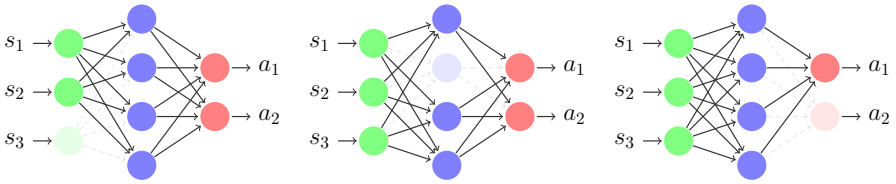
Applications of Configurable Markov Decision Processes

CHAPTER 8

Policy Space Identification

8.1 Introduction

We introduced in Chapter 2 the nature of the interaction between an artificial agent and an environment in the typical RL setting. The agent *perceives* the state of the environment and performs *actions* that trigger an evolution of the state and generate a reward signal. The agent aims at finding an optimal policy, i.e., a prescription of actions that maximizes a performance index. Clearly, the performance of an agent in an environment is constrained by its perception and its actuation possibilities, along with the ability to *map* observations to actions. These three elements (perception, actuation, and mapping) define the *policy space* available to the agent in the learning process. Agents having access to different policy spaces may exhibit different optimal behaviors, even in the same environment. Therefore, the notion of optimality is necessarily connected to the space of policies the agent can access, that we will call *agent's policy space* in the following. While in tabular RL we typically assume access to the complete (and finite) space of Markovian stationary policies, in continuous control the policy space needs to be limited. In policy search methods (Deisenroth et al., 2013), the policies are explicitly modeled considering a parametric function space (Sutton et al., 1999a; Peters and Schaal, 2008) or a kernel space (Deisenroth and Rasmussen, 2011; Levine and Koltun, 2013); but even in value-based RL, a function approximator induces a set of representable (greedy) policies. It is important to point out that the notion of policy space is not just an algorithmic convenience. Indeed, the need to limit the policy space naturally emerges in many industrial applications, where some behaviors have to be avoided for safety reasons.



(a) *Perception limitation* **(b)** *Mapping limitation* **(c)** *Actuation limitation*

Figure 8.1: *An example of policy space modeled as a 1-layer neural network showing a limitation in the (a) perception, (b) mapping, and (c) actuation.*

The knowledge of the agent's policy space turns out to be relevant in several fields of RL. In the Conf-MDP framework, we have observed (Chapter 4) that the best environment configuration is intimately related to the agent's possibilities in terms of policy space. When the configuration activity is performed by an external supervisor, it might be helpful to know which parameters the agent can control in order to select the most appropriate configuration. Furthermore, in the field of Imitation Learning (IL, Osa et al., 2018), having a grasp on the policy space of the expert's agent, can aid the learning process of the imitating policy, mitigating overfitting/underfitting phenomena.

Motivated by the examples presented above, we study the problem of *identifying* the agent's policy space in a Conf-MDP,[1] by observing the agent's behavior and, possibly, exploiting the *configuration* opportunities of the environment. We consider the case in which the agent's policy space is a subset of a known super-policy space Π_Θ induced by a parameter space $\Theta \subseteq \mathbb{R}^d$. Thus, any policy $\pi_{\boldsymbol{\theta}}$ is determined by a d-dimensional parameter vector $\boldsymbol{\theta} \in \Theta$. However, the agent has control over a smaller number $d^{\mathsf{Ag}} < d$ of parameters (which are unknown), while the remaining ones have a fixed value, namely zero.[2] The choice of zero as a fixed value might appear arbitrary, but it is rather a common case in practice. Indeed, the formulation based on the identification of the *parameters* effectively covers the limitations of the policy space related to perception, actuation, and mapping. For instance, in a linear policy, the fact that the agent does not observe a state feature is equivalent to set the corresponding parameters to zero. Similarly, in a neural network, removing a neuron is equivalent to neglecting all of its connections, which in turn can be realized by setting the relative weights to zero. Figure 8.1 shows three examples of policy space limitations in the case of a one hidden layer neural network policy, which can be realized by setting the appropriate weights to zero.

Our goal is to identify the parameters that the agent can control (and possibly change) by observing some demonstrations of an optimal policy π^{Ag} in the policy space Π_Θ.[3] To this end, we formulate the problem as deciding whether each parameter θ_i for $i \in \{1, ..., d\}$ is zero, and we address it by means of a frequentist statistical test. In other words, we check

[1] Although we assume to act in a Conf-MDP, we stress that our primary goal is to identify the policy space of the agent, rather than learning a profitable configuration in the Conf-MDP.

[2] By "controllable" parameter we mean a parameter whose value can be changed by the agent, while the "uncontrollable" parameters are those which are permanently set to zero. This is a way of modeling the limitations of the policy space.

[3] We stress that, since we restrict the search to the policy space Π_Θ, π^{Ag} might be suboptimal compared to the optimal policy in the space of Markovian stationary policies.

whether there is a statistically significant difference between the likelihood of the agent's behavior with the full set of parameters and the one in which θ_i is set to zero. In such a case, we conclude that θ_i is not zero and, consequently, the agent can control it. On the contrary, either the agent cannot control the parameter or zero is the value consciously chosen by the agent.

Indeed, there could be parameters that, given the peculiarities of the environment, are useless for achieving an optimal behavior or whose optimal value is actually zero, while they could prove essential in a different environment. For instance, in a grid world where the goal is to reach the right edge, the vertical position of the agent is useless, while if the goal is to reach the upper right corner both horizontal and vertical positions become relevant. In this spirit, configuring the environment can help the supervisor in identifying whether a parameter set to zero is actually uncontrollable by the agent or just useless in the current environment. Thus, the supervisor can change the environment configuration $\boldsymbol{\omega} \in \Omega$, so that the agent will adjust its policy, possibly by changing the parameter value and revealing whether it can control such a parameter. Consequently, the new configuration should induce an optimal policy in which the considered parameters have a value significantly different from zero. We formalize this notion as the problem of finding the new environment configuration that maximizes the *power* of the statistical test and we propose a surrogate objective for this purpose.

Chapter Outline The chapter is organized as follows. In Section 8.2, we introduce the necessary background on likelihood ratio tests. The *identification rules* (combinatorial and simplified) to perform parameter identification in a fixed environment are presented in Section 8.3 and the simplified one is analyzed in Section 8.4. Section 8.5 shows how to improve them by exploiting the environment configurability. In Section 8.6, we present the connections between policy space identification and existing works in the literature. The experimental evaluation, on discrete and continuous domains, is provided in Section 8.7. Besides studying the ability of our identification rules in identifying the agent's policy space, we apply them to the IL and Conf-MDP frameworks. The results and proofs not reported in this chapter can be found in Appendix A.3.

8.2 Generalized Likelihood Ratio Test

The Generalized Likelihood Ratio test (GLR, Barnard, 1959; Casella and Berger, 2002) aims at testing the goodness of fit of two statistical models. Given a parametric model having density function $p(\cdot|\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \Theta$, we aim at testing the pair of hypothesis:

$$\mathcal{H}_0 : \boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta_0 \qquad \text{vs} \qquad \mathcal{H}_1 : \boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta \backslash \Theta_0,$$

where $\Theta_0 \subset \Theta$ is a subset of the parametric space. Given a dataset $\mathcal{D} = \{X_i\}_{i=1}^n$ sampled independently from $p(\cdot|\boldsymbol{\theta}^{\mathsf{Ag}})$, where $\boldsymbol{\theta}^{\mathsf{Ag}}$ is the true parameter, the GLR statistic is:

$$\Lambda = \frac{\sup_{\boldsymbol{\theta} \in \Theta_0} \{p(\mathcal{D}|\boldsymbol{\theta})\}}{\sup_{\boldsymbol{\theta} \in \Theta} \{p(\mathcal{D}|\boldsymbol{\theta})\}} = \frac{\sup_{\boldsymbol{\theta} \in \Theta_0} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}}{\sup_{\boldsymbol{\theta} \in \Theta} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}}, \tag{8.1}$$

where the likelihood function is defined as:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \widehat{\mathcal{L}}(\boldsymbol{\theta}) = \prod_{i=1}^{n} p(X_i|\boldsymbol{\theta}).$$

Moreover, we denote with $\widehat{\ell}(\boldsymbol{\theta}) = -\log \widehat{\mathcal{L}}(\boldsymbol{\theta})$ the negative log-likelihood function, $\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}\in\Theta}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$ and $\widehat{\boldsymbol{\theta}}_0 \in \arg\max_{\boldsymbol{\theta}\in\Theta_0}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$, i.e., the maximum likelihood solutions in Θ and Θ_0 respectively. Moreover, we define the expectation of the likelihood under the true parameter: $\ell(\boldsymbol{\theta}) = \mathbb{E}_{X_i\sim p(\cdot|\boldsymbol{\theta}^{\mathsf{Ag}})}[\widehat{\ell}(\boldsymbol{\theta})]$. As the maximization is carried out employing the same dataset $\mathcal{D}$ and recalling that $\Theta_0 \subset \Theta$, we have that $\Lambda \in [0, 1]$. It is usually convenient to consider the logarithm of the GLR statistic:

$$\lambda = -2\log\Lambda = 2\left(\widehat{\ell}(\widehat{\boldsymbol{\theta}}_0) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})\right).$$

Therefore, $\mathcal{H}_0$ is rejected for large values of λ, i.e., when the maximum likelihood parameter searched in the restricted set Θ_0 significantly underfits the data $\mathcal{D}$, compared to Θ. Wilk's theorem provides the asymptomatic distribution of λ when $\mathcal{H}_0$ is true (Wilks, 1938; Casella and Berger, 2002).

Theorem 8.1 (Casella and Berger (2002), Theorem 10.3.3)**.** *Let $d = \dim(\Theta)$ and $d_0 = \dim(\Theta_0) < d$. Under suitable regularity conditions (see Casella and Berger (2002) Section 10.6.2), if $\mathcal{H}_0$ is true, then when $n \to +\infty$, the distribution of λ tends to a χ^2 distribution with $d - d_0$ degrees of freedom.*

The *significance* of a test $\alpha \in [0, 1]$, or *type I error* probability, is the probability to reject $\mathcal{H}_0$ when $\mathcal{H}_0$ is true, while the *power* of a test $1 - \beta \in [0, 1]$ is the probability to reject $\mathcal{H}_0$ when $\mathcal{H}_0$ is false, β is the *type II error* probability.

8.3 Policy Space Identification in a Fixed Environment

As we introduced in Section 8.1, we aim at identifying the agent's policy space, by observing a set of demonstrations coming from the optimal policy in the considered policy space $\pi^{\mathsf{Ag}} \in \Pi_\Theta$[4] only, i.e., $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ where $S_i \sim \nu$ and $A_i \sim \pi^{\mathsf{Ag}}(\cdot|S_i)$ sampled independently. $\nu \in \mathscr{P}(\mathcal{S})$ is a sampling distribution over the state space. Although we will present the method for a generic ν, in practice we employ as ν the γ-discounted stationary distribution $\mu_\gamma^{\pi^{\mathsf{Ag}}}$ induced by π^{Ag} (Sutton et al., 1999a). We assume that the agent has control over a limited number of parameters $d^{\mathsf{Ag}} < d$ whose value can be changed during learning, while the remaining $d - d^{\mathsf{Ag}}$ are kept fixed to zero.[5] Given a set of indexes $I \subseteq \{1, ..., d\}$ we define the subset of the parameter space:

$$\Theta_I = \{\boldsymbol{\theta} \in \Theta : \theta_i = 0,\ \forall i \in \{1, ..., d\}\backslash I\}.$$

[4] It is important to stress π^{Ag} is one of the possibly many optimal policies within the policy space Π_Θ, which might be unable to represent the optimal Markovian stationary policy. Furthermore, we do not explicitly report the dependence on the agent's parameter $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta$ as, in the general case, there might exist multiple parameters yielding the same policy π^{Ag}.

[5] The extension of the identification rules to (known) fixed values different from zero is straightforward.

Thus, the set I represents the indexes of the parameters that can be changed if the agent's parameter space were Θ_I. Our goal is to find a set of parameter indexes I^{Ag} that are *sufficient* to explain the agent's policy, i.e., $\pi^{\mathsf{Ag}} \in \Pi_{\Theta_{I^{\mathsf{Ag}}}}$ but also *necessary*, in the sense that when removing any $i \in I^{\mathsf{Ag}}$ the remaining ones are insufficient to explain the agent's policy, i.e., $\pi^{\mathsf{Ag}} \notin \Pi_{\Theta_{I^{\mathsf{Ag}}\setminus\{i\}}}$. We formalize these notions in the following definition.

Definition 8.1 (Correctness)**.** *Let $\pi^{\mathsf{Ag}} \in \Pi_\Theta$. A set of parameter indexes $I^{\mathsf{Ag}} \subseteq \{1, ..., d\}$ is* correct *w.r.t. π^{Ag} if:*

$$\pi^{\mathsf{Ag}} \in \Pi_{\Theta_{I^{\mathsf{Ag}}}} \ \wedge \ \forall i \in I^{\mathsf{Ag}} : \pi^{\mathsf{Ag}} \notin \Pi_{\Theta_{I^{\mathsf{Ag}}\setminus\{i\}}}.$$

We denote with $\mathcal{I}^{\mathsf{Ag}}$ the set of all correct set of parameter indexes I^{Ag}.

Thus, there exist multiple I^{Ag} when multiple parametric representations of the agent's policy π^{Ag} are possible. The uniqueness of I^{Ag} is guaranteed under the assumption that each policy admits a unique representation in Π_Θ, i.e., under the identifiability assumption.

Assumption 8.1 (Identifiability)**.** *The policy space Π_Θ is* identifiable*, i.e., for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, we have that if $\pi_{\boldsymbol{\theta}}(\cdot|s) = \pi_{\boldsymbol{\theta}'}(\cdot|s)$ almost surely for all $s \in \mathcal{S}$ then $\boldsymbol{\theta} = \boldsymbol{\theta}'$.*

The identifiability property allows rephrasing Definition 8.1 in terms of the policy parameters only, leading to the following result.

Lemma 8.2 (Correctness under Identifiability)**.** *Under Assumption 8.1, let $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta$ be the unique parameter such that $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|s) = \pi^{\mathsf{Ag}}(\cdot|s)$ almost surely for all $s \in \mathcal{S}$. Then, there exists a unique set of parameter indexes $I^{\mathsf{Ag}} \subseteq \{1, ..., d\}$ that is correct w.r.t. π^{Ag} defined as:*

$$I^{\mathsf{Ag}} = \left\{ i \in \{1, ..., d\} \ : \ \theta_i^{\mathsf{Ag}} \neq 0 \right\}.$$

Consequently, $\mathcal{I}^{\mathsf{Ag}} = \{I^{\mathsf{Ag}}\}$.

Proof. The uniqueness of I^{Ag} is ensured by Assumption 8.1. Let us rewrite the condition of Definition 8.1 under Assumption 8.1:

$$\begin{aligned}
\pi^{\mathsf{Ag}} \in \Pi_{\Theta_{I^{\mathsf{Ag}}}} \ \wedge \ \forall i \in I^{\mathsf{Ag}} : \pi^{\mathsf{Ag}} &\notin \Pi_{\Theta_{I^{\mathsf{Ag}}\setminus\{i\}}} \\
&\iff \boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta_{I^{\mathsf{Ag}}} \ \wedge \ \forall i \in I^{\mathsf{Ag}} : \boldsymbol{\theta}^{\mathsf{Ag}} \notin \Theta_{I^{\mathsf{Ag}}\setminus\{i\}} && \text{(P.1)} \\
&\iff \forall i \in I^{\mathsf{Ag}} : \theta_i^{\mathsf{Ag}} \neq 0 \ \wedge \ \forall i \in \{1, ..., d\}\setminus I^{\mathsf{Ag}} : \theta_i^{\mathsf{Ag}} = 0 && \text{(P.2)} \\
&\iff I^{\mathsf{Ag}} = \left\{ i \in \{1, ..., d\} \ : \ \theta_i^{\mathsf{Ag}} \neq 0 \right\},
\end{aligned}$$

where line (P.1) follows since there is a unique representation for π^{Ag} determined by parameter $\boldsymbol{\theta}^{\mathsf{Ag}}$ and line (P.2) is obtained from the definition of Θ_I. □

Remark 8.1 (About the Optimality of π^{Ag})**.** *We started this section stating that π^{Ag} is an optimal policy within the policy space Π_Θ. This is motivated by the fact that typically we start with an overparametrized policy space Π_Θ and we seek for the minimal set of parameters that allows the agent reaching an optimal policy within Π_Θ. However, in practice, we usually have access to an ϵ-optimal policy $\pi_\epsilon^{\mathsf{Ag}}$, meaning that the performance*

of $\pi_\epsilon^{\mathsf{Ag}}$ is ϵ-close to the optimal performance.[6] *Nevertheless, the notion of correctness (Definition 8.1) makes no assumptions on the optimality of π^{Ag}. If we replace π^{Ag} with $\pi_\epsilon^{\mathsf{Ag}}$ we will recover a set of parameter indexes I_ϵ^{Ag} that is, in general, different from I_ϵ^{Ag}, but we can still provide some guarantees. If $I^{\mathsf{Ag}} \subseteq I_\epsilon^{\mathsf{Ag}}$, then I_ϵ^{Ag} is sufficient to explain the optimal policy π^{Ag}, but not necessary in general (it might contain useless parameters for π^{Ag}). Instead, if $I^{\mathsf{Ag}} \nsubseteq I_\epsilon^{\mathsf{Ag}}$, then I_ϵ^{Ag} is not sufficient to explain the optimal policy π^{Ag}. In any case, I_ϵ^{Ag} is necessary and sufficient to represent, at least, an ϵ-optimal policy.*

The following two subsections are devoted to the presentation of the *identification rules* based on the application of Definition 8.1 (Section 8.3.1) and Lemma 8.2 (Section 8.3.2) when we only have access to a dataset of samples $\mathcal{D}$. The goal of an identification rule consists in producing a set $\widehat{\mathcal{I}}$, approximating $\mathcal{I}^{\mathsf{Ag}}$. The idea at the basis of our identification rules consists in employing the GLR test to assess the correctness (Definition 8.1 or Lemma 8.2) of a candidate set of indexes.

8.3.1 Combinatorial Identification Rule

In principle, using $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$, we could compute the maximum likelihood parameter $\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}\in\Theta}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$ and employ it with Definition 8.1. However, this approach has, at least, two drawbacks. First, when Assumption 8.1 is not fulfilled, it would produce a single approximate parameter, while multiple choices might be viable. Second, because of the estimation errors, we would hardly get a zero value for the parameters the agent might not control. For these reasons, we employ a GLR test to assess whether a specific set of parameters is zero. Specifically, for all $I \subseteq \{1, ..., d\}$ we consider the pair of hypotheses:

$$\mathcal{H}_{0,I} \,:\, \pi^{\mathsf{Ag}} \in \Pi_{\Theta_I} \qquad \text{vs} \qquad \mathcal{H}_{1,I} \,:\, \pi^{\mathsf{Ag}} \in \Pi_{\Theta\setminus\Theta_I}$$

and the GLR statistic is given by:

$$\lambda_I = -2\log\frac{\sup_{\boldsymbol{\theta}\in\Theta_I}\left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}}{\sup_{\boldsymbol{\theta}\in\Theta}\left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}} = 2\left(\widehat{\ell}(\widehat{\boldsymbol{\theta}}_I) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})\right), \tag{8.2}$$

where the likelihood is defined as:

$$\widehat{\mathcal{L}}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \pi_{\boldsymbol{\theta}}(A_i|S_i),$$

and the maximum likelihood solutions are defined as $\widehat{\boldsymbol{\theta}}_I \in \arg\max_{\boldsymbol{\theta}\in\Theta_I}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$ and $\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}\in\Theta}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$ respectively. We are now ready to state the identification rule derived from Definition 8.1.

Identification Rule 8.1. *The* combinatorial identification rule *with threshold function $\{c_l\}_{l=0}^d$ selects $\widehat{\mathcal{I}}_c$ containing all and only the sets of parameter indexes $I \subseteq \{1, ..., d\}$ such that:*

$$\lambda_I \leqslant c_{|I|} \,\wedge\, \forall i \in I : \lambda_{I\setminus\{i\}} > c_{|I|-1}. \tag{8.3}$$

[6] We can also look at $\pi_\epsilon^{\mathsf{Ag}}$ as the optimal policy within Π_Θ for a different MDP $\mathcal{M}_\epsilon$, that is an approximation of the original MDP $\mathcal{M}$.

Algorithm 8.1: Identification Rule 8.1 (Combinatorial).

Input: dataset $\mathcal{D}$, parameter space Θ, threshold function c (e.g., $c_l = \chi^2_{l,1-\delta/2^d}$)

Output: approximate set of correct sets of parameter indexes $\widehat{\mathcal{I}}_c$

$\widehat{\mathcal{I}}_c \leftarrow \{\}$

$\widehat{\mathcal{L}} = \max_{\boldsymbol{\theta}\in\Theta} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}$

for $I \subseteq \{1, ..., d\}$ *sorted by cardinality* **do**

  $\widehat{\mathcal{L}}_I = \max_{\boldsymbol{\theta}\in\Theta_I} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}$

  $\lambda_I = -2 \log \frac{\widehat{\mathcal{L}}_I}{\widehat{\mathcal{L}}}$

  if $\lambda_I \leqslant c_{|I|}$ ***and*** $\forall i \in I \,:\, \lambda_{I\setminus\{i\}} > c_{|I|-1}$ **then**

    $\widehat{\mathcal{I}}_c \leftarrow \widehat{\mathcal{I}}_c \cup \{I\}$

return $\widehat{\mathcal{I}}_c$

Thus, I is defined in such a way that the null hypothesis $\mathcal{H}_{0,I}$ is not rejected, i.e., I contains parameters that are sufficient to explain the data $\mathcal{D}$, and necessary since for all $i \in I$ the set $I\setminus\{i\}$ is no longer sufficient, as $\mathcal{H}_{0,I\setminus\{i\}}$ is rejected. The threshold function c_l, that depends on the cardinality l of the tested set of indexes, controls the behavior of the tests. In practice, we recommend to set them by exploiting the Wilk's asymptotic approximation (Theorem 8.1) to enforce (asymptotic) guarantees on the type I error. Given a significance level $\delta \in [0, 1]$, since for Identification Rule 8.1 we perform 2^d statistical tests by using the same dataset $\mathcal{D}$, we partition δ using Bonferroni correction and setting $c_l = \chi^2_{l,1-\delta/2^d}$, where $\chi^2_{l,\star}$ is the $\star$-quantile of a chi square distribution with l degrees of freedom. Refer to Algorithm 8.1 for the pseudocode of the identification procedure.[7]

8.3.2 Simplified Identification Rule

Identification Rule 8.1 is hard to be employed in practice, as it requires performing $\mathcal{O}(2^d)$ statistical tests. However, under Assumption 8.1, to retrieve I^{Ag} we do not need to test all subsets, but we can just examine one parameter at a time (see Lemma 8.2). Thus, for all $i \in \{1, ..., d\}$ we consider the pair of hypotheses:

$$\mathcal{H}_{0,i} \,:\, \theta_i^{\mathsf{Ag}} = 0 \qquad \text{vs} \qquad \mathcal{H}_{1,i} \,:\, \theta_i^{\mathsf{Ag}} \neq 0,$$

and define the set of parameters:

$$\Theta_i = \{\boldsymbol{\theta} \in \Theta \,:\, \theta_i = 0\}.$$

The GLR test can be performed straightforwardly, using the following statistic:

$$\lambda_i = -2 \log \frac{\sup_{\boldsymbol{\theta}\in\Theta_i} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}}{\sup_{\boldsymbol{\theta}\in\Theta} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}} = 2 \left(\widehat{\ell}(\widehat{\boldsymbol{\theta}}_i) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})\right), \tag{8.4}$$

[7] The algorithm is designed to output all the sets of controllable parameters explaining the behavior demonstrated by the agent. Clearly, within the set $\widehat{\mathcal{I}}_c$ we could select the "most reliable" set, i.e., the one with maximum value of the likelihood function.

Algorithm 8.2: Identification Rule 8.2 (Simplified).

Input: dataset $\mathcal{D}$, parameter space Θ, threshold function c (e.g., $c_1 = \chi^2_{l,1-\delta/d}$)
Output: approximate correct set of parameter indexes $\{\widehat{I}_c\}$

$\widehat{I}_c \leftarrow \{\}$
$\widehat{\mathcal{L}} = \max_{\boldsymbol{\theta}\in\Theta} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}$
for $i \in \{1, ..., d\}$ **do**
  $\widehat{\mathcal{L}}_i = \max_{\boldsymbol{\theta}\in\Theta_i} \left\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\right\}$
  $\lambda_i = -2 \log \frac{\widehat{\mathcal{L}}_i}{\widehat{\mathcal{L}}}$
  if $\lambda_i > c_1$ **then**
    $\widehat{I}_c \leftarrow \widehat{I}_c \cup \{i\}$
return $\{\widehat{I}_c\}$

where the likelihood is defined as $\widehat{\mathcal{L}}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \pi_{\boldsymbol{\theta}}(A_i|S_i)$, $\widehat{\boldsymbol{\theta}}_i = \arg\max_{\boldsymbol{\theta}\in\Theta_i}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$ and $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\Theta}\{\widehat{\mathcal{L}}(\boldsymbol{\theta})\}$.[8] In the spirit of Lemma 8.2, we define the following identification rule.

Identification Rule 8.2. *The* simplified identification rule *with threshold function c_1 selects $\widehat{\mathcal{I}}_c$ containing the unique set of parameter indexes $\widehat{I}_c$ such that:*

$$\widehat{I}_c = \{i \in \{1, ..., d\} : \lambda_i > c_1\}\,. \tag{8.5}$$

Therefore, the identification rule constructs $\widehat{I}_c$ by taking all the indexes $i \in \{1, ..., d\}$ such that the corresponding null hypothesis $\mathcal{H}_{0,i} : \theta_i^{\mathsf{Ag}} = 0$ is rejected, i.e., those for which there is statistical evidence that their value is not zero. Similarly to the combinatorial identification rule, we recommend setting the threshold function c_1 based on the Wilk's approximation. Given a significance level $\delta \in [0, 1]$, since we perform d statistical tests, we employ Bonferroni correction and we set $c_1 = \chi^2_{1,1-\delta/d}$. Refer to Algorithm 8.2 for the pseudocode of the identification rule.

This second procedure requires a test for every parameter, i.e., $\mathcal{O}(d)$ instead of $\mathcal{O}(2^d)$ tests. However, it comes with the cost of assuming the identifiability property. What happens if we employ this second procedure in a case where the assumption does not hold?

Example 8.1. *Consider for instance the case in which two parameters θ_1 and θ_2 are exchangeable, we will include none of them in $\widehat{I}_c$ as, individually, they are not necessary to explain the agent's policy, while the pair $(\theta_1, \theta_2)^T$ is indeed necessary. We will discuss how to enforce identifiability (Assumption 8.1), for the case of policies belonging to the exponential family, in the following section.*

Remark 8.2 (On Frequentist and Bayesian Statistical Tests)**.** *In this work, we restrict our attention to frequentist statistical tests, but, in principle, the same approaches can be extended to the Bayesian setting (Jeffreys, 1935). Indeed, the GLR test admits a Bayesian*

[8] This setting is equivalent to a particular case the combinatorial rule in which $\mathcal{H}_{\star,i} \equiv \mathcal{H}_{\star,\{1,...,d\}\setminus\{i\}}$, with $\star \in \{0, 1\}$ and, consequently, $\lambda_i \equiv \lambda_{\{1,...,d\}\setminus\{i\}}$ and $\Theta_i = \Theta_{\{1,...,d\}\setminus\{i\}}$.

counterpart, known as the Bayes Factor *(BF, Goodman, 1999; Morey et al., 2016). We consider the same setting presented in Section 8.2 in which we aim at testing the null hypothesis* $\mathcal{H}_0 : \boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta_0$*, against the alternative* $\mathcal{H}_1 : \boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta \backslash \Theta_0$*. We take the Bayesian perspective, looking at each* $\boldsymbol{\theta}$ *not as an unknown fixed quantity but as a realization of* prior *distributions on the parameters defined in terms of the hypothesis:* $p(\boldsymbol{\theta}|\mathcal{H}_\star)$ *for* $\star \in \{0, 1\}$*. Thus, given a dataset* $\mathcal{D} = \{X_i\}_{i=1}^n$*, we can compute the likelihood of* $\mathcal{D}$ *given a parameter* $\boldsymbol{\theta}$ *as usual:* $p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^n p(X_i|\boldsymbol{\theta})$*. Combining the likelihood and the prior, we define the* Bayes Factor *as:*

$$\Lambda^{BF} = \frac{p(\mathcal{D}|\mathcal{H}_0)}{p(\mathcal{D}|\mathcal{H}_1)} = \frac{\int_\Theta p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{H}_0) \, \mathrm{d}\boldsymbol{\theta}}{\int_\Theta \underbrace{p(\mathcal{D}|\boldsymbol{\theta})}_{\textit{likelihood}} \underbrace{p(\boldsymbol{\theta}|\mathcal{H}_1)}_{\textit{prior}} \mathrm{d}\boldsymbol{\theta}}$$

The Bayesian approach has the clear advantage of incorporating additional domain knowledge by means of the prior. Furthermore, if also a prior on the hypothesis is available $p(\mathcal{H}_\star)$ *for* $\star \in \{0, 1\}$ *it is possible to compute the ratio of the* posterior *probability of each hypothesis:*

$$\underbrace{\frac{p(\mathcal{H}_0|\mathcal{D})}{p(\mathcal{H}_1|\mathcal{D})}}_{\textit{posterior ratio}} = \underbrace{\frac{p(\mathcal{D}|\mathcal{H}_0)}{p(\mathcal{D}|\mathcal{H}_1)}}_{\textit{Bayes factor}} \cdot \underbrace{\frac{p(\mathcal{H}_0)}{p(\mathcal{H}_1)}}_{\textit{prior ratio}} .$$

Compared to the GLR test, the Bayes factor provides richer information, since we can compute the likelihood of each hypothesis, given the data $\mathcal{D}$*. However, like any Bayesian approach, the choice of the prior turns out to be of crucial importance. The computationally convenient prior (which might allow computing the integral in closed form) is typically not correct, leading to a biased test. In this sense, GLR replaces the integral with a single-point approximation centered in the maximum likelihood estimate. For these reasons, we leave the investigation of Bayesian approaches for policy space identification as future work.*

8.4 Analysis for the Exponential Family

In this section, we provide an analysis of the Identification Rule 8.2 for a policy $\pi_{\boldsymbol{\theta}}$ linear in some state features $\boldsymbol{\phi}$ that belongs to the exponential family.[9] The section is organized as follows. We first introduce the exponential family, deriving a concentration result of independent interest (Theorem 8.4), and then we apply it for controlling the identification errors made by our identification rule (Theorem 8.5). We provide in the following an overview of the main results, while we defer to the Appendix A.3 the complete derivation.

8.4.1 Exponential Family

We refer to the definition of exponential family given in Brown (1986).

[9] We limit our analysis to Identification Rule 8.2 since we will show that, in the case of linear policies belonging to the exponential family, the identifiability property can be easily enforced.

Policy	Gaussian	Boltzmann
$\mathcal{A}$	$\mathbf{a} \in \mathbb{R}^k$	$a_i \in \{a_1, ..., a_{k+1}\}$
$\pi_{\tilde{\boldsymbol{\theta}}}$	$\frac{1}{(2\pi)^{\frac{k}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{a}-\tilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s))^T \boldsymbol{\Sigma}^{-1}(\mathbf{a}-\tilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s))}$	$\begin{cases} \frac{e^{\tilde{\boldsymbol{\theta}}_i^T \boldsymbol{\phi}(s)}}{1+\sum_{j=1}^k e^{\tilde{\boldsymbol{\theta}}_j^T \boldsymbol{\phi}(s)}} & \text{if } i \leqslant k \\ \frac{1}{1+\sum_{j=1}^k e^{\tilde{\boldsymbol{\theta}}_j^T \boldsymbol{\phi}(s)}} & \text{if } i = k \end{cases}$
$\mathbf{t}$	$\boldsymbol{\Sigma}^{-1}\mathbf{a} \otimes \boldsymbol{\phi}(s)$	$\begin{cases} \mathbf{e}_i \otimes \boldsymbol{\phi}(s) & \text{if } i \leqslant k \\ \mathbf{0} & \text{if } i = k+1 \end{cases}$
h	$\frac{1}{(2\pi)^{\frac{k}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{a}^T \boldsymbol{\Sigma}^{-1}\mathbf{a}}$	1

Table 8.1: *Action space $\mathcal{A}$, probability density function $\pi_{\tilde{\boldsymbol{\theta}}}$, sufficient statistic $\mathbf{t}$, and function h for the Gaussian linear policy with fixed covariance and the Boltzmann linear policy. For convenience of representation $\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^{k \times q}$ is a matrix and $\boldsymbol{\theta} = \mathrm{vec}(\widetilde{\boldsymbol{\theta}}^T) \in \mathbb{R}^d$, with $d = kq$. We denote with $\mathbf{e}_i$ the i-th vector of the canonical basis of $\mathbb{R}^k$ and with $\otimes$ the Kronecker product.*

Definition 8.2 (Exponential Family)**.** *Let $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^q$ be a feature function. The policy space Π_Θ is a space of linear policies, belonging to the exponential family, if $\Theta = \mathbb{R}^d$ and all policies $\pi_{\boldsymbol{\theta}} \in \Pi_\Theta$ have probability density function of the form:*

$$\pi_{\boldsymbol{\theta}}(a|s) = h(a) \exp\left\{\boldsymbol{\theta}^T \mathbf{t}\,(s,a) - A(\boldsymbol{\theta}, s)\right\}, \tag{8.6}$$

where h is a positive function, $\mathbf{t}\,(s,a)$ is the sufficient statistic *depending on the state via the features $\boldsymbol{\phi}$, i.e., $\mathbf{t}\,(s,a) = \mathbf{t}(\boldsymbol{\phi}(s), a)$, and $A(\boldsymbol{\theta}, s) = \log \int_{\mathcal{A}} h(a) \exp\{\boldsymbol{\theta}^T \mathbf{t}(s,a)\} \mathrm{d}a$ is the log partition function. We denote with $\overline{\mathbf{t}}(s, a, \boldsymbol{\theta}) = \mathbf{t}(s,a) - \mathbb{E}_{\overline{A} \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s, \overline{A})\right]$ the centered sufficient statistic.*

This definition allows modeling the linear policies that are often used in RL (Deisenroth et al., 2013). Table 8.1 shows how to map the Gaussian linear policy with fixed covariance, typically used in continuous action spaces, and the Boltzmann linear policy, suitable for finite action spaces, to Definition 8.2. The complete derivation is reported in Appendix B.1).

For the sake of the analysis, we enforce the following assumption concerning the tail behavior of the policy $\pi_{\boldsymbol{\theta}}$.

Assumption 8.2 (Subgaussianity)**.** *For any $\boldsymbol{\theta} \in \Theta$ and for any $s \in \mathcal{S}$ the centered sufficient statistic $\overline{\mathbf{t}}(s, a, \boldsymbol{\theta})$ is subgaussian with parameter $\sigma \geqslant 0$, i.e., for any $\boldsymbol{\alpha} \in \mathbb{R}^d$:*

$$\mathop{\mathbb{E}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\exp\left\{\boldsymbol{\alpha}^T \overline{\mathbf{t}}(s, A, \boldsymbol{\theta})\right\}\right] \leqslant \exp\left\{\frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 \sigma^2\right\}.$$

The subgaussianity property is easily met by the Gaussian and Boltzmann policies. Proposition B.4 of Appendix B.3 proves that, when the features are uniformly bounded, i.e., $\|\boldsymbol{\phi}(s)\|_2 \leqslant \Phi_{\max}$ for all $s \in \mathcal{S}$, Assumption 8.2 is fulfilled by both Boltzmann and

Gaussian linear policies with parameter $\sigma = 2\Phi_{\max}$ and $\sigma = \Phi_{\max}/\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})}$ respectively.

Furthermore, limited to the policies complying with Definition 8.2, the identifiability (Assumption 8.1) can be restated in terms of the Fisher Information Matrix (FIM, Rothenberg, 1971; Little et al., 2010).

Lemma 8.3 (Rothenberg (1971), Theorem 3)**.** *Let Π_Θ be a policy space, as in Definition 8.2. Then, under suitable regularity conditions (see Rothenberg (1971)), if the Fisher Information matrix (FIM) $\mathcal{F}(\boldsymbol{\theta})$:*

$$\mathcal{F}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{S\sim\nu \\ A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\bar{\mathbf{t}}(S,A,\boldsymbol{\theta})\bar{\mathbf{t}}(S,A,\boldsymbol{\theta})^T\right] \tag{8.7}$$

is non-singular for all $\boldsymbol{\theta} \in \Theta$, then Π_Θ is identifiable. In this case, we denote with $\lambda_{\min} = \inf_{\boldsymbol{\theta}\in\Theta}\{\lambda_{\min}(\mathcal{F}(\boldsymbol{\theta}))\} > 0$.

Proposition B.2 of Appendix B.2 shows that a sufficient condition for the identifiability in the case of Gaussian and Boltzmann linear policies is that the second moment matrix of the feature vector $\mathbb{E}_{S\sim\nu}\left[\boldsymbol{\phi}(S)\boldsymbol{\phi}(S)^T\right]$ is non-singular along with the fact that the policy $\pi_{\boldsymbol{\theta}}$ plays each action with positive probability for the Boltzmann policy.

Remark 8.3 (How to enforce identifiability?)**.** *Requiring that $\mathbb{E}_{S\sim\nu}\left[\boldsymbol{\phi}(S)\boldsymbol{\phi}(S)^T\right]$ is full rank is essentially equivalent to require that all features ϕ_i are linearly independent for all $i \in \{1,...,d\}$. This condition can be easily met with a preprocessing phase that removes the linearly dependent features, for instance by employing Principal Component Analysis (Jolliffe, 2011). For this reason, in our experimental evaluation, we will always consider the case of linearly independent features.*

We are now ready to present a concentration result, of independent interest, for the parameters and the negative log-likelihood that represents the central tool of our analysis.

Theorem 8.4. *Under Assumption 8.1 and Assumption 8.2, let $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ be a dataset of $n > 0$ independent samples, where $S_i \sim \nu$ and $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$. Let $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\widehat{\ell}(\boldsymbol{\theta})\}$ and $\boldsymbol{\theta}^{\mathsf{Ag}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\ell(\boldsymbol{\theta})\}$. If the empirical FIM:*

$$\widehat{\mathcal{F}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|S_i)} \left[\bar{\mathbf{t}}(S_i,A,\boldsymbol{\theta})\bar{\mathbf{t}}(S_i,A,\boldsymbol{\theta})^T\right] \tag{8.8}$$

has a positive minimum eigenvalue $\widehat{\lambda}_{\min} > 0$ for all $\boldsymbol{\theta} \in \Theta$, then, for any $\delta \in [0,1]$, with probability at least $1-\delta$ it holds that:

$$\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2 \leqslant \frac{\sigma}{\widehat{\lambda}_{\min}}\sqrt{\frac{2d}{n}\log\frac{2d}{\delta}}.$$

Furthermore, with probability at least $1-\delta$, it holds that, individually:

$$\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) \leqslant \frac{d^2\sigma^4}{\widehat{\lambda}_{\min}^2 n}\log\frac{2d}{\delta} \quad \textit{and}$$

$$\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) \leqslant \frac{d^2\sigma^4}{\widehat{\lambda}_{\min}^2 n}\log\frac{2d}{\delta}.$$

Proof Sketch. The idea of the proof is to first obtain a probabilistic bound on the parameter difference in norm $\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2$. This result is given in Theorem A.14. Then, we use the latter result together with Taylor expansion to bound the differences $\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})$ and $\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})$, as in Corollary A.15. The full derivation can be found in Appendix A.3.1. □

The theorem shows that the L_2-norm of the difference between the maximum likelihood parameter $\widehat{\boldsymbol{\theta}}$ and the true parameter $\boldsymbol{\theta}^{\mathsf{Ag}}$ concentrates with rate $\mathcal{O}(n^{-1/2})$ while the likelihood $\widehat{\ell}$ and its expectation ℓ concentrate with faster rate $\mathcal{O}(n^{-1})$. Note that the result assumes that the empirical FIM $\widehat{\mathcal{F}}(\boldsymbol{\theta})$ has a strictly positive eigenvalue $\widehat{\lambda}_{\min} > 0$. This condition can be enforced as long as the true Fisher matrix $\mathcal{F}(\boldsymbol{\theta})$ has a positive minimum eigenvalue $\lambda_{\min}$, i.e., under identifiability assumption (Lemma 8.3) and given a sufficiently large number of samples. Proposition B.6 of Appendix B.2 provides the minimum number of samples such that with high probability it holds that $\widehat{\lambda}_{\min} > 0$.

8.4.2 Identification Rule Analysis

We are now ready to start the analysis of Identification Rule 8.2. The goal of the analysis is, informally, to bound the probability of an identification error, as a function of the number of samples n and the threshold function c_1. For this purpose, we define the following quantities.

Definition 8.3. *Consider an identification rule producing $\widehat{I}$ as approximate parameter index set. We define the significance α and the power $1-\beta$ of the identification rule as:*

$$\alpha = \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : i \in \widehat{I}\right),$$
$$\beta = \mathbb{P}\left(\exists i \in I^{\mathsf{Ag}} : i \notin \widehat{I}\right).$$

Thus, α represents the probability that the identification rule selects a parameter that the agent does not control, whereas β is the probability that the identification rule does not select a parameter that the agent does control.[10] By employing the results we derived for the exponential family (Theorem 8.4) we can now bound α and β.

Theorem 8.5. *Let $\widehat{I}_c$ be the set of parameter indexes selected by the Identification Rule 8.2 obtained using $n > 0$ i.i.d. samples collected with $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}$, with $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta$. Then, under Assumption 8.1 and Assumption 8.2, let $\boldsymbol{\theta}_i^{\mathsf{Ag}} = \arg\min_{\boldsymbol{\theta} \in \Theta_i}\{\ell(\boldsymbol{\theta})\}$ for all $i \in \{1, ..., d\}$ and $\xi = \min\left\{1, \frac{\lambda_{\min}}{\sigma^2}\right\}$. If $\widehat{\lambda}_{\min} \geqslant \frac{\lambda_{\min}}{2\sqrt{2}}$ and $\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) \geqslant c_1$, it holds that:*

$$\alpha \leqslant 2d \exp\left\{-\frac{c_1 \lambda_{\min}^2 n}{16 d^2 \sigma^4}\right\},$$
$$\beta \leqslant (2d-1) \sum_{i \in I^{\mathsf{Ag}}} \exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - c_1\right) \lambda_{\min} \xi n}{16(d-1)^2 \sigma^2}\right\}.$$

[10] We use the symbols α and β to highlight the analogy between these probabilities and the type I and type II error probabilities of a statistical test.

Proof Sketch. Concerning $\alpha = \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : i \in \widehat{I}_c\right)$, we employ a technique similar to that of Lemma 2 in (Garivier and Kaufmann, 2019) to remove the existential quantification. Instead, for $\beta = \mathbb{P}\left(\exists i \in I^{\mathsf{Ag}} : i \notin \widehat{I}_c\right)$ we first perform a union bound over $i \in I^{\mathsf{Ag}}$ and then we bound the individual $\mathbb{P}\left(i \notin \widehat{I}_c\right)$. The full derivation can be found in Appendix A.3.2. □

In principle, we could employ Theorem 8.5 to derive a proper value of c_1 and n, given a user-defined value of α and β. Unfortunately, their expressions depend on $\lambda_{\min}$ which is unknown in practice. As already mentioned in the previous sections, we recommend employing the Wilk's asymptotic approximation to set the threshold function as $c_1 = 1 - \delta/d$. This choice allows an asymptotic control of the significance of the identification rule.

Theorem 8.6. *Let $\widehat{I}_c$ be the set of parameter indexes selected by the Identification Rule 8.2 obtained using $n > 0$ i.i.d. samples collected with $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}$, with $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta$. Then, under suitable regularity conditions (see Casella and Berger (2002) Section 10.6.2), if $c_1 = \chi^2_{l,1-\delta/d}$ it holds that $\alpha \leqslant \delta$ when $n \to +\infty$.*

Proof. Starting from the definition of α, we first perform a union bound over $i \notin I^{\mathsf{Ag}}$ to remove the existential quantification.

$$\alpha = \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : i \in \widehat{I}_c\right) = \mathbb{P}\left(\bigvee_{i \notin I^{\mathsf{Ag}}} i \in \widehat{I}_c\right) \leqslant \sum_{i \notin I^{\mathsf{Ag}}} \mathbb{P}\left(i \in \widehat{I}_c\right).$$

Now, we bound each $\mathbb{P}\left(i \in \widehat{I}_c\right)$ individually, recalling that λ_i is distributed asymptotically as a χ^2 distribution with 1 degree of freedom and that $c_1 = \chi_{1,1-\delta/d}$.

$$\mathbb{P}\left(i \in \widehat{I}_c\right) = \mathbb{P}\left(\lambda_i > \chi_{1,1-\delta/d}\right) \to \frac{\delta}{d}, \quad n \to \infty.$$

Thus, we have that when $n \to +\infty$:

$$\alpha \leqslant \frac{d - d^{\mathsf{Ag}}}{d}\delta \leqslant \delta. \tag{P.3}$$

□

8.5 Policy Space Identification in a Configurable Environment

The identification rules presented so far are unable to distinguish between a parameter set to zero because the agent cannot control it, or because zero is its optimal value. To overcome this issue, we employ the Conf-MDP properties to select a configuration in which the parameters we want to examine have an optimal value other than zero. More formally, like in Chapter 7, we consider a class of parametric Conf-MDPs whose transition model $P_{\boldsymbol{\omega}}$ is parametrized in $\boldsymbol{\omega} \in \Omega \subseteq \mathbb{R}^q$. We denote with $J(\boldsymbol{\theta}, \boldsymbol{\omega})$ for every $\boldsymbol{\theta}, \boldsymbol{\omega} \in \Theta \times \Omega$ the expected return of executing policy $\pi_{\boldsymbol{\theta}}$ with the transition model $P_{\boldsymbol{\omega}}$.

Intuitively, if we want to test whether the agent can control parameter θ_i, we should place the agent in an environment $\boldsymbol{\omega}_i \in \Omega$ where θ_i is "maximally important" for the optimal policy. This intuition is justified by Theorem 8.5, since to maximize the *power* of the test $(1 - \beta)$, all other things being equal, we should maximize the log-likelihood gap

$\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})$, i.e., parameter θ_i should be essential to justify the agent's behavior. Let $I \subseteq \{1, ..., d\}$ be a set of parameter indexes we want to test, our ideal goal is to find the environment $\boldsymbol{\omega}_I$ such that:

$$\boldsymbol{\omega}_I \in \arg\max_{\boldsymbol{\omega}\in\Omega} \left\{ \ell(\boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega})) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega})) \right\}, \tag{8.9}$$

where $\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}) \in \arg\max_{\boldsymbol{\theta}\in\Theta}\{J(\boldsymbol{\theta}, \boldsymbol{\omega})\}$ and $\boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega}) \in \arg\max_{\boldsymbol{\theta}\in\Theta_I}\{J(\boldsymbol{\theta}, \boldsymbol{\omega})\}$ are the parameters of the optimal policies in the environment $P_{\boldsymbol{\omega}}$ considering Π_Θ (the full policy space) and Π_{Θ_I} (the policy space in which θ_i is fixed to zero) as policy spaces respectively. Clearly, given the samples $\mathcal{D}$ collected with a single optimal policy $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}$ in a single environment $P_{\boldsymbol{\omega}_0}$, solving problem in Equation (8.9) is hard as it requires performing an off-distribution optimization both on the space of policy parameters and configurations. For these reasons, we consider a surrogate objective that assumes that the optimal parameter in the new configuration can be reached by performing a single gradient step.[11]

Theorem 8.7. *Let $I \in \{1, ..., d\}$ and $\overline{I} = \{1, ..., d\}\backslash I$. For a vector $\mathbf{v} \in \mathbb{R}^d$, we denote with $\mathbf{v}|_I$ the vector obtained by setting to zero the components in I. Let $\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0) \in \Theta$ the initial parameter. Let $\alpha \geqslant 0$ be a learning rate, $\boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega}) = \boldsymbol{\theta}_0 + \alpha\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega})|_I$ and $\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}) = \boldsymbol{\theta}_0 + \alpha\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega})$. Then, under Assumption 8.1, we have:*

$$\ell(\boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega})) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega})) \geqslant \frac{\lambda_{\min}\alpha^2}{2} \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega})|_{\overline{I}} \right\|_2^2.$$

Proof. By second-order Taylor expansion of ℓ and recalling that $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega})) = \mathbf{0}$, we have:

$$\begin{aligned}
\ell(\boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega})) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega})) &\geqslant \frac{\lambda_{\min}}{2} \left\| \boldsymbol{\theta}_I^{\mathsf{Ag}}(\boldsymbol{\omega}) - \boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}) \right\|_2^2 \\
&= \frac{\lambda_{\min}}{2} \left\| \boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0) + \alpha\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega})|_I - \boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0) - \alpha\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega}) \right\|_2^2 \\
&= \frac{\lambda_{\min}\alpha^2}{2} \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega})|_{\overline{I}} \right\|_2^2.
\end{aligned}$$

□

Thus, we maximize the L_2-norm of the gradient components that correspond to the parameters we want to test. Since we have at our disposal only samples $\mathcal{D}$ collected with the current policy $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}$ and in the current environment $\boldsymbol{\omega}_0$, we have to perform an off-distribution optimization over $\boldsymbol{\omega}$. To this end, we employ an approach analogous to that of (Metelli et al., 2018b), as introduced in Section 3.3.2, where we optimize the empirical version of the objective with a penalization that accounts for the distance between the distribution over trajectories:

$$\begin{aligned}
\mathcal{C}_I(\boldsymbol{\omega}/\boldsymbol{\omega}_0) = &\left\| \underbrace{\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0), \boldsymbol{\omega}/\boldsymbol{\omega}_0)|_{\overline{I}}}_{\text{gradient estimator}} \right\|_2^2 \\
&- \zeta \sqrt{\frac{1}{n}\sum_{t=0}^{T-1} \gamma^{2t} d_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}_0}}\right)},
\end{aligned} \tag{8.10}$$

[11] This idea shares some analogies with the *adapted parameter* in the meta-learning setting (Finn et al., 2017).

Algorithm 8.3: Identification Rule 8.2 (Simplified) with Environment Configuration.

Input: parameter space Θ, configuration space Ω, threshold function c_l, number of configuration attempts N_{conf}
Output: approximate correct set of parameter indexes $\{\widehat{I}_c\}$

Initialize $\boldsymbol{\omega}_0$ arbitrarily
Collect $\mathcal{D}_0$ observing π_0^{Ag} in environment $P_{\boldsymbol{\omega}_0}$
Run the Identification Rule 8.2 on $\mathcal{D}_0$ and obtain $\widehat{I}_0$
$\widehat{I} \leftarrow \widehat{I}_0$
for $i \in \{1, ..., d\} : i \notin \widehat{I}$ **do**
    $\boldsymbol{\omega}_{i,0} \leftarrow \boldsymbol{\omega}_0$
    $\mathcal{D}_{i,0} \leftarrow \mathcal{D}_0$
    for $j = 1, ..., N_{\text{conf}}$ **do**
        Optimize $\mathcal{C}_{\{i\}}(\boldsymbol{\omega}/\boldsymbol{\omega}_{i,j-1})$ getting $\boldsymbol{\omega}_{i,j}$
        Collect $\mathcal{D}_{i,j}$ observing $\pi_{i,j}^{\mathsf{Ag}}$ in environment $P_{\boldsymbol{\omega}_{i,j}}$
        Run the Identification Rule 8.2 on $\mathcal{D}_{i,j}$ and obtain $\widehat{I}_{i,j}$
        $\widehat{I} \leftarrow \widehat{I} \cup \widehat{I}_{i,j}$
return $\{\widehat{I}\}$

where $\zeta \geqslant 0$ is a regularization parameter and $d_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}_0}}\right)$ is the Rényi divergence between the length t trajectory distributions. This penalization term favors configurations $\boldsymbol{\omega}$ not too far away from $\boldsymbol{\omega}_0$. We assume to have access to a dataset of trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^n$ independently collected using policy $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega_0})}$ in the environment $P_{\boldsymbol{\omega}_0}$. Using $\mathcal{D}$, we can estimate the gradient:

$$\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}/\boldsymbol{\omega}_0) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{T-1}\gamma^t R_{\tau_i,t}\underbrace{\prod_{j=0}^{t}\frac{p_{\boldsymbol{\omega}}(S_{\tau_i,j+1}|S_{\tau_i,j}, A_{\tau_i,j})}{p_{\boldsymbol{\omega}_0}(S_{\tau_i,j+1}|S_{\tau_i,j}, A_{\tau_i,j})}}_{\text{importance weight}}\sum_{j=0}^{t}\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}\left(A_{\tau_i,j}|S_{\tau_i,j}\right).$$

The expression is obtained starting from the well-known G(PO)MDP gradient estimator (Section 3.3.1) and adapting for off-distribution estimation, by introducing the importance weight (Metelli et al., 2018b). The dissimilarity penalization term corresponds to the 2-Rényi divergence (Rényi, 1961) that is estimated as the second moment of the importance weight:

$$\widehat{d}_2\left(\mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}}} \| \mathbb{P}_t^{\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}(\boldsymbol{\omega}_0)}, P_{\boldsymbol{\omega}_0}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{t=1}^{T}\frac{p_{\boldsymbol{\omega}}(S_{\tau_i,t+1}|S_{\tau_i,t}, A_{\tau_i,t})}{p_{\boldsymbol{\omega}_0}(S_{\tau_i,t+1}|S_{\tau_i,t}, A_{\tau_i,t})}\right)^2.$$

We refer the reader to Section 3.3.2 and to (Metelli et al., 2018b, 2020b) for the theoretical background behind the choice of this objective function. We report the pseudocode of the identification procedure in a configurable environment for the Identification Rule 8.2 in Algorithm 8.3, while the pseudocode for Identification Rule 8.2 is reported in Algorithm 8.4.

Algorithm 8.4: Identification Rule 8.1 (Combinatorial) with Environment Configuration.

Input: parameter space Θ, configuration space Ω, threshold function c, number of configuration attempts N_{conf}
Output: approximate set of correct sets of parameter indexes $\widehat{\mathcal{I}}_c$

Initialize $\boldsymbol{\omega}_0$ arbitrarily
Collect $\mathcal{D}_0$ observing π_0^{Ag} in environment $P_{\boldsymbol{\omega}_0}$
Run the Identification Rule 8.1 on $\mathcal{D}_0$ with δ' and get $\widehat{\mathcal{I}}_0$
$\widehat{\mathcal{I}} \leftarrow \widehat{\mathcal{I}}_0$
for $I \subseteq \{1, ..., d\} : I \notin \widehat{\mathcal{I}}$ **do**
    $\boldsymbol{\omega}_{i,0} \leftarrow \boldsymbol{\omega}_0$
    $\mathcal{D}_{i,0} \leftarrow \mathcal{D}_0$
    for $j = 1, ..., N_{\text{conf}}$ **do**
        Optimize $\mathcal{C}_I(\boldsymbol{\omega}/\boldsymbol{\omega}_{i,j-1})$ getting $\boldsymbol{\omega}_{i,j}$
        Collect $\mathcal{D}_{i,j}$ observing $\pi_{i,j}^{\mathsf{Ag}}$ in environment $P_{\boldsymbol{\omega}_{i,j}}$
        Run the Identification Rule 8.1 on $\mathcal{D}_{i,j}$ and obtain $\widehat{\mathcal{I}}_{i,j}$
        $\widehat{\mathcal{I}} \leftarrow \widehat{\mathcal{I}} \cup \widehat{\mathcal{I}}_{i,j}$
return $\widehat{\mathcal{I}}$

8.6 Connections with Existing Work

The idea of identifying the policy parameters a learning agent can control by observing its behavior by employing a statistical test, to the best of our knowledge, has not been explored in the literature yet. We believe that this abstract problem is by itself of interest for understanding the *capabilities* of the agent in terms of perception, actuation, and mapping. Furthermore, knowing the parameters an agent can control can help other subfields of RL. In this section, we discuss how policy space identification can be beneficial for Imitation Learning (IL, Osa et al., 2018, Section 8.6.1) algorithms and help a supervisor acting in a Conf-MDP (Section 8.6.2).

8.6.1 Connections with Imitation Learning

IL is the framework in which an agent learns a policy by observing an expert, i.e., an agent playing a (near) optimal policy. Selecting the parameters that an agent can control can be interpreted as applying a form of regularization to the problem of imitating the expert. In the IL literature, a widely used technique is based on *entropy regularization* (Neu et al., 2017), which was employed in several successful algorithms, such as Maximum Causal Entropy IRL methods (MCE, Ziebart et al., 2008, 2010), and Generative Adversarial IL (Ho and Ermon, 2016). Alternatively, other approaches aim at enforcing a *sparsity* constraint on the recovered policy parameters (e.g., Lee et al., 2018; Reddy et al., 2019; Brantley et al., 2020). In the field of IL, we believe that policy space identification could help to prevent possible over/underfitting phenomena. Indeed, knowing the expert's policy space means knowing a suitable hypothesis space in which to look for the imitating policy. While the methods mentioned above state the IL problem at a policy level, i.e., finding an

imitating policy, IRL has the goal of recovering a reward function that explains the expert's choices (Ng and Russell, 2000). The reward is known to be a more succinct and transferable representation of the optimal behavior than the optimal policy. The identification of the parameters controlled by the agent can help to understand which class of objectives the agent is actually able to optimize, with possible benefits in the reward reconstruction phase. More directly, the IRL approaches based on the policy gradient (e.g., Pirotta and Restelli, 2016; Metelli et al., 2017; Tateo et al., 2017; Ramponi et al., 2020; Metelli et al., 2020c) require a parametric representation of the expert's policy, whose choice might affect the quality of the recovered reward function.

8.6.2 Connections with Configurable Markov Decision Processes

The knowledge of the agent's policy space could be of crucial importance when the learning process involves the presence of an external supervisor. As intuition suggests, the best environment configuration is closely related to the agent's capabilities in terms of policy space. For instance, in a car racing problem, the best car configuration depends on the car driver and has to be selected, by a track engineer (the supervisor), according to the driver's skills. Thus, the external supervisor has to be aware of the agent's policy space to select the most appropriate configuration.

It is worth emphasizing that we use the Conf-MDP notion for two purposes. First, we propose the problem of learning the optimal configuration in a Conf-MDP as a motivating example in which the knowledge of the policy space is valuable. Second, we use the environment configurability as a tool to improve the identification of the policy space.

8.7 Experimental Results

In this section, we present the experimental results, focusing on three aspects of policy space identification.

- In Section 8.7.1, we provide experiments to assess the quality of our identification rules in terms of the ability to correctly identifying the parameters controlled by the agent.
- In Section 8.7.2, we focus on the application of policy space identification to IL, comparing our identification rules with commonly employed regularization techniques.
- In Section 8.7.3, we consider the Conf-MDP framework and we show how properly identifying the parameters controlled by the agent allows learning better (more specific) environment configurations.

The complete experimental campaign, together with the implementation details and the hyperparameter values can be found in Metelli et al. (2019c).

8.7.1 Identification Rules Experiments

In this section, we provide two experiments to test the ability of our identification rules in properly selecting the parameters the agent controls in different settings. We start with an

experiment on a discrete grid world to highlight the beneficial effects of environment configuration in the parameter identification. Then, we provide an experiment on a simulated car driving domain in which we compare the combinatorial and the simplified identification rules.

Discrete Grid World The grid world environment is a simple representation of a two-dimensional world (5×5 cells) in which an agent has to reach a target position by moving in the four directions. Whenever an action is performed there is a small probability of failure (0.1) triggering a random action. The initial position of the agent and the target position are drawn at the beginning of each episode from a Boltzmann distribution $\mu_{0,\boldsymbol{\omega}}$. The agent plays a Boltzmann linear policy $\pi_{\boldsymbol{\theta}}$ with binary features $\boldsymbol{\phi}$ indicating its current row and column and the row and column of the goal.[12] For each run, the agent can control a subset I^{Ag} of the parameters $\boldsymbol{\theta}_{I^{\mathsf{Ag}}}$ associated with those features, which is randomly selected. Furthermore, the supervisor can configure the environment by changing the parameters $\boldsymbol{\omega}$ of the initial state distribution $\mu_{0,\boldsymbol{\omega}}$.[13] Thus, the supervisor can induce the agent to explore certain regions of the grid world and, consequently, change the relevance of the corresponding parameters in the optimal policy.

The goal of this set of experiments is to show the advantages of configuring the environment when performing the policy space identification using rule 8.2. Figure 8.2 shows the empirical $\widehat{\alpha}$ and $\widehat{\beta}$, i.e., the fraction of parameters that the agent does not control that are wrongly selected and the fraction of those the agent controls that are not selected respectively, as a function of the number m of episodes used to perform the identification. We compare two cases: *conf* where the identification is carried out by also configuring the environment, i.e., optimizing Equation (8.10), and *no-conf* in which the identification is performed in the original environment only. In both cases, we can see that $\widehat{\alpha}$ is almost independent of the number of samples, as it is directly controlled by the threshold function c_1. Differently, $\widehat{\beta}$ decreases as the number of samples increases, i.e., the power of the test $1-\widehat{\beta}$ increases with m. Remarkably, we observe that configuring the environment gives a significant advantage in understanding the parameters controlled by the agent w.r.t. using a fixed environment, as $\widehat{\beta}$ decreases faster in the *conf* case. This phenomenon also justifies empirically our choice of objective (Equation (8.10)) for selecting the new environment.

Simulated Car Driving We consider a simple version of a car driving simulator, in which an agent has to drive a car to reach the end of the track without running off the road. The control directives are the acceleration and the steering, and are expressed through a two-dimensional bounded action space. The car has four sensors oriented in different directions: $-\frac{\pi}{4}$, $-\frac{\pi}{6}$, $\frac{\pi}{6}$, $\frac{\pi}{4}$ w.r.t. the axis pointing toward the front of the car. The values of these sensors are the normalized distances from the car to the nearest road margin along the direction of the sensor, or the maximum value if the margin is outside the range of the sensor. The complete set of state features is made up of the normalized car speed and the values of the four sensors. In the experiments, the agent has access to the speed and the sensor at angles $\frac{\pi}{6}$ and $\frac{\pi}{4}$. The track consists of a single road segment with a

[12] The features are selected to fulfill Lemma 8.3.

[13] Although in our Conf-MDP definition we limit the configurability part of the environment to the transition model, assuming that also the initial state distribution can be configured is not an issue. Indeed, it is always possible to define an MDP in which the effect of the initial state distribution is included in the transition model.

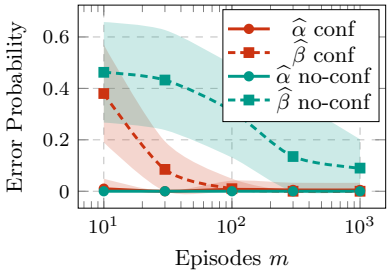


Figure 8.2: Discrete Grid World*: $\widehat{\alpha}$ and $\widehat{\beta}$ error for* conf *and* no-conf *cases varying the number of episodes. 25 runs 95% c.i.*

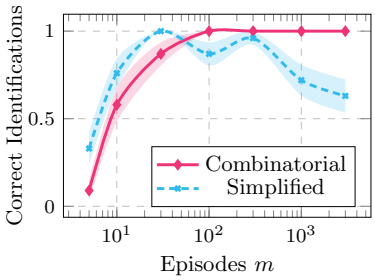


Figure 8.3: Simulated Car Driving*: fraction of correct identifications varying the number of episodes. 100 runs 95% c.i.*

fixed curvature. The rewards are given proportionally to the speed of the car, i.e., greater speeds yield higher rewards. The episode finishes when the car goes outside the road, and a negative reward is given in this case, when the track is completed, or when a maximum number of time steps is elapsed.

The purpose of this experiment is to show a case in which the identifiability assumption (Assumption 8.1) may not be satisfied. The policy $\pi_{\boldsymbol{\theta}}$ is modeled as a Gaussian policy whose mean is computed via a single hidden layer neural network with 8 neurons. Some of the sensors are not available to the agent, our goal is to identify which ones the agent can perceive. In Figure 8.3, we compare the performance of the Identification Rules 8.1 (Combinatorial) and 8.2 (Simplified), showing the fraction of runs that correctly identify the policy space. We note that, while for a small number of samples the simplified rule seems to outperform, when the number of samples increases the combinatorial rule displays remarkable stability, approaching the correct identification in all the runs. This is explained by the fact that, when multiple representations for the same policy are possible (like in this case when having a neural network as policy), considering one parameter at a time might induce the simplified rule to select a wrong set of parameters.

8.7.2 Imitation Learning Experiment

In this section, we present an experiment to study the application of policy space identification to the IL framework. The goal of this experiment consists in showing that if we know which parameters are actually controlled by the expert agent, we can mitigate overfitting/underfitting phenomena, with a general benefit on the process of learning the imitating policy. This experiment is conducted in the grid world domain, introduced in Section 8.7.1, using the same setting. In each run, the expert agent plays a (near) optimal Boltzmann policy $\pi_{\boldsymbol{\theta}^{\text{Ag}}}$ that makes use of a subset of the available parameters and provides a dataset $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^{n}$ of n samples coming from m episodes.

As we mentioned in Section 8.6.1, in the IL framework knowing the policy space of the expert agent means properly tailoring the hypothesis space in which we search for the imitation policy. For this reason, we propose a comparison with common regularization techniques, applied to maximum likelihood estimation. Figure 8.4 shows on the left the

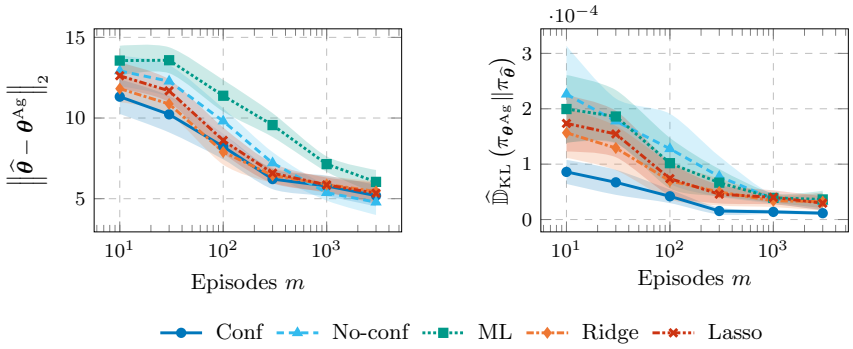


Figure 8.4: Discrete Grid World*: Norm of the difference between the expert's parameter $\boldsymbol{\theta}^{\mathrm{Ag}}$ and the estimated parameter $\widehat{\boldsymbol{\theta}}$ (left) and expected KL-divergence between the expert's policy $\pi_{\boldsymbol{\theta}^{\mathrm{Ag}}}$ and the estimated policy $\pi_{\widehat{\boldsymbol{\theta}}}$ (right) as a function of the number of collected episodes m. 25 runs, 95% c.i.*

norm of the parameter difference $\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathrm{Ag}}\right\|_2$ between the parameter recovered by the different IL methods $\widehat{\boldsymbol{\theta}}$ and the true parameter employed by the expert $\boldsymbol{\theta}^{\mathrm{Ag}}$, whereas on the right we plot the estimated expected KL-divergence between the imitation policy and the expert's policy computed as:

$$\widehat{\mathbb{D}}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\theta}^{\mathrm{Ag}}} \| \pi_{\widehat{\boldsymbol{\theta}}}\right) = \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left(\pi_{\boldsymbol{\theta}^{\mathrm{Ag}}}(\cdot|S_i) \| \pi_{\widehat{\boldsymbol{\theta}}}(\cdot|S_i)\right).$$

The lines *Conf* and *No-conf* refer to the results of ML estimation obtained by restricting the policy space to the parameters identified by our simplified rule with and without employing environment configurability respectively (precisely as in Section 8.7.1). *ML*, *Ridge*, and *Lasso* correspond to maximum likelihood estimation in the full parameter space. Specifically, they are obtained by minimizing the objective:

$$\mathcal{Q}(\boldsymbol{\theta}; \lambda^{\mathrm{R}}, \lambda^{\mathrm{R}}) = \underbrace{-\sum_{i=1}^{n} \log \pi_{\boldsymbol{\theta}}(A_i|S_i)}_{\widehat{\ell}(\boldsymbol{\theta})\text{ log-likelihood}} + \underbrace{\lambda^{\mathrm{R}} \left\|\boldsymbol{\theta}\right\|_2^2}_{\text{ridge}} + \underbrace{\lambda^{\mathrm{L}} \left\|\boldsymbol{\theta}\right\|_1}_{\text{lasso}}.$$

For ML we perform no regularization ($\lambda^{\mathrm{R}} = \lambda^{\mathrm{L}} = 0$), for Ridge we set $\lambda^{\mathrm{R}} = 0.001$ and $\lambda^{\mathrm{L}} = 0$, and for Lasso we have $\lambda^{\mathrm{R}} = 0$ and $\lambda^{\mathrm{L}} = 0.001$.

We observe that Conf, i.e., the usage of our identification rule, together with environment configuration, outperforms the other methods. This is more evident in the expected KL-divergence plot (right), which is a more robust index compared to the norm of the parameter difference (left). Ridge and Lasso regularizations display good behavior, better than both the identification rule without configuration (No-Conf) and the plain maximum likelihood without regularization (ML). This illustrates two important points. First, it confirms the benefits of configuring the environment for policy space identification. Second, it shows that a proper selection of the parameters controlled by the agent allows improving

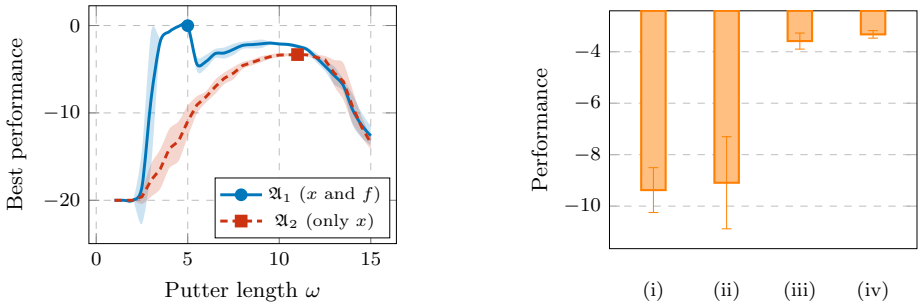


Figure 8.5: Mingolf*: Performance of the optimal policy varying the putter length ω for agents $\mathfrak{A}_1$ and $\mathfrak{A}_2$ (left) and performance of the optimal policy for agent $\mathfrak{A}_2$ with four different strategies for selecting ω (right). 100 runs 95% c.i.*

over standard ML, which tends to overfit.[14]

It is worth noting that the specific IL setting we consider, i.e., the availability of an initial dataset $\mathcal{D}$ of expert's demonstrations with no further interaction allowed[15] rules out from the comparison a large body of the literature that requires the possibility to interact with the expert or with the environment (e.g., Ho and Ermon, 2016; Lee et al., 2018). Nevertheless, these IL algorithms could be in principle adapted to this challenging no-interaction setting at the cost of restoring to off-policy estimation techniques (Owen, 2013), that however might inject further uncertainty in the learning process.

8.7.3 Conf-MDP Experiment

In the Minigolf environment (Lazaric et al., 2007), an agent hits a ball using a putter with the goal of reaching the hole in the minimum number of attempts. Surpassing the hole causes the termination of the episode and a large penalization. The agent selects the force applied to the putter by playing a Gaussian policy linear in some polynomial features (complying to Lemma 8.3) of the distance from the hole (x) and the friction of the green (f). Specifically, we consider the following polynomial features:

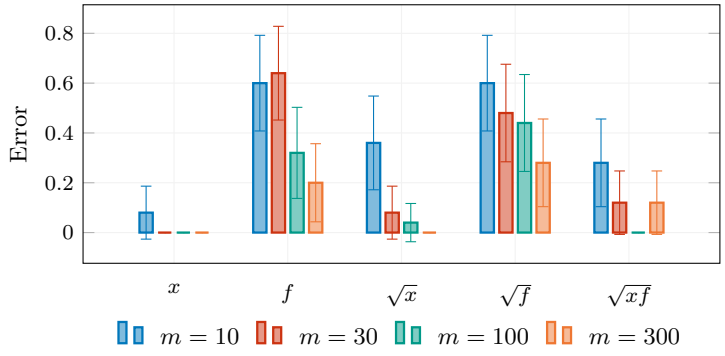
$$\boldsymbol{\phi}(x, f) = \left(1,\, x,\, f,\, \sqrt{x},\, \sqrt{f},\, \sqrt{xf}\right)^T.$$

When an action is performed a Gaussian noise is added whose magnitude depends on the green friction and on the action itself.

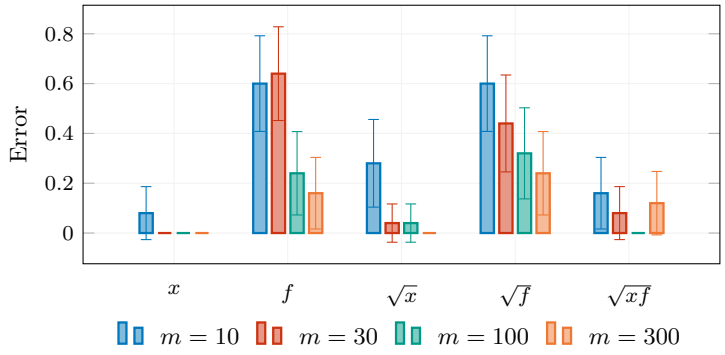
Experiment with fixed features The goal of this experiment is to highlight that knowing the policy space might be of crucial importance when learning in a Conf-MDP. We consider two agents: $\mathfrak{A}_1$ has access to both the x and f whereas $\mathfrak{A}_2$ knows only x. Thus, we expect that $\mathfrak{A}_1$ learns a policy that allows reaching the hole in a smaller number of hits,

[14] It is worth noting that the classical regularization techniques, like ridge and lasso, require choosing the regularization hyperparameter $\lambda^\star$ with $\star \in \{R, L\}$. In our experiments, we searched for the best parameter in $\{0.0001, 0.001, 0.01, 0.1, 1\}$.

[15] This setting was recently defined "truly batch model-free" (Ramponi et al., 2020).



(a) *Without configuration*



(b) *With configuration*

Figure 8.6: *Experiment with randomly chosen features on the minigolf domain for different numbers of episodes m. 100 runs, 95% c.i.*

compared to $\mathfrak{A}_2$, as it can calibrate force according to friction; whereas $\mathfrak{A}_2$ has to be more conservative, being unaware of f. Thus, while agent $\mathfrak{A}_1$ perceives all the features, agent $\mathfrak{A}_2$ has access to $(1,\ x,\ \sqrt{x})^T$ only. There is also a supervisor in charge of selecting, for the two agents, the best putter length ω, i.e., the configurable parameter of the environment.

Figure 8.5-left shows the performance of the optimal policy as a function of the putter length ω. We can see that for agent $\mathfrak{A}_1$ the optimal putter length is $\omega^{\mathsf{Ag}}_{\mathfrak{A}_1} = 5$ while for agent $\mathfrak{A}_2$ is $\omega^{\mathsf{Ag}}_{\mathfrak{A}_2} = 11.5$. Figure 8.5-right compares the performance of the optimal policy of agent $\mathfrak{A}_2$ when the putter length ω is chosen by the supervisor using four different strategies. In (i) the configuration is sampled uniformly in the interval $[1, 15]$. In (ii) the supervisor employs the optimal configuration for agent $\mathfrak{A}_1$ ($\omega = 5$), i.e., assuming the agent is aware of the friction. (iii) is obtained by selecting the optimal configuration of the policy space produced by using our identification rule 8.2. Finally, (iv) is derived by employing an oracle that knows the true agent's policy space ($\omega = 11.5$). We can see that the performance of the identification procedure (iii) is comparable with that of the oracle (iv) and notably higher than the performance when employing an incorrect policy space

(ii).

Experiment with randomly chosen features In the following, we report an additional experiment in the minigolf domain in which the features that the agent can perceive are randomly selected at the beginning, comparing the case in which we do not configure the environment and the case in which environment configuration is performed, and for different number of episodes collected. Although, less visible w.r.t. to the previous examples, we can see that for some features (e.g., $\sqrt{x}$ and $\sqrt{xf}$) the environment configurability is beneficial (Figure 8.6).

8.7.4 Summary of the Experiments

The experimental evaluation highlights some essential points. First, we have shown that configuring the environment is beneficial for speeding up the identification process (Section 8.7.1). This aspect is analyzed in the Grid World experiment, showing that when configuring the environment is possible, the performance of the identification rules improves. Second, we have verified that policy space identification can improve the quality of the policy derived through imitation learning (Section 8.7.2). Finally, the identification of the policy space brings advantages to the learning process in a Conf-MDP, helping to choose wisely the most suitable environment configuration. This is particularly visible in the Minigolf experiment, in which we have illustrated that a wrong identification might result in a suboptimal choice of the environment configuration (Section 8.7.3).

CHAPTER 9

Control Frequency Adaptation

9.1 Introduction

In the previous chapters, we modeled the sequential decision-making problem as a *discrete-time* MDP (Puterman, 2014), or Conf-MDP (Metelli et al., 2018a), whenever altering some parts of the environment is allowed. In these models, the control signal is issued at discrete time instants. However, many relevant real-world problems are more naturally defined in the continuous-time domain (Luenberger, 1979). Even though a branch of literature has studied RL in *continuous-time* MDPs (e.g., Bradtke and Duff, 1994; Munos and Bourgine, 1997; Doya, 2000), the majority of the research has focused on the discrete-time formulation, which appears to be a necessary, but effective, approximation.

Intuitively, increasing the *control frequency* of the system offers the agent more control opportunities, possibly leading to improved performance as the agent has access to a larger *policy space*. This might wrongly suggest that we should control the system with the highest frequency possible, within its physical limits. However, in the RL framework, the environment dynamics is unknown, thus, a too fine discretization could result in an undesired effect, making the problem harder to solve. Indeed, any RL algorithm needs samples to figure out (implicitly or explicitly) how the environment evolves as an effect of the agent's actions. When increasing the control frequency, the *advantage* of individual actions becomes infinitesimal, making them almost indistinguishable for standard *value-based* RL approaches (Tallec et al., 2019). As a consequence, the *sample complexity* increases. Instead, low frequencies allow the environment to evolve longer, making the effect of individual actions more easily detectable. Furthermore, in the presence of a

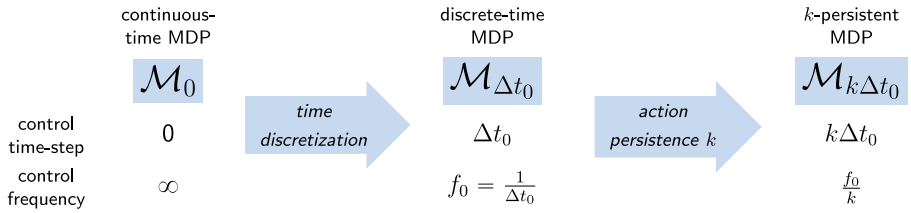


Figure 9.1: *Graphical representation of the discretization process and application of action persistence.*

system characterized by a "slowly evolving" dynamics, the gain obtained by increasing the control frequency might become negligible. Finally, in robotics, lower frequencies help to overcome some partial observability issues, like action execution delays (Kober et al., 2013).

Therefore, we experience a fundamental *trade-off* in the control frequency choice that involves the policy space (larger at high frequency) and the sample complexity (smaller at low frequency). Thus, it seems natural to wonder: "*what is the optimal control frequency?*" An answer to this question can disregard neither the task we are facing nor the learning algorithm we intend to employ. Indeed, the performance loss we experience by reducing the control frequency strictly depends on the properties of the system and, thus, of the task. Similarly, the dependence of the sample complexity on the control frequency is related to how the learning algorithm will employ the collected samples.

In this chapter, we analyze and exploit this trade-off in the context of batch RL (Lange et al., 2012), with the goal of enhancing the learning process and achieving higher performance. It is worth noting that the control frequency can be seen as an *environmental parameter* of a Conf-MDP, that can be configured externally. In this sense, we can look at the choice of the control frequency as a form of *environment configuration* having an effect on the transition dynamics. Although we know in advance that the optimal control frequency is the largest one, when only finite samples are available, smaller frequencies can help to improve the learning experience.

We assume to have access to a discrete-time MDP $\mathcal{M}_{\Delta t_0}$, called base MDP, which is obtained from the time discretization of a continuous-time MDP with fixed base control time step Δt_0, or equivalently, a control frequency equal to $f_0 = \frac{1}{\Delta t_0}$. In this setting, we want to select a suitable *control time step* Δt that is an integer multiple of the base time step Δt_0, i.e., $\Delta t = k\Delta t_0$ with $k \in \mathbb{N}_{\geqslant 1}$. This process is graphically represented in Figure 9.1.[1] Any choice of k generates an MDP $\mathcal{M}_{k\Delta t_0}$ obtained from the base one $\mathcal{M}_{\Delta t_0}$ by altering the transition model so that each action is repeated for k times. For this reason, we refer to k as the action *persistence*, i.e., the number of decision epochs in which an action is kept fixed. It is possible to appreciate the same effect in the base MDP $\mathcal{M}_{\Delta t_0}$ by executing a (non-Markovian and non-stationary) policy that persists every action

[1] We are considering the *near-continuous* time setting. This is almost w.l.o.g. compared to the continuous time since the discretization time step Δt_0 can be chosen to be arbitrarily small. Typically, a lower bound on Δt_0 is imposed by the physical limitations of the system. Thus, we restrict the search of Δt from the continuous set $\mathbb{R}_{>0}$ to the discrete set $\{k\Delta t_0 \,:\, k \in \mathbb{N}_{\geqslant 1}\}$. Moreover, considering an already discretized MDP simplifies the mathematical treatment.

for k time steps. The idea of repeating actions has been previously employed, although heuristically, with deep RL architectures (Lakshminarayanan et al., 2017).

Chapter Outline The chapter is organized as follows. We start in Section 9.2 elaborating on the notion of action persistence and showing that it can be represented by a suitable modification of the Bellman operators, which preserves the contraction property and, consequently, allows deriving the corresponding value functions. Since increasing the duration of the control time step $k\Delta t_0$ has the effect of degrading the performance of the optimal policy, in Section 9.3, we derive an algorithm-independent bound for the difference between the optimal value functions of MDPs $\mathcal{M}_{\Delta t_0}$ and $\mathcal{M}_{k\Delta t_0}$, which holds under Lipschitz conditions. Then, in Section 9.4, we apply the notion of action persistence in the batch RL scenario, proposing and analyzing an extension of Fitted Q-Iteration (FQI, Ernst et al., 2005). The resulting algorithm, *Persistent Fitted Q-Iteration* (PFQI) takes as input a target persistence k and estimates the corresponding optimal value function, assuming to have access to a dataset of samples collected in the base MDP $\mathcal{M}_{\Delta t_0}$. Once we estimate the value function for a set of candidate persistences $\mathcal{K} \subset \mathbb{N}_{\geqslant 1}$, we aim at selecting the one that yields the best performing greedy policy. Thus, we introduce a persistence selection heuristic able to approximate the optimal persistence, without requiring further interactions with the environment (Section 9.5). After having revised the approaches related to action persistence (Section 9.6), we present an experimental evaluation on benchmark domains, to confirm our theoretical findings and evaluate our persistence selection method (Section 9.7). We conclude in Section 9.8 by discussing some open questions related to action persistence and presenting some preliminary results.

9.2 Persisting Actions in MDPs

With the phrase "executing a policy π at persistence k", with $k \in \mathbb{N}_{\geqslant 1}$, we mean the following type of agent-environment interaction. At decision step $t = 0$, the agent selects an action according to its policy $A_0 \sim \pi(\cdot|S_0)$. Action A_0 is kept fixed, or *persisted*, for the subsequent $k-1$ decision steps, i.e., actions $A_1, ..., A_{k-1}$ are all equal to A_0. Then, at decision step $t = k$, the agent queries again the policy $A_k \sim \pi(\cdot|S_k)$ and persists action A_k for the subsequent $k-1$ decision steps and so on. In other words, the agent employs its policy only at decision steps t that are integer multiples of the persistence k ($t \bmod k = 0$). Clearly, the usual execution of π corresponds to persistence 1.

9.2.1 Duality of Action Persistence

Unsurprisingly, the execution of a Markovian stationary policy π at persistence $k > 1$ produces a behavior that, in general, cannot be represented by executing any Markovian stationary policy at persistence 1. Indeed, at any decision step t, such a policy needs to remember which action was taken at the previous decision step $t-1$ (thus it is non-Markovian with memory 1) and has to understand whether to select a new action based on t (so it is non-stationary).

Definition 9.1 (k-persistent policy)**.** *Let $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. For any $k \in \mathbb{N}_{\geqslant 1}$, the* k-persistent policy *induced by π is a history-dependent policy $\pi_k \in \Pi^{\mathrm{HR}}$,*

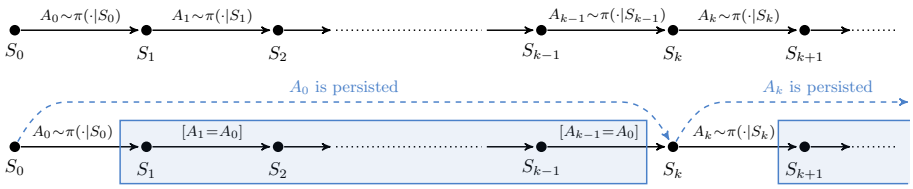


Figure 9.2: *Agent-environment interaction without (top) and with (bottom) action persistence, highlighting duality. The transition generated by the k-persistent MDP $\mathcal{M}_k$ is the cyan dashed arrow, while the actions played by the k-persistent policy are inside the cyan rectangle.*

defined for every $t \in \mathbb{N}$, state-ending history $h_t = (s_0, a_0, r_1 \ldots, s_{t-1}, a_{t-1}, r_t, s_t) \in \mathcal{H}_{\mathcal{S},t}$ and $a \in \mathcal{A}$ as:

$$\pi_{t,k}(\mathrm{d}a|h_t) = \begin{cases} \pi(\mathrm{d}a|s_t) & \text{if } t \bmod k = 0 \\ \delta_{a_{t-1}}(\mathrm{d}a) & \text{otherwise} \end{cases}. \tag{9.1}$$

Moreover, we denote with $\Pi_k = \{(\pi_{t,k})_{t\in\mathbb{N}} : \pi \in \Pi^{\mathrm{SR}}\}$ the set of the k-persistent policies.

Clearly, for $k = 1$ we recover policy π as we always satisfy the condition $t \bmod k = 0$ i.e., $\pi = \pi_{t,1}$ for all $t \in \mathbb{N}$. We refer to this interpretation of action persistence as *policy view*.

A different perspective towards action persistence consists in looking at the effect of the original policy π in a suitably modified MDP. To this purpose, we introduce the (state-action) persistent transition probability kernel $P^\delta : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S} \times \mathcal{A})$ defined for every $(s, a), (s', a') \in \mathcal{S} \times \mathcal{A}$ as:

$$P^\delta(\mathrm{d}s', \mathrm{d}a'|s, a) = P(\mathrm{d}s'|s, a)\delta_a(\mathrm{d}a'). \tag{9.2}$$

The crucial difference between P^π and P^δ is that the former samples the action a' to be executed in the next state s' according to π, whereas the latter replicates in state s' action a that was previously executed in state s. We are now ready to define the k-persistent MDP.[2]

Definition 9.2 (k-persistent MDP)**.** *Let $\mathcal{M}$ be an MDP. For any $k \in \mathbb{N}_{\geqslant 1}$, the* k-persistent MDP *is the following MDP $\mathcal{M}_k = \left(\mathcal{S}, \mathcal{A}, P_k, \mu_0, R_k, \gamma^k\right)$, where P_k and R_k are the k-persistent transition model and reward model respectively, defined for every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and every $s' \in \mathcal{S}$ and $r \in \mathbb{R}$ as:*

$$P_k(\mathrm{d}s'|s, a) = \left((P^\delta)^{k-1}P\right)(\mathrm{d}s'|s, a), \tag{9.3}$$

$$R_k(\mathrm{d}r|s, a) = \sum_{i=0}^{k-1} \gamma^i \left((P^\delta)^i R\right)(\mathrm{d}r|s, a), \tag{9.4}$$

and $r_k(s, a) = \int_{\mathbb{R}} r R_k(\mathrm{d}r|s, a) = \sum_{i=0}^{k-1} \gamma^i \left((P^\delta)^i r\right)(s, a)$ is the reward function, uniformly bounded by $R_{\max} \frac{1-\gamma^k}{1-\gamma}$.

[2]For the sake of simplicity, we consider reward models depending on the current state and current action only.

The k-persistent transition model P_k keeps action a fixed for $k-1$ steps while making the state evolve according to P. Similarly, the k-persistent reward R_k provides the cumulative discounted reward over k steps in which a is persisted. We define the transition kernel $P_k^\pi : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S} \times \mathcal{A})$ for every $(s, a), (s', a') \in \mathcal{S} \times \mathcal{A}$ as:

$$P_k^\pi(\mathrm{d}s', \mathrm{d}a' | s, a) = P_k(\mathrm{d}s' | s, a)\pi(\mathrm{d}a' | s').$$

Clearly, for $k = 1$ we recover the base MDP, i.e., $\mathcal{M} = \mathcal{M}_1$. Indeed, if $\mathcal{M}$ is the base MDP $\mathcal{M}_{\Delta t_0}$, the k-persistent MDP $\mathcal{M}_k$ corresponds to $\mathcal{M}_{k\Delta t_0}$ (Figure 9.1). We typically omit the subscript Δt_0 for brevity, whenever clear from the context. Therefore, executing policy π in $\mathcal{M}_k$ at persistence 1 is equivalent to executing policy π at persistence k in the original MDP $\mathcal{M}$. We refer to this interpretation of persistence as *environment view* (Figure 9.2).

Thus, solving the base MDP $\mathcal{M}$ in the space of k-persistent policies Π_k (Definition 9.1), thanks to this *duality*, is equivalent to solving the k-persistent MDP $\mathcal{M}_k$ (Definition 9.2) in the space of Markovian stationary policies Π^{SR}.

Remark 9.1 (Persistence as Environment Configurability)**.** *As we already mentioned in Section 9.1, the persistence $k \in \mathbb{N}_{\geqslant 1}$ can be seen as an environmental parameter affecting the transition model P, the reward model R, and the discount factor γ, which can be externally configured with the goal to improve the learning process for the agent. In this sense, the MDP $\mathcal{M}_k$ can be seen as a Conf-MDP with parameter $k \in \mathbb{N}_{\geqslant 1}$. More specifically, we are considering a slightly extended version of the Conf-MDP, compared to that of Chapter 4, in which the reward model and the discount factor can be configured, in addition to the transition model. This is, by the way, an interesting setting in which configuring the agent reward function (although in a quite constrained manned) is meaningful for the learning process.*

Remark 9.2 (Persistence as Reducing the Planning Horizon)**.** *A persistence of k induces a k-persistent MDP $\mathcal{M}_k$ with smaller discount factor γ^k. Therefore, the effective horizon in $\mathcal{M}_k$ is $\frac{1}{1-\gamma^k} < \frac{1}{1-\gamma}$. Interestingly, the end effect of persisting actions is similar to reducing the planning horizon, by explicitly reducing the discount factor of the task (Petrik and Scherrer, 2008; Jiang et al., 2016) or setting a maximum trajectory length (Farahmand et al., 2016).*

9.2.2 Persistent Bellman Operators

When executing policy π at persistence k in the base MDP $\mathcal{M}$, we can evaluate its performance starting from any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, inducing a Q-function that we denote with Q_k^π and call *k-persistent action-value function* of π. Thanks to duality, Q_k^π is also the action-value function of policy π when executed in the k-persistent MDP $\mathcal{M}_k$. Therefore, Q_k^π is the fixed point of the Bellman Expectation Operator of $\mathcal{M}_k$, i.e., the operator $T_k^\pi : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathscr{B}(\mathcal{S} \times \mathcal{A})$ defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as:

$$(T_k^\pi f)(s, a) = r_k(s, a) + \gamma^k \int_{\mathcal{S}} P_k^\pi(\mathrm{d}s', \mathrm{d}a' | s, a) f(s, a).$$

We call this operator *k-persistent Bellman Expectation Operator*. Similarly, thanks to duality, the optimal Q-function in the space of k-persistent policies Π_k, denoted by Q_k^* and called *k-persistent optimal action-value function*, corresponds to the optimal Q-function of the k-persistent MDP, i.e., $Q_k^*(s,a) = \sup_{\pi \in \Pi^{\mathrm{SR}}} \{Q_k^\pi(s,a)\}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. As a consequence, Q_k^* is the fixed point of the Bellman Optimal Operator of $\mathcal{M}_k$, i.e., $T_k^* : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathscr{B}(\mathcal{S} \times \mathcal{A})$ defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ as:

$$(T_k^* f)(s,a) = r_k(s,a) + \gamma^k \int_{\mathcal{S}} P_k(\mathrm{d}s'|s,a) \sup_{a' \in \mathcal{A}} \{f(s',a')\},$$

We call this operator *k-persistent Bellman Optimal Operator*. Since they are the operators for the k-persistent MDP $\mathcal{M}_k$, both T_k^π and T_k^* are γ^k-contractions in L_∞-norm and their unique fixed points are the value functions Q_k^π and Q_k^* respectively. We now prove that the k-persistent Bellman operators are obtained as the composition of the base operators T^π and T^*.

Theorem 9.1. *Let $\mathcal{M}$ be an MDP, $k \in \mathbb{N}_{\geqslant 1}$ and $\mathcal{M}_k$ be the k-persistent MDP. Let $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. Then, T_k^π and T_k^* can be expressed as:*

$$T_k^\pi = \left(T^\delta\right)^{k-1} T^\pi \quad \textit{and} \quad T_k^* = \left(T^\delta\right)^{k-1} T^*, \tag{9.5}$$

where $T^\delta : \mathscr{B}(\mathcal{S} \times \mathcal{A}) \to \mathscr{B}(\mathcal{S} \times \mathcal{A})$ is the Bellman Persistent Operator*, defined for every bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ as:*

$$\left(T^\delta f\right)(s,a) = r(s,a) + \gamma \int_{\mathcal{S}} \int_{\mathcal{A}} P^\delta(\mathrm{d}s', \mathrm{d}a'|s,a) f(s',a'). \tag{9.6}$$

Proof. We derive the result by explicitly writing the definitions of the k-persistent transition model P_k and k-persistent reward distribution R_k in terms of P, R and γ in the definition of the k-persistent Bellman expectation operator T_k^π. Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
(T_k^\pi f)(s,a) &= r_k(s,a) + \gamma^k (P_k^\pi f)(s,a) \\
&= \sum_{i=0}^{k-1} \gamma^i \left((P^\delta)^i r\right)(s,a) + \gamma^k ((P^\delta)^{k-1} P^\pi f)(s,a) && \text{(P.1)} \\
&= \left(\sum_{i=0}^{k-1} \gamma^i (P^\delta)^i r + \gamma^k (P^\delta)^{k-1} P^\pi f\right)(s,a) \\
&= \left(\sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1} (P^\delta)^{k-1} \left(r + \gamma P^\pi f\right)\right)(s,a) && \text{(P.2)} \\
&= \left(\sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1} (P^\delta)^{k-1} T^\pi f\right)(s,a), && \text{(P.3)}
\end{aligned}$$

where line (P.1) follows from Definition 9.2, line (P.2) is obtained by isolating the last term in the summation $\gamma^{k-1}(P^\delta)^{k-1} r$ and collecting $\gamma^{k-1}(P^\delta)^{k-1}$ thanks to the linearity of $(P^\delta)^{k-1}$, and line (P.3) derives from the definition of the Bellman expectation operator T^π. It remains to prove that for $g \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have the following identity:

$$(T^\delta)^{k-1} g = \sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1} (P^\delta)^{k-1} g. \tag{P.4}$$

We prove it by induction on $k \in \mathbb{N}_{\geqslant 1}$. For $k = 1$ we have only $g = (T^\delta)^0 g$. Let us assume that the identity hold for all integers $h < k$, we prove the statement for k:

$$\begin{aligned}
\left((T^\delta)^{k-1} g\right)(s,a) &= \left(\sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1}(P^\delta)^{k-1} g\right)(s,a) \\
&= \left(\sum_{i=0}^{k-3} \gamma^i (P^\delta)^i r + \gamma^{k-2}(P^\delta)^{k-2}(r + \gamma P^\delta g)\right)(s,a) && \text{(P.5)} \\
&= \left(\sum_{i=0}^{k-3} \gamma^i (P^\delta)^i r + \gamma^{k-2}(P^\delta)^{k-2} T^\delta g\right)(s,a) && \text{(P.6)} \\
&= \left((T^\delta)^{k-2} T^\delta g\right)(s,a) = \left((T^\delta)^{k-1} g\right)(s,a), && \text{(P.7)}
\end{aligned}$$

where line (P.5) derives from isolating the last term in the summation and collecting $\gamma^{k-2}(P^\delta)^{k-2}$ thanks to the linearity of $(P^\delta)^{k-2}$, line (P.6) comes from the definition of the Bellman persisted operator T^δ, and finally line (P.7) follows from the inductive hypothesis. We get the result by taking $g = T^\pi f$.

Concerning the k-persistent Bellman optimal operator the derivation is analogous. The Bellman optimal operator becomes: $T^* f = r + \gamma P \mathrm{M}_{\mathcal{A}} f$. Therefore, we have:

$$\begin{aligned}
(T_k^* f)(s,a) &= r_k(s,a) + \gamma^k \int_{\mathcal{S}} P_k(\mathrm{d}s'|s,a) \sup_{a' \in \mathcal{A}} \{f(s',a')\} \\
&= r_k(s,a) + \gamma^k \int_{\mathcal{S}} P_k(\mathrm{d}s'|s,a)(\mathrm{M}_{\mathcal{A}} f)(s') && \text{(P.8)} \\
&= \left(r_k + \gamma^k P_k \mathrm{M}_{\mathcal{A}} f\right)(s,a) && \text{(P.9)} \\
&= \left(\sum_{i=0}^{k-1} \gamma^i (P^\delta)^i r + \gamma^k (P^\delta)^{k-1} P \mathrm{M}_{\mathcal{A}} f\right)(s,a) \\
&= \left(\sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1}(P^\delta)^{k-1}\left(r + \gamma P \mathrm{M}_{\mathcal{A}} f\right)\right)(s,a) && \text{(P.10)} \\
&= \left(\sum_{i=0}^{k-2} \gamma^i (P^\delta)^i r + \gamma^{k-1}(P^\delta)^{k-1} T^* f\right)(s,a), && \text{(P.11)}
\end{aligned}$$

where line (P.8) derives from the definition of the max-operator $\mathrm{M}_{\mathcal{A}}$ and line (P.9) from the definition of the operator P_k. By applying Equation (P.4) we get the result. □

The fixed point equations for the k-persistent Q-functions become:

$$\begin{aligned}
Q_k^\pi &= \left(T^\delta\right)^{k-1} T^\pi Q_k^\pi, \\
Q_k^* &= \left(T^\delta\right)^{k-1} T^* Q_k^*.
\end{aligned}$$

9.3 Bounding the Performance Loss

Learning in the space of k-persistent policies Π_k, means reducing the control opportunities available to the learner. Therefore, increasing k can only lower the performance of the optimal policy, i.e., for every $k, k' \in \mathbb{N}_{\geqslant 1}$ and state $s \in \mathcal{S}$:

$$k \leqslant k' \implies V_k^*(s) \geqslant V_{k'}^*(s).$$

The goal of this section is to quantify the performance loss by deriving a bound on the quantity $\|Q^* - Q^*_k\|_{p,\rho}$ as a function of the persistence $k \in \mathbb{N}_{\geqslant 1}$, where $p \geqslant 1$ and $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ is an evaluation distribution. To this purpose, we first focus on $\|Q^\pi - Q^\pi_k\|_{p,\rho}$ for a fixed policy $\pi \in \Pi^{\mathrm{SR}}$. Then, we show how to employ it to control $\|Q^* - Q^*_k\|_{p,\rho}$.

9.3.1 General Bound on $\|Q^\pi - Q^\pi_k\|_{p,\rho}$

We start presenting the following result that provides an exact expression of the difference between the Q-functions of the same policy $\pi \in \Pi^{\mathrm{SR}}$ when run at persistence 1 and at persistence $k \in \mathbb{N}_{\geqslant 1}$. The auxiliary results are reported in Appendix A.4.

Lemma 9.2 (Persistence Lemma)**.** *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy, then for any $k \in \mathbb{N}_{\geqslant 1}$ the following identity holds:*

$$Q^\pi - Q^\pi_k = \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-(i-1) \bmod k} T^\pi Q^\pi_k.$$

Proof. Let us consider the first identity of Lemma A.17:

$$\begin{aligned}
Q^\pi - Q^\pi_k &= \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k \left(P^\pi\right)^k\right)^{-1} \left(\left(T^\pi\right)^k Q^\pi_k - \left(T^\delta\right)^{k-1} T^\pi Q^\pi_k\right) \\
&= \left(\sum_{j=0}^{\infty} \gamma^{kj} \left(P^\pi\right)^{kj}\right) \left(\left(T^\pi\right)^k Q^\pi_k - \left(T^\delta\right)^{k-1} T^\pi Q^\pi_k\right) && \text{(P.12)} \\
&= \left(\sum_{j=0}^{\infty} \gamma^{kj} \left(P^\pi\right)^{kj}\right) \sum_{l=0}^{k-2} \gamma^{l+1} \left(P^\pi\right)^l \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-l} T^\pi Q^\pi_k && \text{(P.13)} \\
&= \sum_{j=0}^{\infty} \gamma^{kj} \left(P^\pi\right)^{kj} \sum_{l=0}^{k-2} \gamma^{l+1} \left(P^\pi\right)^l \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-l} T^\pi Q^\pi_k \\
&= \sum_{j=0}^{\infty} \sum_{l=0}^{k-2} \gamma^{kj+l+1} \left(P^\pi\right)^{kj+l} \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-l} T^\pi Q^\pi_k,
\end{aligned}$$

where line (P.12) follows from applying the Neumann series at the first factor, line (P.13) is obtained by applying the first identity of Lemma A.18 to the bounded measurable function $T^\pi Q^\pi_k$. The subsequent lines are obtained by straightforward algebraic manipulations. Now we rename the indexes by setting $i = kj + l + 1$. Since $l \in \{0, \dots, k-2\}$ we have that $j = (i-1) \operatorname{div} k$ and $l = (i-1) \bmod k$. Moreover, we observe that i ranges over all non-negative integers values except for the multiples of the persistence k, i.e., $i \in \{n \in \mathbb{N} \,:\, n \bmod k \neq 0\}$. Now, recalling that $i \bmod k \neq 0$, we observe that for the distributive property of the modulo operator we have $(i-1) \bmod k = (i \bmod k - 1 \bmod k) \bmod k = (i \bmod k - 1) \bmod k = i \bmod k - 1$. □

It is worth noting that in Metelli et al. (2020a) another identity was provided in which the roles of P^π and P^δ are switched. From this result, we can derive a bound on the norm of the difference between the Q-functions, as shown in the following result.

Theorem 9.3. *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. Let $\mathcal{Q}_k = \{\left(T^\delta\right)^{k-2-l} T^\pi Q^\pi_k \,:\, l \in \{0, \dots, k-2\}\}$ and for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ let us define:*

$$d^\pi_{\mathcal{Q}_k}(s,a) = \sup_{f \in \mathcal{Q}_k} \left\{ \left| \int_{\mathcal{S}} \int_{\mathcal{A}} \left(P^\pi(\mathrm{d}s', \mathrm{d}a'|s,a) - P^\delta(\mathrm{d}s', \mathrm{d}a'|s,a)\right) f(s',a') \right| \right\}.$$

Then, for any $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$, $p \geqslant 1$, and $k \in \mathbb{N}_{\geqslant 1}$, it holds that:

$$\|Q^\pi - Q_k^\pi\|_{p,\rho} \leqslant \frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)} \left\|d_{\mathcal{Q}_k}^\pi\right\|_{p,\eta_k^{\rho,\pi}},$$

where $\eta_k^{\rho,\pi} \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$ is a probability measure defined for every state $(s,a) \in \mathcal{S} \times \mathcal{A}$ as:

$$\eta_k^{\rho,\pi}(\mathrm{d}s, \mathrm{d}a) = \frac{(1-\gamma)(1-\gamma^k)}{\gamma(1-\gamma^{k-1})} \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(\rho \left(P^\pi\right)^{i-1}\right)(\mathrm{d}s, \mathrm{d}a).$$

Proof. We start from the first equality derived in Lemma 9.2, and we apply the $L_p(\rho)$-norm both sides, with $p \geqslant 1$:

$$\begin{aligned}
\|Q^\pi - Q_k^\pi\|_{p,\rho}^p &= \left\| \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-(i-1) \bmod k} T^\pi Q_k^\pi \right\|_{p,\rho}^p \\
&= \rho \left| \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-(i-1) \bmod k} T^\pi Q_k^\pi \right|^p && \text{(P.14)} \\
&\leqslant \rho \left| \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} \sup_{f \in \mathcal{Q}_k} \left| \left(P^\pi - P^\delta\right) f \right| \right|^p && \text{(P.15)} \\
&= \left(\frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)} \right)^p \rho \left| \frac{(1-\gamma)(1-\gamma^k)}{\gamma(1-\gamma^{k-1})} \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} d_{\mathcal{Q}_k}^\pi \right|^p && \text{(P.16)} \\
&\leqslant \left(\frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)} \right)^p \frac{(1-\gamma)(1-\gamma^k)}{\gamma(1-\gamma^{k-1})} \rho \sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i \left(P^\pi\right)^{i-1} \left|d_{\mathcal{Q}_k}^\pi\right|^p && \text{(P.17)} \\
&= \left(\frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)} \right)^p \eta_k^{\rho,\pi} \left|d_{\mathcal{Q}_k}^\pi\right|^p && \text{(P.18)} \\
&= \left(\frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)} \right)^p \left\|d_{\mathcal{Q}_k}^\pi\right\|_{p,\eta^{\rho,\pi}}^p. && \text{(P.19)}
\end{aligned}$$

where line (P.14) is obtained by the definition of norm, written in the operator form, line (P.15) is obtained by bounding $\left(P^\pi - P^\delta\right)\left(T^\delta\right)^{k-2-(i-1) \bmod k} \leqslant \sup_{f \in \mathcal{Q}_k} \left\{\left|\left(P^\pi - P^\delta\right) f\right|\right\}$, recalling the definition of $\mathcal{Q}_k$ and that $(i-1) \bmod k \leqslant k-2$ for all $i \in \mathbb{N}$ and $i \bmod k \neq 0$. Then, line (P.16) follows from deriving the normalization constant to make the summation $\sum_{i \bmod k \neq 0} \gamma^i \left(P^\pi\right)^{i-1}$ a proper probability distribution. Such a constant can be obtained as follows:

$$\sum_{\substack{i \in \mathbb{N} \\ i \bmod k \neq 0}} \gamma^i = \sum_{i \in \mathbb{N}} \gamma^i - \sum_{i \in \mathbb{N}} \gamma^{ki} = \frac{\gamma(1-\gamma^{k-1})}{(1-\gamma)(1-\gamma^k)}.$$

Line (P.17) is obtained by applying Jensen's inequality recalling that $p \geqslant 1$. Finally, line (P.18) derives from the definition of the distribution $\eta_k^{\rho,\pi}$ and line (P.19) from the definition of $L_p(\eta_k^{\rho,\pi})$-norm. □

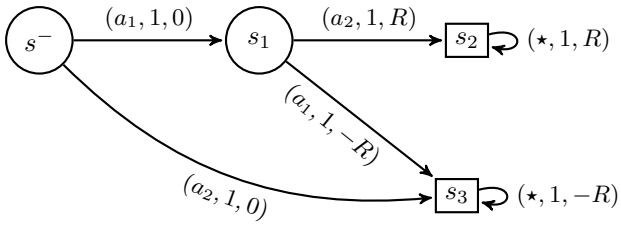


Figure 9.3: *The MDP counter-example of Proposition 9.4, where $R > 0$. Each arrow connecting two states s and s' is labeled with the 3-tuple $(a, p(s'|s,a), r(s,a))$; the symbol $\star$ denotes any action in $\mathcal{A}$. While the optimal policy in the original MDP starting in s^- can avoid negative rewards by executing an action sequence of the kind $(a_1, a_2, \dots)$, every policy in the k-persistent MDP, with $k \in \mathbb{N}_{\geqslant 2}$, inevitably ends in the negative terminal state, as the only possible action sequences are of the kind $(a_1, a_1, \dots)$ and $(a_2, a_2, \dots)$.*

The bound shows that the Q-function difference depends on the discrepancy $d^{\pi}_{\mathcal{Q}_k}$ between the transition-kernel P^{π} and the corresponding persistent version P^{δ}, which is a form of *integral probability metric* (Müller, 1997), defined in terms of the set $\mathcal{Q}_k$. This term is averaged with the distribution $\eta^{\rho,\pi}_k$, which encodes the (discounted) probability of visiting a state-action pair, ignoring the visitations made at decision steps i that are multiple of the persistence k. Indeed, in those steps, we play policy π regardless which persistence is used.[3] The dependence on k is included in the term $\frac{1-\gamma^{k-1}}{1-\gamma^k}$. When $k \to 1$ this term displays a linear growth in k, being asymptotic to $(k-1)\log\frac{1}{\gamma}$, and, clearly, vanishes for $k = 1$. Instead, when $k \to \infty$ this term tends to 1.

We can employ result derived above to obtain a bound on $\|Q^* - Q^*_k\|_{p,\rho}$. Indeed, let $\pi^* \in \Pi^{\mathrm{SR}}$ be an optimal policy of $\mathcal{M}$ and with $\pi^*_k \in \Pi_k$ an optimal policy of $\mathcal{M}_k$, we have that for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$Q^*(s,a) - Q^*_k(s,a) = Q^{\pi^*}(s,a) - Q^{\pi^*_k}_k(s,a) \leqslant Q^{\pi^*}(s,a) - Q^{\pi^*}_k(s,a),$$

since $Q^{\pi^*_k}_k(s,a) \geqslant Q^{\pi^*}_k(s,a)$. Thus, we have that $\|Q^* - Q^*_k\|_{p,\rho} \leqslant \left\|Q^{\pi^*} - Q^{\pi^*}_k\right\|_{p,\rho}$.

9.3.2 Performance Loss without Regularity

The general bound we derived in the previous section applies for every MDP, without further assumptions. However, it is defined in terms of the dissimilarity index $d^{\pi}_{\mathcal{Q}_k}$, whose value can become sufficiently large to make the bound vacuous. This circumstance is clarified in the following negative result.

Proposition 9.4. *For any MDP $\mathcal{M}$ and $k \in \mathbb{N}_{\geqslant 2}$ it holds that for every state $s \in \mathcal{S}$:*

$$V^*_k(s) \geqslant V^*(s) - \frac{2\gamma R_{\max}}{1-\gamma}. \tag{9.7}$$

[3] $\eta^{\rho,\pi}_k$ resembles the γ-discounted state-action distribution μ^{π}_{γ} (Sutton et al., 1999a), but ignoring the decision steps multiple of k.

Furthermore, there exists an MDP $\mathcal{M}^-$ (Figure 9.3) and a state $s^- \in \mathcal{S}$ such that the bound holds with equality for all $k \in \mathbb{N}_{\geqslant 2}$.

Proof. First of all, we recall that $V^*(s) - V_k^*(s) \geqslant 0$ since we cannot increase performance when executing a policy with a persistence k. Let π^* an optimal policy on the MDP $\mathcal{M}$, we observe that for all $s \in \mathcal{S}$:

$$V^*(s) - V_k^*(s) \leqslant V^{\pi^*}(s) - V_k^{\pi^*}(s), \tag{P.20}$$

since $V^{\pi^*}(s) = V^*(s)$ and $V_k^*(s) \geqslant V_k^{\pi^*}(s)$. Let us now consider the corresponding Q-functions $Q^{\pi^*}(s,a)$ and $Q_k^{\pi^*}(s,a)$. Recalling that they are the fixed points of the Bellman operators T^{π^*} and $T_k^{\pi^*}$ we have:

$$\begin{aligned}
Q^{\pi^*} - Q_k^{\pi^*} &= T^{\pi^*} Q^{\pi^*} - T_k^{\pi^*} Q_k^{\pi^*} \\
&= r + \gamma P^\pi Q^{\pi^*} - r_k - \gamma^k P_k^\pi Q_k^{\pi^*} \\
&= r + \gamma P^\pi Q^{\pi^*} - \sum_{i=0}^{k-1} \gamma^i \left(P^\delta\right)^i r - \gamma^k P_k^\pi Q_k^{\pi^*} \\
&= \gamma P^\pi Q^{\pi^*} - \sum_{i=1}^{k-1} \gamma^i \left(P^\delta\right)^i r - \gamma^k P_k^\pi Q_k^{\pi^*},
\end{aligned}$$

where we exploited the definitions of the Bellman expectation operators in the k-persistent MDP. As a consequence, we have that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
Q^{\pi^*}(s,a) - Q^{\pi^*}(s,a) &\leqslant \gamma \frac{R_{\max}}{1-\gamma} + R_{\max} \sum_{i=1}^{k-1} \gamma^i + \gamma^k \frac{R_{\max}}{1-\gamma} \\
&= \gamma \frac{R_{\max}}{1-\gamma} + R_{\max} \frac{\gamma(1-\gamma^{k-1})}{1-\gamma} + \gamma^k \frac{R_{\max}}{1-\gamma} = \frac{2\gamma R_{\max}}{1-\gamma},
\end{aligned}$$

where we considered the following facts that hold for all $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
&\left(P^\pi Q^{\pi^*}\right)(s,a) \leqslant \frac{R_{\max}}{1-\gamma}, \\
&\left(\left(P^\delta\right)^i r\right)(s,a) \leqslant R_{\max}, \\
&\left(P_k^\pi Q_k^{\pi^*}\right)(s,a) \leqslant frac R_{\max} 1 - \gamma.
\end{aligned}$$

The result follows by observing that $V^{\pi^*}(s) - V_k^{\pi^*}(s) = \int_{\mathcal{A}} \pi^*(\mathrm{d}a|s) \left(Q^{\pi^*}(s,a) - Q^{\pi^*}(s,a)\right)$.

We now prove that the bound is tight for the MDP of Figure 9.3. From inspection, we observe that the optimal policy must reach the terminal state s_2 yielding the positive reward $R > 0$. Thus the optimal policy plays action a_1 in state s^- and action a_2 in state s_1, generating a value function $V^*(s^-) = \frac{\gamma R}{1-\gamma}$. Let us now consider the 2-persistent MDP $\mathcal{M}_2^-$. Whichever action is played in state s^- it is going to be persisted for the subsequent decision epoch and, consequently, we will end up in state s_3, yielding the negative reward $-R < 0$. Thus, the optimal value function will be $V_2^*(s^-) = -\frac{\gamma R}{1-\gamma}$. Clearly, the same rationale holds for any persistence $k \in \mathbb{N}_{\geqslant 3}$. □

The quantity $\frac{2\gamma R_{\max}}{1-\gamma}$ is the maximum performance that we can lose if we play the same action at decision epoch $t = 0$ and then we follow an arbitrary policy thereafter.

9.3.3 Regularity Conditions

We have shown that, if no structure on the MDP and/or on the policy is enforced, the dissimilarity term $d^{\pi}_{\mathcal{Q}_k}$ may become large enough to make the bound vacuous, i.e., larger than $\frac{\gamma R_{\max}}{1-\gamma}$, even for $k = 2$. Intuitively, since action persistence will execute old actions in new states, we need to guarantee that the environment state changes slowly w.r.t. to time and the policy must play similar actions in similar states. This means that if an action is good in a state, it will also be almost good for states encountered in the near future. In order to proceed, we need to introduce some basic notions of Lipschitz MDPs (Rachelson and Lagoudakis, 2010; Pirotta et al., 2015). Although the condition on the policy is directly enforced under Lipschitz conditions, we need a new notion of regularity over time for the MDP.

Lipschitz MDPs Let $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$ be two metric spaces, a function $f : \mathcal{X} \to \mathcal{Y}$ is called L_f-Lipschitz continuous (L_f-LC), where $L_f \geqslant 0$, if for all $x, x' \in \mathcal{X}$ we have:

$$d_{\mathcal{Y}}(f(x), f(x')) \leqslant L_f d_{\mathcal{X}}(x, x').$$

Moreover, we define the Lipschitz semi-norm as:

$$\|f\|_L = \sup_{x,x' \in \mathcal{X}: x \neq x'} \left\{ \frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x, x')} \right\}.$$

For real functions we employ Euclidean distance $d_{\mathcal{Y}}(y, y') = \|y - y'\|_2$, while for probability distributions we use the Kantorovich (L_1-Wasserstein) metric defined for every pair of probability measures $\mu, \nu \in \mathscr{P}(\mathcal{Z})$ as (Villani, 2008):

$$d_{\mathcal{Y}}(\mu, \nu) = \mathcal{W}_1(\mu, \nu) = \sup_{f: \|f\|_L \leqslant 1} \left| \int_{\mathcal{Z}} f(z)(\mu - \nu)(\mathrm{d}z) \right|. \tag{9.8}$$

We now introduce the notions of Lipschitz MDP and Lipschitz policy that we will employ in the following (Rachelson and Lagoudakis, 2010; Pirotta et al., 2015).

Assumption 9.1 (Lipschitz MDP)**.** *Let $\mathcal{M}$ be an MDP. $\mathcal{M}$ is called (L_P, L_r)-LC if for every $(s, a), (\overline{s}, \overline{a}) \in \mathcal{S} \times \mathcal{A}$:*

$$\begin{aligned} &\mathcal{W}_1\left(P(\cdot|s, a), P(\cdot|\overline{s}, \overline{a})\right) \leqslant L_P \, d_{\mathcal{S}\times\mathcal{A}}\left((s, a), (\overline{s}, \overline{a})\right), \\ &|r(s, a) - r(\overline{s}, \overline{a})| \leqslant L_r \, d_{\mathcal{S}\times\mathcal{A}}\left((s, a), (\overline{s}, \overline{a})\right). \end{aligned}$$

Assumption 9.2 (Lipschitz Policy)**.** *Let $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. π is called L_π-LC if for every $s, \overline{s} \in \mathcal{S}$:*

$$\mathcal{W}_1\left(\pi(\cdot|s), \pi(\cdot|\overline{s})\right) \leqslant L_\pi \, d_{\mathcal{S}}\left(s, \overline{s}\right).$$

Time-Lipschitz MDP We now introduce a novel regularity condition for the MDP that will turn out essential to complete the analysis of the performance loss due to action persistence.

Assumption 9.3. *Let $\mathcal{M}$ be an MDP. $\mathcal{M}$ is L_T-*Time-Lipschitz Continuous *(L_T-TLC) if for every $(s,a) \in \mathcal{S} \times \mathcal{A}$:*

$$\mathcal{W}_1\left(P(\cdot|s,a), \delta_s\right) \leqslant L_T. \tag{9.9}$$

This assumption requires that the Kantorovich distance between the distribution of the next state s' and the deterministic distribution centered in the current state s is bounded by L_T, i.e., the system does not evolve "too fast".

Remark 9.3. *We draw a connection between the rate at which a dynamical system evolves and the L_T constant of Assumption 9.3. Consider a continuous-time dynamical system having $\mathcal{S} = \mathbb{R}^{d_\mathcal{S}}$ and $\mathcal{A} = \mathbb{R}^{d_\mathcal{A}}$ governed by the law $\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{a}(t))$ such that $\sup_{\mathbf{s}\in\mathcal{S},\mathbf{a}\in\mathcal{A}} \{\|\mathbf{f}(\mathbf{s},\mathbf{a})\|\} \leqslant F < \infty$. Suppose to control the system with a discrete-time step $\Delta t_0 > 0$, inducing an MDP with transition model $P_{\Delta t_0}$. Using the norm $\|\cdot\|$, Assumption 9.3 becomes:*

$$\begin{aligned}\mathcal{W}_1\left(P_{\Delta t_0}(\cdot|\mathbf{s},\mathbf{a}), \delta_\mathbf{s}\right) &= \|\mathbf{s}(t+\Delta t_0) - \mathbf{s}(t)\| \\ &= \left\| \int_t^{t+\Delta t_0} \dot{\mathbf{s}}(\mathrm{d}t) \right\| \leqslant F\Delta t_0.\end{aligned}$$

Thus, the Time Lipschitz constant L_T depends on: i) how fast the dynamical system evolves (F); ii) the duration of the control time step (Δt_0).

Bound We are now ready to bound the dissimilarity term $d^\pi_{\mathcal{Q}_k}$ under the regularity assumptions introduced above.

Theorem 9.5. *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. Under Assumptions 9.1, 9.2, and 9.3, if $\gamma \max\{L_P + 1, L_P(1+L_\pi)\} < 1$ and if $\rho(\mathrm{d}s, \mathrm{d}a) = \rho_\mathcal{S}(\mathrm{d}s)\pi(\mathrm{d}a|s)$ for every $(s,a) \in \mathcal{S}\times\mathcal{A}$ with $\rho_\mathcal{S} \in \mathscr{P}(\mathcal{S})$, then for any $k \in \mathbb{N}_{\geqslant 1}$:*

$$\left\| d^\pi_{\mathcal{Q}_k} \right\|_{p,\eta^{\rho,\pi}_k} \leqslant L_{\mathcal{Q}_k}\left[(L_\pi + 1)L_T + \sigma_p\right].$$

where:

$$\begin{aligned}L_{\mathcal{Q}_k} &= \frac{L_r}{1 - \gamma\max\{L_P + 1, L_P(1+L_\pi)\}}, \\ \sigma_p^p &= \sup_{s\in\mathcal{S}} \int_\mathcal{A}\int_\mathcal{A} d_\mathcal{A}\left(a,a'\right)^p \pi(\mathrm{d}a|s)\pi(\mathrm{d}a'|s).\end{aligned}$$

Proof. Let us now consider the dissimilarity term in norm:

$$\begin{aligned}\|d^\pi_{\mathcal{Q}_k}\|^p_{p,\eta^{\rho,\pi}_k} &= \int_\mathcal{S}\int_\mathcal{A} \eta^{\rho,\pi}_k(\mathrm{d}s,\mathrm{d}a) \left| \sup_{f\in\mathcal{Q}_k} \left| \int_\mathcal{S}\int_\mathcal{A} \left(P^\pi(\mathrm{d}s',\mathrm{d}a'|s,a) - P^\delta(\mathrm{d}s',\mathrm{d}a'|s,a)\right) f(s',a') \right| \right|^p \\ &\leqslant L^p_{\mathcal{Q}_k} \int_\mathcal{S}\int_\mathcal{A} \eta^{\rho,\pi}_k(\mathrm{d}s,\mathrm{d}a) \\ &\quad \times \left| \sup_{f:\|f\|_L\leqslant 1} \left| \int_\mathcal{S}\int_\mathcal{A} \left(P^\pi(\mathrm{d}s',\mathrm{d}a'|s,a) - P^\delta(\mathrm{d}s',\mathrm{d}a'|s,a)\right) f(s',a') \right| \right|^p,\end{aligned}$$

where the inequality follows from Lemma A.20. We now consider the inner term and perform the following algebraic manipulations:

$$
\begin{aligned}
&\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\int_{\mathcal{A}}\Big(P^{\pi}(\mathrm{d}s',\mathrm{d}a'|s,a)-P^{\delta}(\mathrm{d}s',\mathrm{d}a'|s,a)\Big)\,f(s',a')\right| \\
&\quad=\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\int_{\mathcal{A}}P(\mathrm{d}s'|s,a)\pi(\mathrm{d}a'|s')f(s',a')-\int_{\mathcal{S}}\int_{\mathcal{A}}P(\mathrm{d}s'|s,a)\delta_a(\mathrm{d}a')f(s',a')\right. \\
&\qquad\left.\pm\int_{\mathcal{S}}\int_{\mathcal{A}}\delta_s(\mathrm{d}s')\pi(\mathrm{d}a'|s')f(s',a')\pm\int_{\mathcal{S}}\int_{\mathcal{A}}\delta_s(\mathrm{d}s')\delta_a(\mathrm{d}a')f(s',a')\right| \\
&\quad\leqslant\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\big(P(\mathrm{d}s'|s,a)-\delta_s(\mathrm{d}s')\big)\int_{\mathcal{A}}\pi(\mathrm{d}a'|s')f(s',a')\right| \\
&\qquad+\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\big(P(\mathrm{d}s'|s,a)-\delta_s(\mathrm{d}s')\big)\int_{\mathcal{A}}\delta_a(\mathrm{d}a')f(s',a')\right| \\
&\qquad+\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\delta_s(\mathrm{d}s')\int_{\mathcal{A}}\big(\pi(\mathrm{d}a'|s')-\delta_a(\mathrm{d}a')\big)\,f(s',a')\right|.
\end{aligned}
$$

We now consider the first two terms:

$$
\begin{aligned}
&\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\big(P(\mathrm{d}s'|s,a)-\delta_s(\mathrm{d}s')\big)\int_{\mathcal{A}}\pi(\mathrm{d}a'|s')f(s',a')\right| \\
&\qquad+\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\big(P(\mathrm{d}s'|s,a)-\delta_s(\mathrm{d}s')\big)\int_{\mathcal{A}}\delta_a(\mathrm{d}a')f(s',a')\right| \\
&\quad\leqslant(L_\pi+1)\mathcal{W}_1\left(P(\cdot|s,a),\delta_s\right) \qquad \text{(P.21)}\\
&\quad\leqslant(L_\pi+1)L_T,
\end{aligned}
$$

where line (P.21) follows from observing that the function $g_f(s')=\int_{\mathcal{A}}\pi(\mathrm{d}a'|s')f(s',a')$ is L_π-LC, and function $h_f(s')=\int_{\mathcal{A}}\delta_a(\mathrm{d}a')f(s',a')=f(s',a)$ is 1-LC. Moreover, under Assumption 9.3, we have that $\mathcal{W}_1\left(P(\cdot|s,a),\delta_s\right)\leqslant L_T$. Let us now focus on the third term:

$$
\begin{aligned}
&\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{S}}\delta_s(\mathrm{d}s')\int_{\mathcal{A}}\big(\pi(\mathrm{d}a'|s')-\delta_a(\mathrm{d}a')\big)\,f(s',a')\right| \\
&\quad=\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}}\big(\pi(\mathrm{d}a'|s)-\delta_a(\mathrm{d}a')\big)\,f(s,a')\right| \\
&\quad=\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}}\big(\pi(\mathrm{d}a'|s)-\delta_a(\mathrm{d}a')\big)\,f(a')\right| \qquad \text{(P.22)}\\
&\quad=\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}}\left(\int_{\mathcal{A}}\pi(\mathrm{d}a''|s)\delta_{a'}(\mathrm{d}a'')-\delta_a(\mathrm{d}a')\right)f(a')\right| \qquad \text{(P.23)}\\
&\quad=\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}}\pi(\mathrm{d}a''|s)\int_{\mathcal{A}}\big(\delta_{a''}(\mathrm{d}a')-\delta_a(\mathrm{d}a')\big)\,f(a')\right| \qquad \text{(P.24)}\\
&\quad\leqslant\int_{\mathcal{A}}\pi(\mathrm{d}a''|s)\sup_{f:\|f\|_L\leqslant 1}\left|\int_{\mathcal{A}}\big(\delta_{a''}(\mathrm{d}a')-\delta_a(\mathrm{d}a')\big)\,f(a')\right| \qquad \text{(P.25)}\\
&\quad=\int_{\mathcal{A}}\pi(\mathrm{d}a''|s)d_{\mathcal{A}}(a,a''), \qquad \text{(P.26)}
\end{aligned}
$$

where line (P.22) follows from observing that the dependence on s for function f can be neglected because of the supremum, line (P.23) is obtained from the equality $\pi(\mathrm{d}a'|s)=\int_{\mathcal{A}}\pi(\mathrm{d}a''|s)\delta_{a'}(\mathrm{d}a'')$,

line (P.24) derives from moving the integral over a'' outside and recalling that $\delta_{a''}(\mathrm{d}a') = \delta_{a'}(\mathrm{d}a'')$, line (P.25) comes from Jensen's inequality. Finally, line (P.26) is obtained from the definition of Kantorovich distance between Dirac deltas. Now, we take the expectation w.r.t. $\eta_k^{\rho,\pi}$. Recalling that $\rho(\mathrm{d}s, \mathrm{d}a) = \rho_{\mathcal{S}}(\mathrm{d}s)\pi(\mathrm{d}a|s)$ it follows that the same decomposition holds for $\eta_k^{\rho,\pi}(\mathrm{d}s, \mathrm{d}a) = \eta_{k,\mathcal{S}}^{\rho,\pi}(\mathrm{d}s)\pi(\mathrm{d}a|s)$. Consequently, exploiting the above equation, we have:

$$\begin{aligned}
&\int_{\mathcal{S}} \eta_{k,\mathcal{S}}^{\rho,\pi}(\mathrm{d}s) \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \left| \int_{\mathcal{A}} \pi(\mathrm{d}a''|s) d_{\mathcal{A}}(a, a'') \right|^p \\
&\qquad \leqslant \int_{\mathcal{S}} (\eta_k^{\rho,\pi})_{\mathcal{S}}(\mathrm{d}s) \int_{\mathcal{A}} \pi(\mathrm{d}a|s) \int_{\mathcal{A}} \pi(\mathrm{d}a''|s) d_{\mathcal{A}}(a, a'')^p \\
&\qquad \leqslant \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{A}} \pi(\mathrm{d}a|s)\pi(\mathrm{d}a''|s) d_{\mathcal{A}}(a, a'')^p = \sigma_p^p,
\end{aligned}$$

where the first inequality follows from an application of Jensen's inequality. An application of Minkowski's inequality on the norm $\left\| d_{\mathcal{Q}_k}^{\pi} \right\|_{p, \eta_k^{\rho,\pi}}$ concludes the proof. □

Thus, the dissimilarity $d_{\mathcal{Q}_k}^{\pi}$ between P^{π} and P^{δ} can be bounded with four terms:

i. $L_{\mathcal{Q}_k}$ is (an upper-bound of) the Lipschitz constant of the functions in the set $\mathcal{Q}_k$. Indeed, under Assumptions 9.1 and 9.2 we can reduce the dissimilarity term to the Kantorivich distance (Lemma A.20) for every $(s, a) \in \mathcal{S} \times \mathcal{A}$:

$$d_{\mathcal{Q}_k}^{\pi}(s, a) \leqslant L_{\mathcal{Q}_k} \mathcal{W}_1 \left(P^{\pi}(\cdot|s, a), P^{\delta}(\cdot|s, a) \right).$$

ii. $(L_{\pi} + 1)$ accounts for the Lipschitz continuity of the policy, i.e., policies that prescribe similar actions in similar states have a small value of this quantity.

iii. L_T represents the speed at which the environment state evolves over time.

iv. σ_p denotes the average distance (in L_p-norm) between two actions prescribed by the policy in the same state. This term is zero for deterministic policies and can be related to the maximum policy variance as shown in the following result.

Lemma 9.6. *If $\mathcal{A} = \mathbb{R}^{d_{\mathcal{A}}}$, and $d_{\mathcal{A}}(\mathbf{a}, \mathbf{a}') = \|\mathbf{a} - \mathbf{a}'\|_2$, then it holds that:*

$$\sigma_2^2 \leqslant 2 \sup_{s \in \mathcal{S}} \left\{ \operatorname*{\mathbb{V}ar}_{A \sim \pi(\cdot|s)}[A] \right\}.$$

Proof. Let $s \in \mathcal{S}$ and define the mean-action in state s as:

$$\bar{\mathbf{a}}(s) = \int_{\mathcal{A}} \mathbf{a} \pi(\mathrm{d}\mathbf{a}|s).$$

Thus, we have:

$$\begin{aligned}
\sigma_2^2 &= \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{A}} \|\mathbf{a} - \mathbf{a}'\|_2^2 \pi(\mathrm{d}\mathbf{a}|s)\pi(\mathrm{d}\mathbf{a}'|s) \\
&= \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{A}} \|\mathbf{a} - \mathbf{a}' \pm \bar{\mathbf{a}}(s)\|_2^2 \pi(\mathrm{d}\mathbf{a}|s)\pi(\mathrm{d}\mathbf{a}'|s) \\
&\leqslant \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{A}} \|\mathbf{a} - \bar{\mathbf{a}}(s)\|_2^2 \pi(\mathrm{d}\mathbf{a}|s)\pi(\mathrm{d}\mathbf{a}'|s) + \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{A}} \|\mathbf{a}' - \bar{\mathbf{a}}(s)\|_2^2 \pi(\mathrm{d}\mathbf{a}|s)\pi(\mathrm{d}\mathbf{a}'|s)
\end{aligned}$$

$$= \sup_{s\in\mathcal{S}} \int_{\mathcal{A}} \|\mathbf{a} - \overline{\mathbf{a}}(s)\|_2^2 \, \pi(\mathrm{d}\mathbf{a}|s) + \sup_{s\in\mathcal{S}} \int_{\mathcal{A}} \|\mathbf{a}' - \overline{\mathbf{a}}(s)\|_2^2 \, \pi(\mathrm{d}\mathbf{a}'|s)$$
$$= 2\sup_{s\in\mathcal{S}} \int_{\mathcal{A}} \|\mathbf{a} - \overline{\mathbf{a}}(s)\|_2^2 \, \pi(\mathrm{d}\mathbf{a}|s) = 2\sup_{s\in\mathcal{S}} \left\{ \mathop{\mathbb{V}\mathrm{ar}}_{A\sim\pi(\cdot|s)}[A] \right\}.$$

□

A more detailed discussion on the conditions requested in Theorem 9.5, with particular reference to dynamical systems, is reported in Appendix B.4 of (Metelli et al., 2020a).

9.4 Persistent Fitted Q-Iteration

In this section, we introduce an extension of Fitted Q-Iteration (FQI, Ernst et al., 2005) that employs the notion of persistence.[4] We have introduced the class of AVI algorithms in Section 3.2. *Persisted Fitted Q-Iteration* (PFQI) takes as input a *target persistence* $k \in \mathbb{N}_{\geqslant 1}$ and its goal is to approximate the k-persistent optimal action-value function Q_k^*. Starting from an initial estimate $Q^{(0)}$, at each iteration we compute the next estimate $Q^{(j+1)}$ by performing an approximate application of k-persistent Bellman optimal operator to the previous estimate $Q^{(j)}$, i.e., $Q^{(j+1)} \approx T_k^* Q^{(j)}$. In practice, we have two sources of approximation in this process: i) the representation of the Q-function; ii) the estimation of the k-persistent Bellman optimal operator. (i) comes from the necessity of using function space $\mathcal{F} \subset \mathscr{B}(\mathcal{S}\times\mathcal{A})$ to represent $Q^{(j)}$ when dealing with continuous state spaces. (ii) derives from the approximate computation of T_k^* which needs to be estimated from samples.

Clearly, with samples collected in the k-persistent MDP $\mathcal{M}_k$, the process described above reduces to the standard FQI. However, our algorithm needs to be able to estimate Q_k^* for different values of k, using the same dataset of samples collected in the base MDP $\mathcal{M}$ (at persistence 1).[5] For this purpose, we can exploit the decomposition $T_k^* = (T^\delta)^{k-1} T^*$ of Theorem 9.1 to reduce a single application of T_k^* to a sequence of k applications of the 1-persistent operators. Specifically, at each iteration j with $j \bmod k = 0$, given the current estimate $Q^{(j)}$, we need to perform (in this order) a single application of T^* followed by $k-1$ applications of T^δ, leading to the sequence of approximations:

$$Q^{(j+1)} \approx \begin{cases} T^* Q^{(j)} & \text{if } j \bmod k = 0 \\ T^\delta Q^{(j)} & \text{otherwise} \end{cases}. \tag{9.10}$$

To estimate the Bellman operators, we access a dataset $\mathcal{D} = \{(S_i, A_i, S_i', R_i)\}_{i=1}^n$ collected in the base MDP $\mathcal{M}$, where $(S_i, A_i) \sim \nu$, $S_i' \sim P(\cdot|S_i, A_i)$, $R_i \sim R(\cdot|S_i, A_i)$, and $\nu \in \mathscr{P}(\mathcal{S}\times\mathcal{A})$ is a sampling distribution. We employ $\mathcal{D}$ to compute the *empirical Bellman operators* (Farahmand, 2011) defined for $f \in \mathscr{B}(\mathcal{S}\times\mathcal{A})$ and $i \in \{1, \dots, n\}$ as:

$$(\widehat{T}^* f)(S_i, A_i) = R_i + \gamma \max_{a\in\mathcal{A}} \{ f(S_i', a) \},$$
$$(\widehat{T}^\delta f)(S_i, A_i) = R_i + \gamma f(S_i', A_i).$$

[4] From now on, we assume that $|\mathcal{A}| < \infty$.

[5] In real-world cases, we might be unable to interact with the physical system to collect samples for any persistence k of interest.

Algorithm 9.1: Persistent Fitted Q-Iteration PFQI (PFQI).

Input: k persistence, J number of iterations ($J \bmod k = 0$), $Q^{(0)}$ initial action-value function, $\mathcal{F}$ function space, $\mathcal{D} = \{(S_i, A_i, S_i', R_i)\}_{i=1}^n$ batch samples

Output: greedy policy $\pi^{(J)}$

forall $j = 0, \dots, J-1$ **do**

  if $j \bmod k = 0$ **then** — Phase 1

    $Y_i^{(j)} = \widehat{T}^* Q^{(j)}(S_i, A_i), \quad i \in \{1, ..., n\}$

  else

    $Y_i^{(j)} = \widehat{T}^\delta Q^{(j)}(S_i, A_i), \quad i \in \{1, ..., n\}$

  $Q^{(j+1)} \in \arg\min_{f \in \mathcal{F}} \left\{ \left\| f - Y^{(j)} \right\|_{2,\mathcal{D}}^2 \right\}$ — Phase 2

$\pi^{(J)}(s) \in \arg\max_{a \in \mathcal{A}} Q^{(J)}(s, a), \quad \forall s \in \mathcal{S}$ — Phase 3

return $\pi^{(J)}$

We have already shown in Section 3.1 that $\widehat{T}^*$ is unbiased. Clearly, also $\widehat{T}^\delta$ is unbiased conditioned to the current state-action pair (S_i, A_i), i.e., $\mathbb{E}[(\widehat{T}^\delta f)(S_i, A_i) | S_i, A_i] = (T^\delta f)(S_i, A_i)$.

The pseudocode of PFQI is summarized in Algorithm 9.1. At each iteration $j = 0, \dots J-1$, we first compute the target values $Y^{(j)}$ by applying the empirical Bellman operators, $\widehat{T}^*$ or $\widehat{T}^\delta$, on the current estimate $Q^{(j)}$ (Phase 1). Then, we project the target $Y^{(j)}$ onto the function space $\mathcal{F}$ by solving the least-squares problem (Phase 2):

$$Q^{(j+1)} \in \arg\min_{f \in \mathcal{F}} \left\{ \left\| f - Y^{(j)} \right\|_{2,\mathcal{D}}^2 \right\} = \frac{1}{n} \sum_{i=1}^n \left| f(S_i, A_i) - Y_i^{(j)} \right|^2.$$

Finally, we compute the approximation of the optimal policy $\pi^{(J)}$, i.e., the greedy policy w.r.t. $Q^{(J)}$ (Phase 3).

9.4.1 Theoretical Analysis

In this section, we present the computational complexity analysis and the study of the error propagation in PFQI.

Computational Complexity The computational complexity of PFQI decreases monotonically with the persistence k. Whenever applying $\widehat{T}^\delta$, we need a single evaluation of $Q^{(j)}$, while $|\mathcal{A}|$ evaluations are needed for $\widehat{T}^*$ due to the $\max$ over the action space $\mathcal{A}$. The overall complexity of J iterations of PFQI with n samples is given in the following result.

Proposition 9.7. *Assuming that the evaluation of the estimated Q-function in a state action pair has computational complexity $\mathcal{O}(1)$, the computational complexity of J iterations of PFQI run with a dataset $\mathcal{D}$ of n samples, neglecting the cost of the regression, is given by:*

$$\mathcal{O}\left(Jn \left(1 + \frac{|\mathcal{A}| - 1}{k} \right) \right).$$

Proof. Let us consider an iteration $j = 0, \dots, J-1$. If $j \bmod k = 0$, we perform an application of $\widehat{T}^*$ which requires to perform $n|\mathcal{A}|$ evaluations of the next-state value function in order to compute the maximum over the actions. On the contrary, when $j \bmod k \neq 0$, we perform an application of $\widehat{T}^\delta$ which requires just n evaluations, since the next-state value function is evaluated in the persistent action only. By the definition of PFQI, J must be an integer multiple of the persistence k. Recalling that a single evaluation of the approximate Q-function is $\mathcal{O}(1)$, we have that the overall complexity is given by:

$$\begin{aligned}
&\mathcal{O}\left(\sum_{j\in\{0,\dots,J-1\}\,\wedge\, j \bmod k=0} n|\mathcal{A}| + \sum_{j\in\{0,\dots,J-1\}\,\wedge\, j \bmod k\neq 0} n\right) \\
&\quad = \mathcal{O}\left(\frac{J}{k}n|\mathcal{A}| + \frac{J(k-1)}{k}n\right) \\
&\quad = \mathcal{O}\left(Jn\left(1 + \frac{|\mathcal{A}|-1}{k}\right)\right).
\end{aligned}$$

□

Error Propagation We now study the error propagation in PFQI. Given the sequence of Q-functions estimates $(Q^{(j)})_{j=0}^{J} \subset \mathcal{F}$ produced by PFQI, we define the approximation error at each iteration $j = 0, \dots, J-1$ as:

$$\epsilon^{(j)} = \begin{cases} T^* Q^{(j)} - Q^{(j+1)} & \text{if } j \bmod k = 0 \\ T^\delta Q^{(j)} - Q^{(j+1)} & \text{otherwise} \end{cases}. \tag{9.11}$$

The goal of this analysis is to bound the distance between the k-persistent optimal Q-function Q_k^* and the Q-function $Q_k^{\pi^{(J)}}$ of the greedy policy $\pi^{(J)}$ w.r.t. $Q^{(J)}$, after J iterations of PFQI. Before proving the main result, we need to introduce a variation of the *concentrability* coefficients (Antos et al., 2008; Farahmand, 2011) to account for action persistence.

Definition 9.3 (Persistent Expected Concentrability)**.** *Let $\rho, \nu \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$, $L \in \mathbb{N}_{\geqslant 1}$, and an arbitrary sequence of stationary policies $(\pi^{(l)})_{l=1}^{L}$. Let $k \in \mathbb{N}_{\geqslant 1}$ be the persistence. For any $m_1, m_2, m_3 \in \mathbb{N}_{\geqslant 1}$ and $q \in [1, \infty]$, we define:*

$$c_{\mathrm{VI}_1,k,q,\rho,\nu}(m_1, m_2, m_3; \pi) = \left\| \frac{\mathrm{d}\big(\rho (P_k^\pi)^{m_1} (P_k^{\pi_k^*})^{m_2} (P^\delta)^{m_3}\big)}{\mathrm{d}\nu} \right\|_{\frac{q}{q-1},\nu},$$

$$c_{\mathrm{VI}_2,k,q,\rho,\nu}(m_1, m_2; (\pi^{(l)})_{l=1}^{L}) = \left\| \frac{\mathrm{d}\big(\rho (P_k^{\pi^{(L)}})^{m_1} P_k^{\pi^{(L-1)}} \dots P_k^{\pi^{(1)}} (P^\delta)^{m_2}\big)}{\mathrm{d}\nu} \right\|_{\frac{q}{q-1},\nu}.$$

If $\rho (P_k^\pi)^{m_1} (P_k^{\pi_k^*})^{m_2} (P^\delta)^{m_3}$ (resp. $\rho (P_k^{\pi^{(L)}})^{m_1} P_k^{\pi^{(L-1)}} \dots P_k^{\pi^{(1)}} (P^\delta)^{m_2}$) is not absolutely continuous w.r.t. to ν, then we convene $c_{\mathrm{VI}_1,\rho,\nu}(m_1, m_2, m_3; \pi, k) = \infty$ (resp. $c_{\mathrm{VI}_2,\rho,\nu}(m_1, m_2; (\pi^{(l)})_{l=1}^{L}, k) = \infty$).

This definition is a generalization of that provided in Farahmand (2011), that can be recovered by setting $k = 1$, $q = 2$, and $m_3 = 0$ for the first coefficient and $m_2 = 0$ for the second coefficient. The following result extends Theorem 3.4 of Farahmand (2011) to account for action persistence.

Theorem 9.8 (Error Propagation for PFQI)**.** *Let $p \geqslant 1$, $k \in \mathbb{N}_{\geqslant 1}$, $J \in \mathbb{N}_{\geqslant 1}$ with $J \bmod k = 0$ and $\rho \in \mathscr{P}(\mathcal{S} \times \mathcal{A})$. Then for any sequence $(Q^{(j)})_{j=0}^{J} \subset \mathcal{F}$ uniformly bounded by $Q_{\max} \leqslant \frac{R_{\max}}{1-\gamma}$, the corresponding $(\epsilon^{(j)})_{j=0}^{J-1}$ defined in Equation* (9.11) *and for any $r \in [0,1]$ and $q \in [1,\infty]$ it holds that:*

$$\left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho} \leqslant \frac{2\gamma^k}{(1-\gamma)(1-\gamma^k)}\left[\frac{2}{1-\gamma}\gamma^{\frac{J}{p}} R_{\max} + C_{\mathrm{VI},\rho,\nu}^{\frac{1}{2p}}(J,r,q)\mathcal{E}^{\frac{1}{2p}}(\epsilon^{(0)},\dots,\epsilon^{(J-1)};r,q)\right],$$

where:

$$C_{\mathrm{VI},\rho,\nu}(J;r,q) = \left(\frac{1-\gamma^k}{2}\right)^2 \sup_{\pi_0,\dots,\pi_J \in \Pi^{\mathrm{SR}}} \left\{\sum_{j=0}^{J-1} \alpha_j^{\frac{2(1-r)}{s-1}} \left(\sum_{m=0}^{\infty} \gamma^{km} \times \left(c_{\mathrm{VI}_1,k,q,\rho,\nu}\left(m, \frac{J}{k} - j \operatorname{div} k, k - j \bmod k - 1; \pi_J\right) + c_{\mathrm{VI}_2,k,q,\rho,\nu}\left(m+1, k - j \bmod k - 1; (\pi_l)_{l=1}^{j \operatorname{div} k}\right)\right)\right)^2\right\},$$

$$\mathcal{E}(\epsilon^{(0)},\dots,\epsilon^{(J-1)};r,q) = \sum_{j=0}^{J-1} \alpha_j^{2r} \left\|\epsilon^{(j)}\right\|_{pq,\nu}^{2p},$$

and $\alpha_j = \begin{cases} \frac{(1-\gamma)\gamma^{J-j-1}}{1-\gamma^{J+1}} & \text{if } 0 \leqslant j < J \\ \frac{(1-\gamma)\gamma^J}{1-\gamma^{J+1}} & \text{if } j = J \end{cases}$.

Proof. The proof follows most of the steps of Theorem 3.4 of Farahmand (2011). We start by deriving a bound relating $Q^* - Q^{(J)}$ to $(\epsilon^{(j)})_{j=0}^{J-1}$. To this purpose, let us first define the cumulative error over k iterations for every $j \bmod k = 0$:

$$\epsilon_k^{(j)} = T_k^* Q^{(j)} - Q^{(j+k)}. \tag{P.27}$$

Let us denote with π_k^* one of the optimal policies of the k-persistent MDP $\mathcal{M}_k$. We have:

$$\begin{aligned} Q_k^* - Q^{(j+k)} &= T_k^{\pi_k^*} Q_k^* - T_k^{\pi_k^*} Q^{(j)} + T_k^{\pi_k^*} Q^{(j)} - T_k^* Q^{(j)} + \epsilon_k^{(j)} \\ &\leqslant \gamma^k P_k^{\pi_k^*}(Q_k^* - Q^{(j)}) + \epsilon_k^{(j)}, \\ Q_k^* - Q^{(j+k)} &= T_k^* Q_k^* - T_k^{\pi^{(j)}} Q^* + T_k^{\pi^{(j)}} Q^* - T_k^* Q^{(j)} + \epsilon_k^{(j)} \\ &\geqslant \gamma^k P_k^{\pi^{(j)}}(Q_k^* - Q^{(j)}) + \epsilon_k^{(j)}, \end{aligned}$$

where we exploited the fact that $T_k^* Q^{(j)} \geqslant T_k^{\pi_k^*} Q^{(j)}$, the definition of greedy policy $\pi^{(j)}$ that implies that $T_k^{\pi^{(j)}} Q^{(j)} = T_k^* Q^{(j)}$ and the definition of $\epsilon_k^{(j)}$. By unrolling the expression derived

above, we have that for every $J \bmod k = 0$:

$$\begin{aligned} Q_k^* - Q^{(J)} &\leqslant \sum_{h=0}^{\frac{J}{k}-1} \gamma^{J-k(h+1)} \left(P_k^{\pi_k^*}\right)^{\frac{J}{k}-h-1} \epsilon_k^{(j)} + \gamma^J \left(P_k^{\pi_k^*}\right)^{\frac{J}{k}} (Q_k^* - Q^{(0)}) \\ Q_k^* - Q^{(J)} &\geqslant \sum_{h=0}^{\frac{J}{k}-1} \gamma^{J-k(h+1)} \left(P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(k(h+1))}}\right) \epsilon_k^{(j)} \\ &\quad + \gamma^J \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}}\right) (Q_k^* - Q^{(0)}). \end{aligned} \tag{P.28}$$

We now provide the following bound relating the difference $Q_k^* - Q_k^{\pi^{(J)}}$ to the difference $Q_k^* - Q^{(J)}$:

$$\begin{aligned} Q_k^* - Q_k^{\pi^{(J)}} &= T_k^{\pi_k^*} Q_k^* - T_k^{\pi_k^*} Q^{(J)} + T_k^{\pi_k^*} Q^{(J)} - T_k^* Q^{(J)} + T_k^* Q^{(J)} - T_k^{\pi^{(J)}} Q_k^{\pi^{(J)}} \\ &\leqslant T_k^{\pi_k^*} Q_k^* - T_k^{\pi_k^*} Q^{(J)} + T_k^* Q^{(J)} - T_k^{\pi^{(J)}} Q_k^{\pi^{(J)}} \\ &= \gamma^k P_k^{\pi_k^*} (Q^* - Q^{(J)}) + \gamma^k P_k^{\pi^{(J)}} (Q^{(J)} - Q_k^{\pi^{(J)}}) \\ &= \gamma^k P_k^{\pi_k^*} (Q^* - Q^{(J)}) + \gamma^k P_k^{\pi^{(J)}} (Q^{(J)} - Q_k^* + Q_k^* - Q_k^{\pi^{(J)}}), \end{aligned}$$

where we exploited the fact that $T_k^* Q^{(J)} \geqslant T_k^{\pi_k^*} Q^{(J)}$ and observed that $T_k^* Q^{(J)} = T_k^{\pi^{(J)}} Q^{(J)}$. By using Lemma 4.2 of Munos (2007) we can derive:

$$Q_k^* - Q_k^{\pi^{(J)}} \leqslant \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}}\right)^{-1} \left(P_k^{\pi_k^*} - P_k^{\pi^{(J)}}\right) (Q^* - Q^{(J)}). \tag{P.29}$$

By plugging Equation (P.28) into Equation (P.29):

$$\begin{aligned} Q_k^* - Q_k^{\pi^{(J)}} &\leqslant \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}}\right)^{-1} \\ &\quad \times \Bigg[\sum_{h=0}^{\frac{J}{k}-1} \gamma^{J-k(h+1)} \left(\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}-h} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(k(h+1))}}\right) \right) \epsilon_k^{(j)} \\ &\quad + \gamma^J \left(\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}+1} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}}\right) \right) (Q_k^* - Q^{(0)}) \Bigg]. \end{aligned} \tag{P.30}$$

Before proceeding, we need to relate the cumulative errors $\epsilon_k^{(j)}$ to the single-step errors $\epsilon^{(j)}$:

$$\begin{aligned} \epsilon_k^{(j)} &= T_k^* Q^{(j)} - Q^{(j+k)} \\ &= (T^\delta)^{k-1} T^* Q^{(j)} - (T^\delta)^{k-1} Q^{(j+1)} + (T^\delta)^{k-1} Q^{(j+1)} - Q^{(j+k)} \\ &= \gamma^{k-1} (P^\delta)^{k-1} \left(T^* Q^{(j)} - Q^{(j+1)}\right) + (T^\delta)^{k-1} Q^{(j+1)} - Q^{(j+k)} \\ &= \gamma^{k-1} (P^\delta)^{k-1} \epsilon^{(j)} + (T^\delta)^{k-1} Q^{(j+1)} - Q^{(j+k)}. \end{aligned}$$

Let us now consider the remaining term $(T^\delta)^{k-1} Q^{(j+1)} - Q^{(j+k)}$:

$$\begin{aligned} (T^\delta)^{k-1} Q^{(j+1)} - Q^{(j+k)} &= (T^\delta)^{k-1} Q^{(j+1)} - (T^\delta)^{k-2} Q^{(j+2)} + (T^\delta)^{k-2} Q^{(j+2)} - Q^{(j+k)} \\ &= \gamma^{k-2} (P^\delta)^{k-2} \left(T^\delta Q^{(j+1)} - Q^{(j+2)}\right) + (T^\delta)^{k-2} Q^{(j+2)} - Q^{(j+k)} \\ &= \gamma^{k-2} (P^\delta)^{k-2} \epsilon^{(j+1)} + (T^\delta)^{k-2} Q^{(j+2)} - Q^{(j+k)} \end{aligned}$$

$$= \sum_{l=2}^{k} \gamma^{k-l} (P^{\delta})^{k-l} \epsilon^{(j+l-1)},$$

where the last step is obtained by unrolling the recursion. Putting all together, we get:

$$\epsilon_k^{(j)} = \sum_{l=1}^{k} \gamma^{k-l} (P^{\delta})^{k-l} \epsilon^{(j+l-1)}. \tag{P.31}$$

Consequently, we can rewrite Equation (P.30) as follows:

$$\begin{aligned}
Q_k^* - Q_k^{\pi^{(J)}} &\leqslant \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}} \right)^{-1} \\
&\quad \times \left[\sum_{h=0}^{\frac{J}{k}-1} \gamma^{J-k(h+1)} \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}-h} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(k(h+1))}} \right) \right) \right. \\
&\qquad \times \sum_{l=1}^{k} \gamma^{k-l} (P^{\delta})^{k-l} \epsilon^{(j+l-1)} \qquad \text{(P.32)} \\
&\quad \left. + \gamma^J \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}+1} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}} \right) \right) (Q_k^* - Q^{(0)}) \right] \\
&= \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}} \right)^{-1} \\
&\quad \times \left[\sum_{h=0}^{\frac{J}{k}-1} \sum_{l=1}^{k} \gamma^{J-kh-l} \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}-h} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(k(h+1))}} \right) \right) \right. \qquad \text{(P.33)} \\
&\qquad \times (P^{\delta})^{k-l} \epsilon^{(j+l-1)} \\
&\quad \left. + \gamma^J \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}+1} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}} \right) \right) (Q_k^* - Q^{(0)}) \right] \\
&= \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}} \right)^{-1} \\
&\quad \times \left[\sum_{j=0}^{J-1} \gamma^{J-j-1} \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}-j \text{ div } k} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(J-k(j \text{ div } k+1))}} \right) \right) \right. \\
&\qquad \times (P^{\delta})^{k-j \bmod k-1} \epsilon^{(j)} \\
&\quad \left. + \gamma^J \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}+1} - \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}} \right) \right) (Q_k^* - Q^{(0)}) \right] \qquad \text{(P.34)} \\
&\leqslant \gamma^k \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}} \right)^{-1} \\
&\quad \times \left[\sum_{j=0}^{J-1} \gamma^{J-j-1} \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}-j \text{ div } k} + \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(J-k(j \text{ div } k+1))}} \right) \right) \right. \\
&\qquad \times (P^{\delta})^{k-j \bmod k-1} \left| \epsilon^{(j)} \right| \\
&\quad \left. + \gamma^J \left(\left(P_k^{\pi_k^*} \right)^{\frac{J}{k}+1} + \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}} \right) \right) \left| Q_k^* - Q^{(0)} \right| \right], \qquad \text{(P.35)}
\end{aligned}$$

where line (P.33) derives from rearranging the two summations, line (P.34) is obtained from a re-definition of the indexes. Specifically, we observed that $h = j \text{ div } k$, $j + 1 = kh + l$, and

$l = j \bmod k + 1$. Finally, line (P.35) is obtained by applying the absolute value to the right hand side and using Jensen's inequality. We now introduce the following terms. If $0 \leqslant j < J$:

$$A_j = \frac{1-\gamma^k}{2}\left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}}\right)^{-1}\left(\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}-j \text{ div } k} + \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(J-k(j \text{ div } k+1))}}\right)\right)(P^\delta)^{k-j \bmod k-1}.$$

Instead, if $j = J$:

$$A_J = \frac{1-\gamma^k}{2}\left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}}\right)^{-1}\left(\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}+1} + \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} \dots P_k^{\pi^{(k)}}\right)\right).$$

Let us recall the definition of α_j as in Farahmand (2011):

$$\alpha_j = \begin{cases} \frac{(1-\gamma)\gamma^{J-j-1}}{1-\gamma^{J+1}} & \text{if } 0 \leqslant j < J \\ \frac{(1-\gamma)\gamma^J}{1-\gamma^{J+1}} & \text{if } j = J \end{cases}. \tag{P.36}$$

Recalling that $\left|Q_k^* - Q^{(0)}\right| \leqslant Q_{\max} + \frac{R_{\max}}{1-\gamma} \leqslant \frac{2R_{\max}}{1-\gamma}$ and applying Jensen's inequality we get to the inequality:

$$Q_k^* - Q_k^{\pi^{(J)}} \leqslant \frac{2\gamma^k(1-\gamma^{J+1})}{(1-\gamma^k)(1-\gamma)}\left[\sum_{j=0}^{J-1}\alpha_j A_j\left|\epsilon^{(j)}\right| + \alpha_J \frac{2R_{\max}}{1-\gamma}\mathbf{1}\right],$$

where $\mathbf{1}$ denotes the constant function on $\mathcal{S}\times\mathcal{A}$ with value 1. Taking the $L_p(\rho)$-norm both sides, recalling that $\sum_{j=1}^{J}\alpha_j = 1$ and that the terms A_j are positive linear operators $A_j : \mathscr{B}(\mathcal{S}\times\mathcal{A}) \to \mathscr{B}(\mathcal{S}\times\mathcal{A})$ such that $A_j\mathbf{1} = \mathbf{1}$. Thus, by Lemma 12 of Antos et al. (2008), we can apply Jensen's inequality twice (once w.r.t. α_j and once w.r.t. A_j), getting:

$$\left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho}^p \leqslant \left(\frac{2\gamma^k(1-\gamma^{J+1})}{(1-\gamma^k)(1-\gamma)}\right)^p \rho\left[\sum_{j=0}^{J-1}\alpha_j A_j\left|\epsilon^{(j)}\right|^p + \alpha_J\left(\frac{2R_{\max}}{1-\gamma}\right)^p\mathbf{1}\right].$$

Consider now the individual terms $\rho A_j\left|\epsilon^{(j)}\right|^p$ for $0 \leqslant j < J$. By the properties of the Neumann series we have:

$$\begin{aligned}
\rho A_j\left|\epsilon^{(j)}\right|^p &= \frac{1-\gamma^k}{2}\rho\left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^{\pi^{(J)}}\right)^{-1} \\
&\quad\times\left(\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}-j \text{ div } k} + \left(P_k^{\pi^{(J)}} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(J-k(j \text{ div } k+1))}}\right)\right) \\
&\quad\times(P^\delta)^{k-j \bmod k-1}\left|\epsilon^{(j)}\right|^p \\
&= \frac{1-\gamma^k}{2}\rho\Bigg[\sum_{m=0}^{\infty}\gamma^{km}\Bigg(\left(P_k^{\pi^{(J)}}\right)^m\left(P_k^{\pi_k^*}\right)^{\frac{J}{k}-j \text{ div } k} \\
&\quad + \left(\left(P_k^{\pi^{(J)}}\right)^{m+1} P_k^{\pi^{(J-k)}} P_k^{\pi^{(J-2k)}} \dots P_k^{\pi^{(J-k(j \text{ div } k))}}\right)\Bigg)\Bigg] \\
&\quad\times(P^\delta)^{k-j \bmod k-1}\left|\epsilon^{(j)}\right|^p.
\end{aligned}$$

We now aim at introducing the concentrability coefficients and for this purpose, we employ the following inequality. For any measurable function $f \in \mathscr{B}(\mathcal{X})$, and the probability measures $\mu_1, \mu_2 \in \mathscr{P}(\mathcal{X})$ such that $\mu_1 \ll \mu_2$, we have the following Hölder's inequality, for any $q \in [1, \infty]$:

$$\int_{\mathcal{X}} f \mathrm{d}\mu_1 \leqslant \left(\int_{\mathcal{X}} \left| \frac{\mathrm{d}\mu_1}{\mathrm{d}\mu_2} \right|^{\frac{q}{q-1}} \mathrm{d}\mu_2 \right)^{\frac{q-1}{q}} \left(\int_{\mathcal{X}} |f|^q \mathrm{d}\mu_2 \right)^{\frac{1}{q}}. \tag{P.37}$$

We now focus on a single term $\rho \left(P_k^{\pi^{(J)}} \right)^m \left(P_k^{\pi_k^*} \right)^{\frac{J}{k} - j \text{ div } k} \left| \epsilon^{(j)} \right|^p$ and we apply the above inequality:

$$\begin{aligned}
&\rho \left(P_k^{\pi^{(J)}} \right)^m \left(P_k^{\pi_k^*} \right)^{\frac{J}{k} - j \text{ div } k} (P^\delta)^{k - j \bmod k - 1} \left| \epsilon^{(j)} \right|^p \leqslant \\
&\qquad \left(\int_{\mathcal{S} \times \mathcal{A}} \left| \frac{\mathrm{d}\rho \left(P_k^{\pi^{(J)}} \right)^m \left(P_k^{\pi_k^*} \right)^{\frac{J}{k} - j \text{ div } k} (P^\delta)^{k - j \bmod k - 1}}{\mathrm{d}\nu} \right|^{\frac{q}{q-1}} \mathrm{d}\nu \right)^{\frac{q-1}{q}} \\
&\qquad \times \left(\int_{\mathcal{S} \times \mathcal{A}} \left| \epsilon^{(j)} \right|^{pq} \mathrm{d}\nu \right)^{\frac{1}{q}} \\
&\qquad = c_{\mathrm{VI}_1, k, q, \rho, \nu} \left(m, \frac{J}{k} - j \text{ div } k, k - j \bmod k - 1; \pi^{(J)} \right) \left\| \epsilon^{(j)} \right\|_{pq, \nu}^p.
\end{aligned}$$

Proceeding in an analogous way for the remaining terms, we get to the expression:

$$\begin{aligned}
\left\| Q_k^* - Q_k^{\pi^{(J)}} \right\|_{p, \rho}^p \leqslant \left(\frac{2\gamma^k (1 - \gamma^{J+1})}{(1 - \gamma^k)(1 - \gamma)} \right)^p \Bigg[& \frac{1 - \gamma^k}{2} \sum_{j=0}^{J-1} \sum_{m=0}^{\infty} \gamma^{km} \\
& \times \left(c_{\mathrm{VI}_1, k, q, \rho, \nu} \left(m, \frac{J}{k} - j \text{ div } k, k - j \bmod k - 1; \pi^{(J)} \right) \right. \\
& \left. + c_{\mathrm{VI}_2, k, q, \rho, \nu} \left(m + 1, k - j \bmod k - 1; (\pi^{(J-lk)})_{l=1}^{j \text{ div } k} \right) \right) \left\| \epsilon^{(j)} \right\|_{pq, \nu}^p \\
& + \alpha_J \left(\frac{2R_{\max}}{1 - \gamma} \right)^p \Bigg].
\end{aligned}$$

To separate the concentrability coefficients and the approximation errors, we apply Hölder's inequality with $s \in [1, \infty]$:

$$\sum_{j=0}^{J} a_j b_j \leqslant \left(\sum_{j=0}^{J} |a_j|^s \right)^{\frac{1}{s}} \left(|b_j|^{\frac{s}{s-1}} \right)^{\frac{s-1}{s}}. \tag{P.38}$$

Let $r \in [0, 1]$, we set:

$$\begin{aligned}
a_j &= \alpha_j^r \left\| \epsilon^{(j)} \right\|_{pq, \nu}^p, \\
b_j &= \alpha_j^{1-r} \frac{1 - \gamma^k}{2} \sum_{j=0}^{J-1} \sum_{m=0}^{\infty} \gamma^{km} \left(c_{\mathrm{VI}_1, k, q, \rho, \nu} \left(m, \frac{J}{k} - j \text{ div } k, k - j \bmod k - 1; \pi^{(J)} \right) \right. \\
&\qquad \left. + c_{\mathrm{VI}_2, k, q, \rho, \nu} \left(m + 1, k - j \bmod k - 1; (\pi^{(J-lk)})_{l=1}^{j \text{ div } k} \right) \right).
\end{aligned}$$

The application of Hölder's inequality leads to:

$$\begin{aligned}\left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho}^p \leqslant \left(\frac{2\gamma^k(1-\gamma^{J+1})}{(1-\gamma^k)(1-\gamma)}\right)^p \frac{1-\gamma^k}{2}\Bigg[\sum_{j=0}^{J-1}\alpha_j^{\frac{s(1-r)}{s-1}}\Bigg(\sum_{m=0}^{\infty}\gamma^{km} \\ \times\Bigg(c_{\mathrm{VI}_1,k,q,\rho,\nu}\left(m,\frac{J}{k}-j \text{ div } k, k-j \bmod k-1;\pi^{(J)}\right) \\ + c_{\mathrm{VI}_2,k,q,\rho,\nu}\left(m+1,k-j \bmod k-1;(\pi^{(J-lk)})_{l=1}^{j \text{ div } k}\right)\Bigg)\Bigg)^{\frac{s}{s-1}}\Bigg]^{\frac{s-1}{s}} \\ \times\left[\sum_{j=0}^{J-1}\alpha_j^{sr}\left\|\epsilon^{(j)}\right\|_{pq,\nu}^{sp}\right]^{\frac{1}{s}} \\ + \left(\frac{2\gamma^k(1-\gamma^{J+1})}{(1-\gamma^k)(1-\gamma)}\right)^p \alpha_J\left(\frac{2R_{\max}}{1-\gamma}\right)^p.\end{aligned}$$

Since the policies $(\pi^{(J-lk)})_{l=1}^{j \text{ div } k}$ are not known, we define the following quantity by taking the supremum over any sequence of policies:

$$\begin{aligned}C_{\mathrm{VI},\rho,\nu}(J;r,s,q) = \left(\frac{1-\gamma^k}{2}\right)^s \sup_{\pi_0,\dots,\pi_J\in\Pi^{\mathrm{SR}}}\Bigg\{\sum_{j=0}^{J-1}\alpha_j^{\frac{s(1-r)}{s-1}}\Bigg(\sum_{m=0}^{\infty}\gamma^{km} \\ \times\Bigg(c_{\mathrm{VI}_1,k,q,\rho,\nu}\left(m,\frac{J}{k}-j \text{ div } k, k-j \bmod k-1;\pi_J\right) \\ + c_{\mathrm{VI}_2,k,q,\rho,\nu}\left(m+1,k-j \bmod k-1;(\pi_l)_{l=1}^{j \text{ div } k}\right)\Bigg)\Bigg)^{\frac{s}{s-1}}\Bigg\}.\end{aligned} \tag{P.39}$$

Moreover, we define the following term that embeds all the terms related to the approximation error:

$$\mathcal{E}(\epsilon^{(0)},\dots,\epsilon^{(J-1)};r,s,q) = \sum_{j=0}^{J-1}\alpha_j^{sr}\left\|\epsilon^{(j)}\right\|_{pq,\nu}^{sp}. \tag{P.40}$$

Observing that $\frac{1-\gamma}{1-\gamma^{J+1}} \leqslant 1$ and $1-\gamma^{J-1} \leqslant 1$, we can put all together and taking the p-th root and recalling that the inequality holds for all $q \in [1,\infty]$, $r \in [0,1]$, and $s \in [1,\infty]$:

$$\begin{aligned}\left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho} \leqslant \frac{2\gamma^k}{(1-\gamma^k)(1-\gamma)}\Bigg[\gamma^{\frac{J}{p}}\frac{2R_{\max}}{1-\gamma} + \\ + \inf_{\substack{q\in[1,\infty]\\ r\in[0,1]\\ s\in[1,\infty]}}\left\{C_{\mathrm{VI},\rho,\nu}(J;r,s,q)^{\frac{s-1}{ps}}\mathcal{E}(\epsilon^{(0)},\dots,\epsilon^{(J-1)};r,s,q)^{\frac{1}{ps}}\right\}\Bigg].\end{aligned}$$

The statement is simplified by taking $s=2$.

□

We immediately observe that for $k=1$ we recover Theorem 3.4 of Farahmand (2011). The term $C_{\mathrm{VI},\rho,\nu}(J;r,q)$, defined in terms of suitable *concentrability coefficients* (Definition 9.3), encodes the distribution shift between the sampling distribution ν and the one induced by the greedy policy sequence $(\pi^{(j)})_{j=0}^J$ encountered along the execution of PFQI. $\mathcal{E}(\cdot;r,q)$ incorporates the approximation errors $(\epsilon^{(j)})_{j=0}^{J-1}$. In principle, it is hard

Algorithm 9.2: Heuristic Persistence Selection.

Input: batch samples $\mathcal{D} = \{\tau_i\}_{i=1}^m$, set of persistences $\mathcal{K}$, set of Q-function $\{Q_k : k \in \mathcal{K}\}$, regressor Reg

Output: approximately optimal persistence $\widetilde{k}$

forall $k \in \mathcal{K}$ **do**

  $\widehat{J}_k^\rho = \frac{1}{m}\sum_{i=1}^m V_k(S_{\tau_i,0})$

  Use the Reg to get an estimate $\widetilde{Q}_k$ of $T_k^* Q_k$

  $\left\|\widetilde{Q}_k - Q_k\right\|_{1,\mathcal{D}} = \frac{1}{\sum_{i=1}^m T(\tau_i)}\sum_{i=1}^m \sum_{t=0}^{T(\tau_i)-1} |\widetilde{Q}_k(S_{\tau_i,t}, A_{\tau_i,t}) - Q_k(S_{\tau_i,t}, A_{\tau_i,t})|$

$\widetilde{k} \in \arg\max_{k\in\mathcal{K}}\{B_k\} = \widehat{J}_k^\rho - \frac{1}{1-\gamma^k}\left\|\widetilde{Q}_k - Q_k\right\|_{1,\mathcal{D}}$

return $\widetilde{k}$

to compare the values of these terms for different persistences k since both the greedy policies and the regression problems are different. Nevertheless, it is worth noting that the multiplicative term $\frac{\gamma^k}{1-\gamma^k}$ decreases in $k \in \mathbb{N}_{\geqslant 1}$. Thus, other things being equal, the bound value decreases when increasing the persistence.

It is worth noting that this analysis and PFQI more in general resembles a particular instance of non-stationary AVI Scherrer and Lesner (2012); Lesner and Scherrer (2015) in which the non-stationary policy is the k-persistent policy instead of the sequence of the last k policies.

Visualizing the Control Frequency Trade-off Thus, the trade-off in the choice of control frequency, which motivates action persistence, can now be stated more formally. We aim at finding the persistence $k \in \mathbb{N}_{\geqslant 1}$ that, for a fixed J, allows learning a policy $\pi^{(J)}$ whose Q-function $Q_k^{\pi^{(J)}}$ is the closest to Q^*. Consider the decomposition obtained via triangular inequality:

$$\left\|Q^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho} \leqslant \left\|Q^* - Q_k^*\right\|_{p,\rho} + \left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho}.$$

The term $\left\|Q^* - Q_k^*\right\|_{p,\rho}$ accounts for the performance degradation due to action persistence: it is algorithm-independent, and it increases in k (Theorem 9.3). Instead, the second term $\left\|Q_k^* - Q_k^{\pi^{(J)}}\right\|_{p,\rho}$ decreases with k and depends on the algorithm (Theorem 9.8). Unfortunately, optimizing their sum is hard since the individual bounds contain terms that are not known in general (e.g., Lipschitz constants, $\epsilon^{(j)}$). The next section proposes heuristics to overcome this problem.

9.5 Persistence Selection

In this section, we discuss how to select a persistence k in a set $\mathcal{K} \subset \mathbb{N}_{\geqslant 1}$ of candidate persistences, when we are given a set of estimated Q-functions: $\{Q_k : k \in \mathcal{K}\}$.[6] Each Q_k induces a greedy policy π_k. Our goal is to find the persistence $k \in \mathcal{K}$ such that π_k has the

[6]For instance, but not necessarily, the Q_k can be obtained by executing PFQI with different persistences $k \in \mathcal{K}$.

maximum expected return in the corresponding k-persistent MDP $\mathcal{M}_k$:

$$k^* \in \arg\max_{k \in \mathcal{K}} \{J_k^{\rho,\pi_k}\}, \tag{9.12}$$

where $\rho \in \mathscr{P}(\mathcal{S})$ is an evaluation distribution and $J_k^{\rho,\pi_k} = \int_{\mathcal{S}} \rho(\mathrm{d}s) V_k^{\pi_k}(s)$ is the expected return of policy π_k executed in the k-persistent MDP $\mathcal{M}_k$.

In principle, we could execute π_k in $\mathcal{M}_k$ to get an estimate of J_k^{ρ,π_k} and employ it to select the persistence k. However, in the batch setting, further interactions with the environment might be not allowed. On the other hand, directly using the estimated Q-function Q_k is inappropriate, since we need to take into account how well Q_k approximates $Q_k^{\pi_k}$. This trade-off is encoded in the following result, which makes use of the *expected Bellman residual*.

Lemma 9.9. *Let $Q \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ and π be a greedy policy w.r.t. Q. Let $J^\rho = \int \rho(\mathrm{d}s) V(s)$, with $V(s) = \max_{a \in \mathcal{A}} \{Q(s,a)\}$ for all $s \in \mathcal{S}$. Then, for any $k \in \mathbb{N}_{\geqslant 1}$, it holds that:*

$$J_k^{\rho,\pi} \geqslant J^\rho - \frac{1}{1-\gamma^k} \left\| T_k^* Q - Q \right\|_{1,\eta^{\rho,\pi}}, \tag{9.13}$$

where $\eta^{\rho,\pi} = (1-\gamma^k)\rho\pi \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^\pi\right)^{-1}$, is the γ-discounted stationary distribution induced by policy π and initial state distribution ρ in MDP $\mathcal{M}_k$.

Proof. We start by providing the following equality, recalling that $T_k^* Q = T_k^\pi Q$, being π the greedy policy w.r.t. Q:

$$\begin{aligned} Q_k^\pi - Q &= T_k^\pi Q_k^\pi - T_k^\pi Q + T_k^* Q - Q \\ &= \gamma^k P_k^\pi \left(Q_k^\pi - Q\right) + T_k^* Q - Q \\ &= \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^\pi\right)^{-1} \left(T_k^* Q - Q\right), \end{aligned}$$

where the last equality follows from the properties of the Neumann series. We take the expectation w.r.t. to the distribution $\rho\pi$ both sides. For the left hand side we have:

$$J_k^{\rho,\pi} - J^\rho = \rho\pi Q_k^\pi - \rho\pi Q.$$

Concerning the right hand side, instead, we have:

$$\rho\pi \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k P_k^\pi\right)^{-1} \left(T_k^* Q - Q\right) = \frac{1}{1-\gamma^k} \eta^{\rho,\pi} \left(T_k^* Q - Q\right),$$

where we introduced the γ-discounted stationary distribution (Sutton et al., 1999a) after normalization. Putting all together, we can derive the following inequality:

$$\begin{aligned} J_k^{\rho,\pi} - J^\rho &= \frac{1}{1-\gamma^k} \eta^{\rho,\pi} \left(T_k^* Q - Q\right) \\ &\geqslant -\frac{1}{1-\gamma^k} \eta^{\rho,\pi} \left|T_k^* Q - Q\right| \\ &= -\frac{1}{1-\gamma^k} \left\|T_k^* Q - Q\right\|_{1,\eta^{\rho,\pi}}. \end{aligned}$$

□

Heuristic To get a usable bound from Lemma 9.9, we need to make some simplifications. First, we assume that $\mathcal{D} \sim \nu$ is composed of m *trajectories*, i.e., $\mathcal{D} = \{\tau_i\}_{i=1}^m$ and the initial states are sampled as $S_{\tau_i,0} \sim \rho$. In this way, J^ρ can be estimated from samples as:

$$\widehat{J}^\rho = \frac{1}{m}\sum_{i=1}^{m} V(S_{\tau_i,0}).$$

Second, since we are unable to compute expectations over $\eta^{\rho,\pi}$, we replace it with the sampling distribution ν.[7] Lastly, estimating the expected Bellman residual is problematic since its empirical version is biased (Antos et al., 2008). Thus, we resort to an approach similar to (Farahmand and Szepesvári, 2011), assuming to have a regressor Reg able to output an approximation $\widetilde{Q}_k$ of $T_k^* Q$. We can proceed to the decomposition, thanks to the triangular inequality:

$$\left\|T_k^* Q - Q\right\|_{1,\nu} \leqslant \left\|\widetilde{Q}_k - Q\right\|_{1,\nu} + \left\|T_k^* Q - \widetilde{Q}_k\right\|_{1,\nu}. \tag{9.14}$$

As discussed in Farahmand and Szepesvári (2011), simply using $\left\|\widetilde{Q}_k - Q\right\|_{1,\nu}$ as a proxy for $\left\|T_k^* Q - Q\right\|_{1,\nu}$ might be overly optimistic. To overcome this problem we must prevent the underestimation of the expected Bellman residual. The idea proposed in Farahmand and Szepesvári (2011) consists in replacing the regression error $\left\|T_k^* Q - \widetilde{Q}_k\right\|_{1,\nu}$ with a high-probability bound $b_{k,\mathcal{G}}$, depending on the function space $\mathcal{G}$ of the chosen regressor Reg. Clearly, we have the new problem of deriving a meaningful bound $b_{k,\mathcal{G}}$. This issue is treated in Section 7.4 of Farahmand and Szepesvári (2011). If $\mathcal{G}$ is a *small* function space, i.e., with finite pseudo-dimension, we can employ a standard learning theory bound (Györfi et al., 2002). Since for the persistence selection, we employ the *same* function space $\mathcal{G}$ and the *same* number of samples m for all persistences $k \in \mathcal{K}$, the value of such a bound will not depend on k and, therefore, it can be neglected in the optimization process. We stress that our goal is to provide a practical method able to suggest a reasonable persistence. In this way, we simply replace $\left\|T_k^* Q - Q\right\|_{1,\nu}$ with $\left\|\widetilde{Q}_k - Q\right\|_{1,\mathcal{D}}$. In practice, we set $Q = Q^{(J)}$ and we obtain $\widetilde{Q}_k$ running PFQI for k additional iterations, setting $\widetilde{Q}_k = Q^{(J+k)}$. Thus, the procedure (Algorithm 9.2) reduces to optimizing the index:

$$\widetilde{k} \in \arg\max_{k \in \mathcal{K}} \{B_k\} = \widehat{J}_k^\rho - \frac{1}{1-\gamma^k}\left\|\widetilde{Q}_k - Q_k\right\|_{1,\mathcal{D}}. \tag{9.15}$$

9.6 Related Works

In this section, we revise the works connected to persistence, focusing on continuous-time RL and temporal abstractions.

[7] This introduces a bias that is negligible if $\left\|\eta^{\rho,\pi}/\nu\right\|_\infty \approx 1$. More intuition about when this condition is realized can be found in Appendix C.1 of Metelli et al. (2020a).

Continuous-time RL Among the first attempts to extend value-based RL to continuous-time there is *advantage updating* (Bradtke and Duff, 1994), in which Q-learning is modified to account for infinitesimal control timesteps. Instead of storing the Q-function, the *advantage function* $A(s,a) = Q(s,a) - V(s)$ is recorded. The continuous time is addressed in Baird (1994) by means of the semi-Markov decision processes (Howard, 1963) for finite-state problems. Restricting our brief treatment to the case of deterministic systems, in which the state evolves through time according to the differential equation $\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t))$, the goal consists in finding the control signal $\mathbf{u}(t) \in \mathscr{B}(\mathcal{U})$, where $\mathcal{U}$ is the space of allowed control signals, for $t \in \mathbb{R}_{\geqslant 0}$ so as to maximize the value function as follows:

$$V^{\mathbf{u}}(\mathbf{s}) = \int_{\mathbb{R}_{\geqslant 0}} e^{\beta t} r(\mathbf{s}(t), \mathbf{u}(t)) \mathrm{d}t,$$

where $\gamma = e^{\beta}$ is the discount factor. The optimal control literature has extensively studied the solution of the Hamilton-Jacobi-Bellman equation (Kirk, 2004), i.e., the continuous-time counterpart of the Bellman equation, that, for deterministic systems can be stated as:

$$V^*(\mathbf{s}) = \sup_{\mathbf{u}(\cdot) \in \mathscr{B}(\mathcal{U})} \{V^{\mathbf{u}}(\mathbf{s})\} = \frac{1}{\beta} \sup_{\mathbf{u} \in \mathcal{U}} \left\{ r(\mathbf{s}, \mathbf{u}) + \frac{\partial V^*}{\partial \mathbf{s}}(\mathbf{s})^T \mathbf{f}(\mathbf{s}, \mathbf{u}) \right\}.$$

However, most of the works assume the knowledge of the environment (Bertsekas, 2005; Menaldi, 1994). The model-free case has been tackled by resorting to time (and space) discretizations (Peterson, 1993), with also convergence guarantees (Munos, 1997; Munos and Bourgine, 1997), and coped with function approximation (Dayan and Singh, 1995; Doya, 2000). More recently, the sensitivity of deep RL algorithm to the time discretization has been analyzed in Tallec et al. (2019), proposing an adaptation of advantage updating to deal with small time scales, that can be employed with deep architectures.

Temporal Abstractions The notion of action persistence can be seen as a form of *temporal abstraction* (Sutton et al., 1999b; Precup, 2001). Temporally extended actions have been extensively used in the hierarchical RL literature to model different time resolutions (Singh, 1992a,b), subgoals (Dietterich, 1998), and combined with the actor-critic architectures (Bacon et al., 2017). Persisting an action is a particular instance of a semi-Markov *option*, always lasting k steps. According to the flat option representation (Precup, 2001), we have as initiation set $\mathcal{I} = \mathcal{S}$ the set of all states, as internal policy the policy that plays deterministically the action taken when the option was initiated, i.e., the k-persistent policy, and as termination condition whether k timesteps have passed after the option started, i.e., $\beta(h_t) = \mathbf{1}_{\{t \bmod k = 0\}}$. Interestingly, in Mann et al. (2015) an approximate value iteration procedure for options lasting at least a given number of steps is proposed and analyzed. This approach shares some similarities with action persistence. Nevertheless, we believe that the option framework is more general and usually the time abstractions are related to the semantic of the tasks, rather than based on the modification of the control frequency, like action persistence.

Environment	Expected return at persistence k ($\widehat{J}_k^{\rho,\pi_k}$, mean $\pm$ std)							Performance loss
	$k=1$	$k=2$	$k=4$	$k=8$	$k=16$	$k=32$	$k=64$	(δ mean $\pm$ std)
Cartpole	169.9 ± 5.8	176.5 ± 5.0	$\mathbf{239.5 \pm 4.4}$	10.0 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	0.0 ± 0.0
MountainCar	-111.1 ± 1.5	-103.6 ± 1.6	-97.2 ± 2.0	$\mathbf{-93.6 \pm 2.1}$	$\mathbf{-94.4 \pm 1.8}$	$\mathbf{-92.4 \pm 1.5}$	-136.7 ± 0.9	1.88 ± 0.85
LunarLander	-165.8 ± 50.4	-12.8 ± 4.7	$\mathbf{1.2 \pm 3.6}$	$\mathbf{2.0 \pm 3.4}$	-44.1 ± 6.9	-122.8 ± 10.5	-121.2 ± 8.6	2.12 ± 4.21
Pendulum	$\mathbf{-116.7 \pm 16.7}$	$\mathbf{-113.1 \pm 16.3}$	$\mathbf{-153.8 \pm 23.0}$	-283.1 ± 18.0	-338.9 ± 16.3	-364.3 ± 22.1	-377.2 ± 21.7	3.52 ± 0.0
Acrobot	-89.2 ± 1.1	$\mathbf{-82.5 \pm 1.7}$	$\mathbf{-83.4 \pm 1.3}$	-122.8 ± 1.3	-266.2 ± 1.9	-287.3 ± 0.3	-286.7 ± 0.6	0.80 ± 0.27
Swimmer	21.3 ± 1.1	$\mathbf{25.2 \pm 0.8}$	$\mathbf{25.0 \pm 0.5}$	$\mathbf{24.0 \pm 0.3}$	22.4 ± 0.3	12.8 ± 1.2	14.0 ± 0.2	2.69 ± 1.71
Hopper	58.6 ± 4.8	61.9 ± 4.2	62.2 ± 1.7	59.7 ± 3.1	60.8 ± 1.0	66.7 ± 2.7	$\mathbf{73.4 \pm 1.2}$	5.33 ± 2.32
Walker 2D	61.6 ± 5.5	37.6 ± 4.0	62.7 ± 18.2	$\mathbf{80.8 \pm 6.6}$	$\mathbf{102.1 \pm 19.3}$	$\mathbf{91.5 \pm 13.0}$	$\mathbf{97.2 \pm 17.6}$	5.10 ± 3.74

Table 9.1: *Results of PFQI in different environments and persistences. For each persistence k, we report the sample mean and the standard deviation of the estimated return of the last policy $\widehat{J}_k^{\rho,\pi_k}$. For each environment, the persistence with the highest average performance and the ones not statistically significantly different from that one (Welch's t-test with $p < 0.05$) are in bold. The last column reports the mean and the standard deviation of the performance loss δ between the optimal persistence and the one selected by the index B_k (Equation* (9.15)*).*

9.7 Experimental Evaluation

In this section, we provide the empirical evaluation of PFQI, with the threefold goal: i) proving that a persistence $k > 1$ can boost learning, leading to more profitable policies, ii) assessing the quality of our persistence selection method, and iii) studying how the batch size influences the performance of PFQI policies for different persistences. For additional experiments, the hyperparameter values, and the implementation details, please refer to Appendix D of the original paper (Metelli et al., 2020a).

9.7.1 Main Experiment

We train PFQI, using extra-trees (Geurts et al., 2006) as a regression model, for J iterations and different values of k, starting with the same dataset $\mathcal{D}$ collected at persistence 1. To compare the performance of the learned policies π_k at the different persistences, we estimate their expected return J_k^{ρ,π_k} in the corresponding MDP $\mathcal{M}_k$. Table 9.1 shows the results for different continuous environments and different persistences averaged over 20 runs and highlighting in bold the persistence with the highest average performance and the ones that are not statistically significantly different from that one. Across the different environments we observe some common trends in line with our theory: i) persistence 1 rarely leads to the best performance; ii) excessively increasing persistence prevents the control at all. In Cartpole (Barto et al., 1983), we easily identify a persistence ($k = 4$) that outperforms all the others. In the Lunar Lander (Brockman et al., 2016) persistences $k \in \{4, 8\}$ are the only ones that lead to positive return (i.e., the lander does not crash) and in the Acrobot domain (Geramifard et al., 2015) we identify $k \in \{2, 4\}$ as optimal persistences. A qualitatively different behavior is displayed in Mountain Car (Moore, 1991), Pendulum (Brockman et al., 2016), and Swimmer (Coulom, 2002), where we observe a plateau of three persistences with similar performance. An explanation for this phenomenon is that, in those domains, the optimal policy tends to persist actions on its own, making the difference less evident. Intriguingly, the more complex Mujoco domains, like Hopper and Walker 2D (Erickson et al., 2019), seem to benefit from the higher persistences.

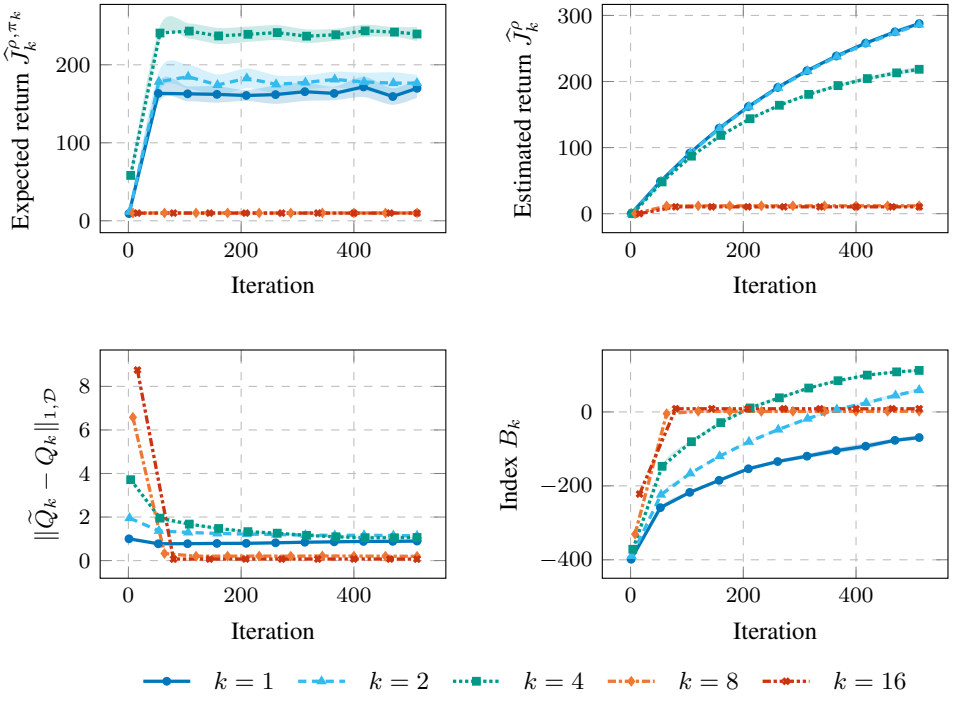


Figure 9.4: *Expected return $\widehat{J}_k^{\rho,\pi_k}$, estimated return $\widehat{J}_k^{\rho}$, estimated expected Bellman residual $\|\widetilde{Q}_k - Q_k\|_{1,\mathcal{D}}$, and persistence selection index B_k in the Cartpole experiment as a function of the number of iterations for different persistences. 20 runs, 95 % c.i.*

9.7.2 Persistence Selection Experiment

To test the quality of our persistence selection method, we compare the performance of the estimated optimal persistence, i.e., the one with the highest estimated expected return $\widehat{k} \in \arg\max_{k \in \mathcal{K}} \{\widehat{J}_k^{\rho,\pi_k}\}$, and the performance of the persistence $\widetilde{k}$ selected by maximizing the index B_k (Equation (9.15)). For each run $i \in \{1, \ldots, 20\}$, we compute the *performance loss* $\delta_i = \widehat{J}_{\widehat{k}}^{\rho,\pi_{\widehat{k}}} - \widehat{J}_{\widetilde{k}_i}^{\rho,\pi_{\widetilde{k}_i}}$ and we report it in the last column of Table 9.1. In the Cartpole experiment, we observe a zero loss, which means that our heuristic always selects the optimal persistence ($k = 4$). Differently, non-zero loss occurs in the other domains, which means that sometimes the index B_k mispredicts the optimal persistence. Nevertheless, in almost all cases the average performance loss is significantly smaller than the magnitude of the return, proving the effectiveness of our heuristics.

In Figure 9.4, we show the learning curves for the Cartpole experiment, highlighting the components that contribute to the index B_k. The first plot reports the estimated *expected return* $\widehat{J}_k^{\rho,\pi_k}$, obtained by averaging 10 trajectories executing π_k in the environment $\mathcal{M}_k$, which confirms that $k = 4$ is the optimal persistence. The second plot shows the *estimated return* $\widehat{J}_k^{\rho}$ obtained by averaging the Q-function Q_k learned with PFQI, over the initial states sampled from ρ. We can see that for $k \in \{1, 2\}$, PFQI tends to overestimate the return, while for $k = 4$ we notice a slight underestimation. The overestimation phenomenon can be explained by the fact that with small persistences we perform a large number of applications of the operator $\widehat{T}^*$, which involves a maximization over the action space, injecting an overestimation bias. By combining this curve with the expected Bellman residual (third plot), we get the value of our persistence selection index B_k (fourth plot). Finally, we observe that B_k correctly ranks persistences 4 and 8, but overestimates persistences 8 and 16, compared to persistence 1.

9.7.3 Batch-Size Experiment

In previous experiments, we assumed we could choose the batch size, however, in real contexts this is not always allowed. In PFQI, lower batch sizes increase the estimation error, but the effect can change according to the used persistence. We investigate how the batch size influences the performance of PFQI policies for different persistences. Therefore, we run PFQI on the Trading environment (described below) changing the number of sampled trajectories. In Figure 9.5, we notice that the performance improves as the batch size increases, for all persistences. Moreover, as it can be noticed in Figure 9.6, if the batch size is small $n \in \{10, 50\}$, higher persistences $k \in \{2, 4, 8\}$ result in better performances, while, with persistence $k = 1$, performance decreases with the iterations. In particular, with 50 trajectories, we can notice that all persistences except from $k = 1$ obtain a positive gain. Since data is taken from real market prices, this environment is very noisy, thus, when the amount of samples is limited, PFQI can exploit higher persistences to mitigate the poor estimation.

FX Trading Environment Description This environment simulates trading on a foreign exchange market. Trader's own currency is USD and it can be traded with EUR. The trader can be in three different positions w.r.t. the foreign currency: long, short or flat, indicated, respectively, with $1, -1, 0$. Short selling is possible, i.e., the agent can sell a stock it does

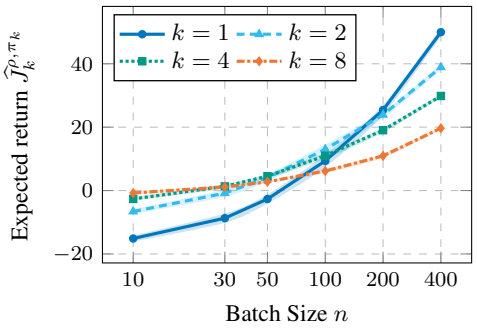


Figure 9.5: *Expected return $\widehat{J}_k^{\rho,\pi_k}$ in the Trading experiment as a function of the batch size. 10 runs, 95 % c.i.*

not own. At each timestep the agent can choose its next position with its action a_t. The exchange rate at time t is p_t, and the reward is equal to $r_t = a_t(p_t - p_{t-1}) - f|a_t - a_{t-1}|$, where the first term is the profit or loss given by the action a_t, and the second term represents the transaction costs, where f is a proportionality constant set to $4 \cdot 10^{-5}$. A timestep corresponds to 1 minute, an episode corresponds to a workday and it is composed by 1170 steps. It is assumed that at each time-step the trader goes long or short of the same unitary amount, thus the profits are not re-invested (and similarly for the losses), which means that the return is the sum of all the daily rewards (with a discount factor equal to 0.9999). The state consists of the last 60 minutes of price differences with the first price of the day ($p_t - p_0$), with the addition of the previous portfolio position as well as the fraction of time remaining until the end of the episode. For our experiments we sampled randomly daily episodes from a window of 64 workdays of 2017, evaluating the performances on the last 20 days of the window.

9.7.4 Summary of the Experiments

The experiments we presented justifies the introduction of persistence. Specifically, we have illustrated three aspects related to action persistence. First, we have shown that action persistence can lead to higher-performing policies when learning under uncertainty (Section 9.7.1). Indeed, the optimal value of persistence is almost never one. Second, we have shown that our persistence selection method, although approximate, is able to select a reasonable persistence value with no need for further interaction with the environment (Section 9.7.2). Finally, the experiment in the trading environment (Section 9.7.3) shows, in alignment with our theoretical findings, that the optimal value of persistence changes as a function of the available samples, finally converging to persistence one as the number of samples grows.

9.8 Open Questions

In this section, we discuss some open questions related to action persistence and we present preliminary results.

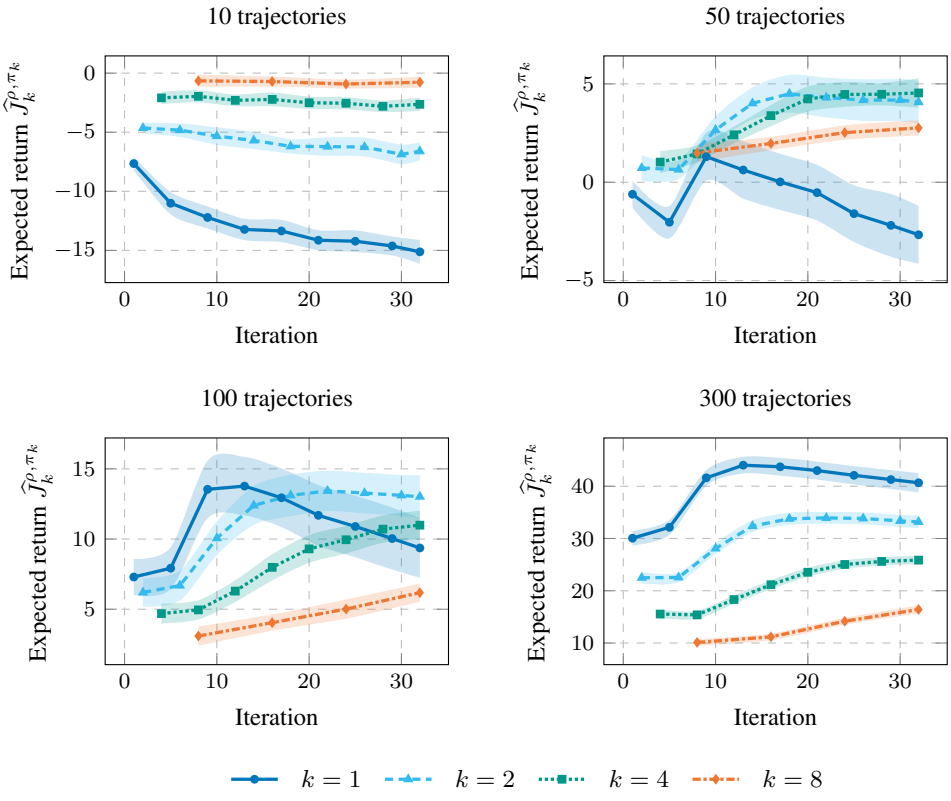
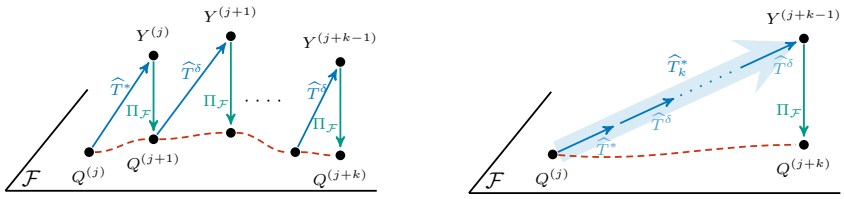


Figure 9.6: *Performances for each persistence along the iterations, with different numbers of trajectories. 10 runs, 95% c.i.*

9.8.1 Improving Exploration with Persistence

Action persistence might have an effect on the exploration properties of distribution ν used to collect samples. To avoid this phenomenon, in the previous experiments, we assumed to feed PFQI with the same dataset collected in the base MDP $\mathcal{M}$, independently of which target persistence k we are interested in. In this section, we briefly analyze what happens when we feed standard FQI with a dataset collected by executing the same policy (e.g., the uniform policy over $\mathcal{A}$) in the k-persistent MDP $\mathcal{M}_k$, in order to estimate the corresponding k-persistence action-value function Q_k^*. In this way, for each persistence k we have a different sampling distribution ν_k used to collect $\mathcal{D}_k$. Refer to Figure 9.7 for a graphical comparison between PFQI executed in the base MDP $\mathcal{M}$ and FQI executed in the k-persistent MDP $\mathcal{M}_k$.

When we compare the performances of the policies obtained with different persistence levels learned starting with a dataset $\mathcal{D}_k \sim \nu_k$, we should consider two different effects: i) how training samples are generated (i.e., the sampling distribution ν_k, which changes for every persistence k); ii) how they affect the learning process in FQI. Unfortunately, in this setting, we are not able to separate the two effects. We compare, for different values



(a) *PFQI on* $\mathcal{M}$ **(b)** *FQI on* $\mathcal{M}_k$

Figure 9.7: *Illustration of (a) PFQI executed in the base MDP* $\mathcal{M}$ *and (b) the standard FQI executed in the* k*-persistent MDP* $\mathcal{M}_k$*.*

of $k \in \mathcal{K} = \{1, 2, \dots 64\}$, the performance of PFQI and the performance of FQI run on the k-persistent MDP $\mathcal{M}_k$. In Figure 9.8, we show the performance at the end of training of the policies obtained with PFQI, the one derived with FQI on $\mathcal{M}_k$, and the uniform policy over the action space. First of all, we observe that when $k = 1$, executing FQI on $\mathcal{M}_1$ is in all regards equivalent to executing PFQI(1) on $\mathcal{M}$. We can see that in the Cartpole environment, fixing a value of $k \in \mathcal{K}$, there is no significant difference in the performances obtained with PFQI and FQI on $\mathcal{M}_k$. The behavior is significantly different when considering Mountain Car. Indeed, we notice that only FQI on $\mathcal{M}_k$ is able to learn a policy that reaches the goal for some specific values of $k \in \mathcal{K}$. We can justify this behavior with the fact that by collecting samples at a persistence k, like in FQI on $\mathcal{M}_k$, the exploration properties of the sampling distribution change, as we can see from the line "Uniform policy". If the input dataset contains no trajectory reaching the goal, our algorithms cannot solve the task. This is why PFQI, that uses persistence 1 to collect the samples, is unable to learn at all.

This experiment gives a preliminary hint on how action persistence can affect exploration. More in general, we wonder which are the characteristics of the environment such that the same sampling policy (e.g., the uniform policy over $\mathcal{A}$) allows performing a more effective exploration. More formally, we ask how the persistence affects the entropy of the stationary distribution induced by the sampling policy.

9.8.2 Learn in $\mathcal{M}_k$ and execute in $\mathcal{M}_{k'}$

In this section, we empirically analyze what happens when a policy is learned with PFQI with a certain persistence level k and executed later on with a different persistence level $k' \neq k$. We consider an experiment on the Cartpole environment, we run PFQI for $k \in \mathcal{K} = \{1, 2, \dots, 256\}$, and then for each k we execute policy π_k (i.e., the policy learned by applying the k-persistent operator) in the k'-persistent MDP $\mathcal{M}_{k'}$ for $k' \in \mathcal{K}$. For each pair (k, k'), Table 9.2 shows the sample mean and the sample standard deviation over 20 runs of the expected return of policy π_k in MDP $\mathcal{M}_k$, i.e., $J_{k'}^{\rho,\pi_k}$. First of all, let us observe that the diagonal of Table 9.2 corresponds to the first row of Table 9.1 (apart from the randomness due to the evaluation). If we select a row k, i.e., we fix the persistence of the operator, we notice that, in the majority of the cases, the persistence k' of the MDP yielding the best performance is smaller than k. Moreover, even if we learn a policy with the operator at a given persistence k and we see that such a policy displays a poor performance in the

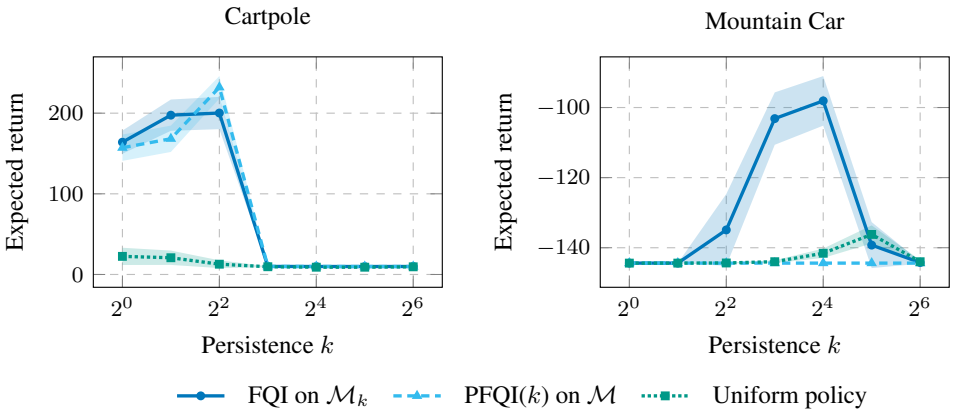


Figure 9.8: *Performance of the policies learned with FQI on $\mathcal{M}_k$, PFQI on $\mathcal{M}$ and the one of the uniform policies for different values of the persistence $k \in \mathcal{K}$. 10 runs. 95% c.i.*

k-persistent MDP (e.g., for $k \geqslant 8$), when we reduce the persistence, the performance of that policy seems to improve.

Figure 9.9 compares for different values of k, corresponding to the persistence of the operator, the performance of the policy π_k when we execute it in $\mathcal{M}_k$ and the performance of π_k in the MDP $\mathcal{M}_{(k')*}$, where $(k')^* \in \arg\max_{k' \in \mathcal{K}} \{\widehat{J}_{k'}^{\rho,\pi_k}\}$. We clearly see that suitably selecting the persistence k' of the MDP in which we will deploy the policy, allows reaching higher performances.

We wonder is whether this behavior is a property of the Cartpole environment or is a general phenomenon that we expect to occur in environments, with certain characteristics. If so, which are those characteristics? Furthermore, when we allow executing π_k in $\mathcal{M}_{k'}$ we should rephrase the persistence selection problem (Equation (9.12)) as follows:

$$k^*, (k')^* \in \arg\max_{k,k' \in \mathcal{K}} \{J_{k'}^{\rho,\pi_k}\}, \tag{9.16}$$

where $\rho \in \mathscr{P}(\mathcal{S})$ is an evaluation distribution. Similarly to the case of Equation (9.12), we cannot directly solve the problem if we are not allowed to interact with the environment. Is it possible to extend Lemma 9.9 and the subsequent heuristic simplifications to get a usable index $B_{k,k'}$ similar to Equation (9.15)?

	$k'=1$	$k'=2$	$k'=4$	$k'=8$	$k'=16$	$k'=32$	$k'=64$	$k'=128$	$k'=256$
$k=1$	$\mathbf{172.0 \pm 6.8}$	$\mathbf{174.1 \pm 6.5}$	113.0 ± 5.3	9.8 ± 0.0	9.7 ± 0.0	9.7 ± 0.1	9.8 ± 0.0	9.7 ± 0.0	9.7 ± 0.0
$k=2$	$\mathbf{178.4 \pm 6.7}$	$\mathbf{182.2 \pm 7.2}$	151.6 ± 5.1	9.9 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0
$k=4$	276.2 ± 3.8	$\mathbf{287.3 \pm 1.1}$	237.0 ± 5.4	10.0 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	9.9 ± 0.0
$k=8$	$\mathbf{284.3 \pm 1.6}$	$\mathbf{281.4 \pm 3.0}$	211.5 ± 4.0	10.0 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.8 ± 0.0	9.9 ± 0.0
$k=16$	$\mathbf{285.9 \pm 1.1}$	$\mathbf{282.9 \pm 2.6}$	223.5 ± 3.2	10.0 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.8 ± 0.0
$k=32$	$\mathbf{285.7 \pm 1.3}$	$\mathbf{283.6 \pm 2.7}$	222.2 ± 3.6	10.0 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	9.9 ± 0.0	9.9 ± 0.0
$k=64$	$\mathbf{283.6 \pm 2.3}$	$\mathbf{284.1 \pm 2.0}$	225.5 ± 4.4	10.0 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	9.9 ± 0.0
$k=128$	$\mathbf{282.9 \pm 2.2}$	$\mathbf{282.5 \pm 3.1}$	221.9 ± 4.7	10.0 ± 0.0	9.8 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.9 ± 0.0
$k=256$	$\mathbf{282.5 \pm 2.3}$	$\mathbf{283.4 \pm 2.4}$	224.3 ± 3.9	10.0 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.9 ± 0.0	9.9 ± 0.0

Table 9.2: *Results of PFQI execution of the policy π_k learned with the k-persistent operator in the k'-persistent MDP $\mathcal{M}_{k'}$ in the Cartpole experiment. For each k, we report the sample mean and the standard deviation of the estimated return of the last policy $\widehat{J}^{\rho,\pi_k}_{k'}$. For each k, the persistence k' with the highest average performance and the ones k' that are not statistically significantly different from that one (Welch's t-test with $p < 0.05$) are in bold.*

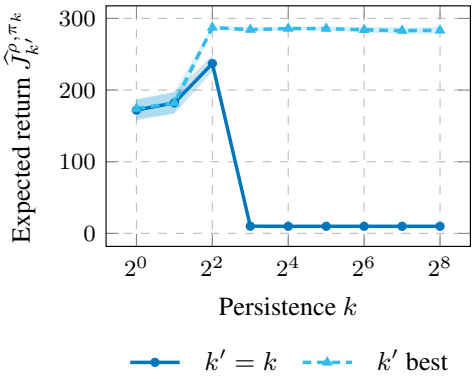


Figure 9.9: *Performance of the policies π_k for $k \in \mathcal{K}$ comparing when they are executed in $\mathcal{M}_k$ and when they are executed in $\mathcal{M}_{(k')*}$. 20 runs, 95% c.i.*

CHAPTER 10

Discussion and Conclusions

In this dissertation, we introduced and explored a novel research area of reinforcement learning, providing theoretical, algorithmic, and experimental contributions. In Part I, we introduced the Configurable Markov Decision Processes, a new tool to model the possibility of configuring the environment in a sequential decision-making problem and we studied the different flavors of interaction between agent and configurator. In Part II, we focused on the cooperative setting, proposing algorithms to learn in both finite and continuous Conf-MDPs and we evaluated them on synthetic and realistic domains. Finally, in Part III, we investigated two applications of the Conf-MDPs: the *policy space identification*, in which we employ environment configurability to infer the agent's policy space and *action persistence* in which we study the configuration of the control frequency of a system.

This research allowed for a better understand of the problem of environment configuration, highlighting, on one hand, its opportunities and identifying, on the other hand, limitations, and possible extensions. In the following, we will revise the contributions of the dissertation and discuss possible future research directions.

10.1 Modeling Environment Configurability

We provided the first formalization of environment configurability. Specifically, we introduced the notion of Configurable Markov Decision Process, as an extension of the traditional MDP model, in which we allow modifications of the transition model and we consider two reward functions R_{Ag} and R_{Conf} to represent the agent's and configurator's

interests (Chapter 4). We extended the notion of value function, Bellman operator, and Bellman equation to the Conf-MDP setting. We observed that, in general, the possibility to alter the environment has to be limited to some extent, to avoid degenerate solutions. This is justified by the fact that the transition model typically encodes portions of the environment that can be configured as well as parts that are immutable (e.g., physical laws). This observation leads to the formalization of the *parametric* Conf-MDP, in which the transition model configuration is performed via a parametric vector.

Then, we focused on investigating the nature of the interaction between the agent and the configurator (Chapter 5). We started with the *cooperative* setting, in which the agent and the configurator share the same interests. This circumstance corresponds to the case in which the reward functions are equal. In such a scenario, it is natural to define a notion of optimality over the policy-transition model joint space. Then, we moved to the *non-cooperative* setting, in which the agent and configurator interests might diverge. In this setting, defining a suitable solution concept is less straightforward. Based on whether the agent is aware of the configurator presence, we proposed to employ game-theoretic equilibria, either Nash equilibrium or Stackelberg equilibrium. For both settings, we extended the notions of value function, Bellman operators, and Bellman equations.

Although our Conf-MDP model is rather simple and the optimality conditions we have proposed for the diverse settings, we believe that there are still situations emerging in real-world applications that cannot be captured. In the following, we outline some of them, that might lead, in the future, to new research directions.

Cost of Environment Configuration Differently from policy learning, in many real-world scenarios the configuration of environmental parameters is an activity that has to be carried out with particular care since it can lead to unsafe behaviors. Moreover, compared to policy learning, it might be performed less frequently and involve additional costs. In our solution concepts, we did not include explicitly a component to account for the cost of altering the environment, although this circumstance was already considered in Silva et al. (2018, 2019). From the viewpoint of our Conf-MDP definition, including a "configuration cost" component would result in a non-Markovian configurator reward function, explicitly depending on the environment configuration.

Multiple Agents and Multiple Configurators The definition of Conf-MDP we provided in this dissertation assumes the presence of *one* agent and *one* configurator. In principle, we might consider scenarios in which multiple agents interact with one another and with multiple configurators. For instance, in Example 4.3, it is quite natural to consider multiple customers in the supermarket, although it is probably unreasonable to take into account multiple configurators. In more generality, we could extend our Conf-MDP definition, accounting for multiple agent reward functions $(R_{\mathsf{Ag}_i})_{i=1}^{N_{\mathsf{Ag}}}$ and multiple configurator reward function $(R_{\mathsf{Conf}_j})_{j=1}^{N_{\mathsf{Conf}}}$. This scenario would open new forms of interaction since there would be multiple configurator entities acting on the same environmental parameters, whereas each agent would act on its individual policy.

Modeling the Configurator Interests At the beginning Chapter 4, we provided an overview of the configuration activity in relation to the curriculum learning literature. We

believe that our Conf-MDP definition, although quite natural and immediate, does not encompass the curriculum learning setting. Indeed, when the configurator is interested in speeding up the learning process for the agent, in the original MDP, its reward cannot be modeled as a Markovian stationary reward. For instance, a more effective modelization of the configurator interest consists in employing an online learning performance index, like the regret (Lattimore and Szepesvári, 2020), determined by the sequence of policies the agent will learn. Clearly, this definition highlights the asymmetry between agent and configurator, which is not completely captured in the present definition. As a consequence, new solution concepts need to be explored.

Configurable Reward Function In our model, we limited the configuration opportunities to the transition model. In principle, we could allow other elements of the MDP definition to be configured. A very interesting element is the agent reward function. Modifying the reward function is tricky since it alters the agent utility function and consequently, defining a suitable goal of the configuration activity becomes more blurred. We have seen an example, in the control frequency adaptation (Chapter 9), in which the reward function changes, although it can be considered a side effect of modifying the persistence. From a curriculum learning perspective, configuring the agent reward function assumes a more interpretable meaning. Indeed, we might be interested in providing the agent with a reward function that is more informative (e.g., dense vs sparse reward) and allows approaching the optimal policy faster. In some sense, this can be thought of as a form of *reward shaping* (Ng et al., 1999).

10.2 Learning in Configurable Markov Decision Process

We studied the learning problem in Conf-MDPs with attention to the cooperative setting, in which a notion of optimal policy-transition model pair is simple to define. In this setting, we first considered the case of finite Conf-MDPs, devising a safe learning approach, SPMI (Chapter 6). SPMI is essentially a prototypical approximate policy iteration algorithm, endowed with strong theoretical guarantees on the performance improvement. However, SPMI requires the full knowledge of the environment model and, for this reason, its applicability is restricted to toy domains.

For these reasons, we investigated the possibility to devise an algorithm that applies to continuous Conf-MDPs, and that overcomes the limitation of knowing the environment model. REMPS (Chapter 7) imports several notions from the trust-region methods and allows solving parametric Conf-MDPs with a procedure that alternates an optimization and a projection phase. Furthermore, we can endow REMPS with an approximation of the transition model learned from samples. The only assumption requested for the configurator is to know which are the parameters it can act on. REMPS allows scaling Conf-MDPs on more realistic scenarios. The experimental evaluation showed that configuring the environment, on the one hand, allows the agent to learn highly performing policies; on the other hand, it might speed up the learning process itself. Moreover, REMPS displayed the ability to overcome some of the limitations of gradient methods when employed to configure environments, even in the presence of approximate models.

The focus of this dissertation, concerning the learning problem in Conf-MDPs, is limited to the cooperative setting. We believe that there is room for further investigations

in this direction as well as on the study of the properties of the solution concepts for the non-cooperative Conf-MDPs. We provide an overview of these research directions in the following.

Online Learning in Cooperative Conf-MDPs An interesting research direction consists of studying the well-known exploration problem, from the point of view of the Conf-MDPs. In this setting, we would play the role of a configurator that is unaware of the agent's reward function and wants to identify the best configuration, within a suitably defined set, that paired with the corresponding agent's optimal policy, optimizes the long-term reward. This problem could be treated as a bandit problem (Lattimore and Szepesvári, 2020), although with additional care. Indeed, whenever a configuration is set in the environment, the agent needs a certain amount of time to adapt its policy. When should we provide the agent with a new configuration? This is an instance of the exploration vs exploitation tradeoff in which we need to decide whether to exploit the current belief on the best configuration or to explore new configurations to gather more information with the risk of lowering the performance. Furthermore, it might be beneficial to exploit more effectively the *structure* (Lattimore and Munos, 2014) underlying the process. Specifically, if the configurator knew the agent's reward function, it could solve the learning problem offline.

Learning in Non-Cooperative Conf-MDPs The study of Conf-MDPs we carried out so far was limited to the cooperative setting. However, there exist several real-world scenarios in which the agent and the configurator display non-cooperative goals. In principle, we could investigate the possibility to extend the algorithms designed for cooperation Conf-MDPs, such as SPMI and REMPS, to the non-cooperative setting. Clearly, the problem needs to be formulated as learning a suitable equilibrium of the Conf-MDP. A possible line of research consists in adapting the learning dynamics of stochastic games (e.g., Jin et al., 2019; Fiez et al., 2019) to our Conf-MDP setting. Clearly, we could also focus on an online learning approach in which the configurator learns the agent's reward function and then solves the game offline. We are convinced that this direction on non-cooperative Conf-MDPs is very appealing and deserves to be further examined in the future.

10.3 Applications of Configurable Markov Decision Processes

We presented two heterogeneous applications in which the environment configuration opportunities can be beneficial. In *policy space identification* (Chapter 8), we studied the problem of identifying the agent's capabilities in terms of perception, actuation, and mapping, formalized in the notion of policy space. The role of the Conf-MDPs in this task is twofold. First, we see the policy space identification as a relevant tool to properly select the optimal configuration for the agent in a cooperative Conf-MDP. Indeed, agents optimizing the same reward function but having access to different policy spaces might benefit from different environment configurations. Second, environment configuration can be seen as a tool to place the agent in a suitable MDP in which it is induced to reveal its capabilities.

Then, we focused on a different application related to the choice of a suitable control frequency for an RL problem (Chapter 9). This issue is particularly relevant in robotics and makes it manifest an important trade-off between control opportunities (larger at high

frequencies) and sample complexity (lower at low frequencies). We started with an analysis of the performance loss we experience when with *action persistence*, i.e., when we reduce the control frequency by an integer factor of the base one and we discussed the required regularity conditions to bound the loss. Then, we provided an algorithmic contribution with PFQI, a batch RL algorithm that is able to learn approximately the value function at different persistences. Related to the topic of action persistence, we believe there are opportunities for further research, that we outline in the following.

Online Action Persistence We considered the batch RL setting, in which the dataset of samples is fixed and no further interaction with the environment is possible. This setting leads to a notion of fixed optimal persistence, that is maintained for the whole learning process. As supported by intuition, the larger the number of samples the lower the optimal persistence. When we move to the online RL setting, in which the interaction with the environment is possible to collect additional samples, it might be convenient to vary dynamically the action persistence during the learning process. For instance, we could start with a high persistence to reach a policy with a reasonable performance with little data. Then, as the available data grows, we could reduce the persistence, in order to refine the learned policy and, eventually, converge to the optimal one.

Appendices

APPENDIX $\mathcal{A}$

Additional Results and Proofs

In this appendix, we report additional results and proofs we have omitted in the main text of the dissertation.

A.1 Additional Results and Proofs of Chapter 6

Lemma A.1. *Let $\mathbb{A}_{\pi,P}^{\pi',P'}$ be the expected coupled relative advantage function, $\mathbb{A}_{\pi,P}^{\pi',P}$ and $\mathbb{A}_{\pi,P}^{\pi,P'}$ be the expected (uncoupled) policy and model relative advantage functions respectively. Then, it holds that:*

$$\left|\mathbb{A}_{\pi,P}^{\pi',P'} - \left(\mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'}\right)\right| \leqslant 2\left\|\pi' - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi',P}} \left\|P' - P\right\|_{\mathrm{TV},\infty} \times \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\big(U^{\pi,P}(s,a,\cdot)\big)\right\}.$$

Proof. We can rewrite the expected relative advantage $\mathbb{A}_{\pi,P}^{\pi',P'}$ using Lemma 6.1:

$$\begin{aligned}
\mathbb{A}_{\pi,P}^{\pi',P'} &= \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) A_{\pi,P}^{\pi',P'}(s) \\
&= \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \left(A_{\pi,P}^{\pi',P}(s) + \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) A_{\pi,P}^{\pi,P'}(s,a)\right) \qquad \text{(P.1)} \\
&= \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) A_{\pi,P}^{\pi',P}(s) + \int_{\mathcal{S}}\int_{\mathcal{A}} \mu_\gamma^{\pi,P}(\mathrm{d}s)\pi(\mathrm{d}a|s) A_{\pi,P}^{\pi,P'}(s,a)
\end{aligned}$$

$$
\begin{aligned}
&+ \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) A_{\pi,P}^{\pi,P'}(s,a) \\
&= \mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'} + \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) A_{\pi,P}^{\pi,P'}(s,a), \qquad \text{(P.2)}
\end{aligned}
$$

where line (P.1) comes from Lemma 6.1. From Equation (P.2) we can straightforwardly state the following inequalities:

$$
\begin{aligned}
\mathbb{A}_{\pi,P}^{\pi',P'} &\geqslant \mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'} - \left| \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) A_{\pi,P}^{\pi,P'}(s,a) \right|, \\
\mathbb{A}_{\pi,P}^{\pi',P'} &\leqslant \mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'} + \left| \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) A_{\pi,P}^{\pi,P'}(s,a) \right|.
\end{aligned}
$$

Then, we bound the absolute value in the right hand side:

$$
\begin{aligned}
\left| \mathbb{A}_{\pi,P}^{\pi',P'} - \left(\mathbb{A}_{\pi,P}^{\pi',P} + \mathbb{A}_{\pi,P}^{\pi,P'} \right) \right| &\leqslant \left| \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) A_{\pi,P}^{\pi,P'}(s,a) \right| \\
&\leqslant \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \left| \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s) A_{\pi,P}^{\pi,P'}(s,a) \right| \\
&\leqslant \int_{\mathcal{S}} \mu_\gamma^{\pi,P}(\mathrm{d}s) \left\| \pi'(\cdot|s) - \pi(\cdot|s) \right\|_{\mathrm{TV}} \mathrm{sp}\left(A_{\pi,P}^{\pi,P'}(s,\cdot) \right) \qquad \text{(P.3)} \\
&\leqslant \left\| \pi' - \pi \right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \mathrm{sp}\left(A_{\pi,P}^{\pi,P'} \right), \qquad \text{(P.4)}
\end{aligned}
$$

where line (P.3) follows from Lemma 6.5 and line (P.4) derives from observing that:

$$
\mathrm{sp}\left(A_{\pi,P}^{\pi,P'}(s,\cdot) \right) \leqslant \sup_{s \in \mathcal{S}} \left\{ \mathrm{sp}\left(A_{\pi,P}^{\pi,P'}(s,\cdot) \right) \right\} \leqslant \mathrm{sp}\left(A_{\pi,P}^{\pi,P'} \right).
$$

We conclude by bounding the term $\mathrm{sp}\left(A_{\pi,P}^{\pi,P'} \right)$:

$$
\begin{aligned}
\mathrm{sp}\left(A_{\pi,P}^{\pi,P'} \right) &\leqslant 2 \left\| A_{\pi,P}^{\pi,P'} \right\|_\infty \\
&\leqslant 2 \sup_{s \in \mathcal{S}, a \in \mathcal{A}} \left\{ \int_{\mathcal{S}} \left(P'(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|s,a) \right) U^{\pi,P}(s,a,s') \right\} \\
&\leqslant 2 \sup_{s \in \mathcal{S}, a \in \mathcal{A}} \left\{ \left\| P'(\cdot|s,a) - P(\cdot|s,a) \right\|_{\mathrm{TV}} \mathrm{sp}\left(U^{\pi,P}(s,a,\cdot) \right) \right\} \qquad \text{(P.5)} \\
&\leqslant 2 \left\| P' - P \right\|_{\mathrm{TV},\infty} \sup_{s, \in \mathcal{S}, a \in \mathcal{A}} \left\{ \mathrm{sp}\left(U^{\pi,P}(s,a,\cdot) \right) \right\},
\end{aligned}
$$

where (P.5) follows from Lemma 6.5. Putting all together we get the result. □

This result has an interesting interpretation. It tells that the maximum advantage (or disadvantage) that can be obtained by moving the policy and the model simultaneously is bounded by the advantage (or disadvantage) gained by moving the policy and the model separately and a term that depends on the policy and model distance. Therefore, it can happen that even if moving the policy and the model separately is convenient, the joint movement may not.

Lemma A.2. *Let $\mathbb{A}_{\pi,P}^{\pi',P'}$ be the expected coupled relative advantage function. Then, it holds that:*

$$
\mathrm{sp}\left(A_{\pi,P}^{\pi',P'} \right) \leqslant 2 \left\| \pi' - \pi \right\|_{\mathrm{TV},\infty} \sup_{s \in \mathcal{S}} \left\{ Q^{\pi,P}(s,\cdot) \right\}
$$

$$+ \left\|P' - P\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\big(U^{\pi,P}\big)(s,a,\cdot)\right\}.$$

Proof. Let us start rewriting the expression of the relative advantage $A_{\pi,P}^{\pi',P'}$ using Lemma 6.1:

$$\begin{aligned}
\mathrm{sp}\Big(A_{\pi,P}^{\pi',P'}\Big) &\leqslant 2\left\|A_{\pi,P}^{\pi',P'}\right\|_\infty \\
&= 2\left\|A_{\pi,P}^{\pi',P} + \int_{\mathcal{A}} \pi'(\mathrm{d}a|s) A_{\pi,P}^{\pi,P'}(\cdot,a)\right\|_\infty \\
&\leqslant 2\left\|A_{\pi,P}^{\pi',P}\right\|_\infty + 2\left\|A_{\pi,P}^{\pi,P'}\right\|_\infty.
\end{aligned}$$

Thus, we have to bound the two L_∞-norms $\left\|A_{\pi,P}^{\pi',P}\right\|_\infty$ and $\left\|A_{\pi,P}^{\pi,P'}\right\|_\infty$. Concerning the second term, we already bounded it in the proof of Lemma A.1:

$$\left\|A_{\pi,P}^{\pi,P'}\right\|_\infty \leqslant \left\|P' - P\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\},$$

whereas for the second term, we proceed analogously:

$$\begin{aligned}
\left\|A_{\pi,P}^{\pi',P}\right\|_\infty &= \sup_{s\in\mathcal{S}} \left\{\int_{\mathcal{A}} \left(\pi'(\mathrm{d}a|s) - \pi(\mathrm{d}a|s)\right) Q^{\pi,P}(s,a)\right\} \\
&\leqslant \sup_{s\in\mathcal{S}} \left\{\left\|\pi'(\cdot|s) - \pi(\cdot|s)\right\|_{\mathrm{TV}} \mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\} \\
&\leqslant \left\|\pi' - \pi\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\}.
\end{aligned}$$

□

Theorem 6.9. *For any $\overline{\pi} \in \Pi^{\mathrm{SR}}$ and $\overline{P} \in \mathcal{P}^{\mathrm{SR}}$, the decoupled bound is optimized for:*

$$(\alpha^*, \beta^*) \in \operatorname*{arg\,max}_{(\alpha,\beta)\in\mathcal{V}} \{B(\alpha,\beta)\},$$

where B is the bound in Theorem 6.7 and $\mathcal{V} = \{(\alpha_0^*, 0), (\alpha_1^*, 1), (0, \beta_0^*), (1, \beta_1^*)\}$ and:

$$\begin{aligned}
\alpha_0^* =& \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{4\gamma \sup_{s\in\mathcal{S}} \{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}, \\
\alpha_1^* =& \alpha_0^* - \frac{\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}{2\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} - \frac{\sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\big(U^{\pi,P}(s,a,\cdot)\big)\right\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}}{2\gamma \sup_{s\in\mathcal{S}} \{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}}, \\
\beta_0^* =& \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\pi,\overline{P}}}{4\gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}} \{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}, \\
\beta_1^* =& \beta_0^* - \frac{\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}{2\gamma\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} - \frac{\sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\big(Q^{\pi,P}(s,\cdot)\big)\right\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}}{2\sup_{s\in\mathcal{S},a\in\mathcal{A}} \{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}},
\end{aligned}$$

to be clipped in the interval $[0,1]$.

Proof. Let us write explicitly the update coefficients in the decoupled bound (6.7):

$$J^{\pi',P'} - J^{\pi,P} \geqslant B(\alpha,\beta) = \frac{1}{1-\gamma}\left(\alpha\mathbb{A}_{\pi,P}^{\overline{\pi},P} + \beta\mathbb{A}_{\pi,P}^{\pi,\overline{P}}\right) - \frac{2}{(1-\gamma)^2}$$
$$\times\left(\beta\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left(\alpha\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}} + \beta\gamma\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\right)\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right.$$
$$\left. + \alpha\gamma\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left(\alpha\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}} + \beta\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\right)\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\right).$$

we now take the derivatives w.r.t. α and β to find the stationary points:

$$\frac{\partial B}{\partial\alpha} = \frac{\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{1-\gamma} - \frac{2}{(1-\gamma)^2}\left(2\alpha\gamma\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\right.$$
$$+\beta\gamma\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}$$
$$\left.+\beta\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right),$$
$$\frac{\partial B}{\partial\beta} = \frac{\mathbb{A}_{\pi,P}^{\pi,\overline{P}}}{1-\gamma} - \frac{2}{(1-\gamma)^2}\left(2\beta\gamma\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right.$$
$$+\alpha\gamma\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}$$
$$\left.+\alpha\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right).$$

When the target policy is different from the current one and, symmetrically, the target model is different from the current model the linear system of the derivatives admits a unique solution. We compute the second order derivative to discover the nature of such point:

$$\frac{\partial B^2}{\partial^2\alpha} = -\frac{4\gamma}{(1-\gamma)^2}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}$$
$$\frac{\partial B^2}{\partial\alpha\partial\beta} = \frac{\partial B^2}{\partial\beta\partial\alpha} = -\frac{2}{(1-\gamma)^2}\left(\gamma\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\right.$$
$$\left.+\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right),$$
$$\frac{\partial B^2}{\partial^2\beta} = -\frac{4\gamma}{(1-\gamma)^2}\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\},$$

from the second order derivatives we can compute the Hessian matrix $\mathcal{H}B(\alpha,\beta)$ and the correspondig trace and determinant:

$$\text{tr}(\mathcal{H}B(\alpha,\beta)) = \frac{\partial B^2}{\partial^2\alpha} + \frac{\partial B^2}{\partial^2\beta}$$
$$= -\frac{4\gamma}{(1-\gamma)^2}\left[\left\|\overline{\pi}-\pi\right\|_{\text{TV},\infty}\left\|\overline{\pi}-\pi\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S}}\left\{\text{sp}\left(Q^{\pi,P}(s,\cdot)\right)\right\}\right.$$
$$\left.+\left\|\overline{P}-P\right\|_{\text{TV},\infty}\left\|\overline{P}-P\right\|_{\text{TV},\mu_\gamma^{\pi,P}}\sup_{s\in\mathcal{S},a\in\mathcal{A}}\left\{\text{sp}\left(U^{\pi,P}(s,a,\cdot)\right)\right\}\right] \leqslant 0,$$
$$\det(\mathcal{H}B(\alpha,\beta)) = \frac{\partial B^2}{\partial^2\alpha}\frac{\partial B^2}{\partial^2\beta} - \frac{\partial B^2}{\partial\alpha\partial\beta}\frac{\partial B^2}{\partial\beta\partial\alpha}$$

$$
\begin{aligned}
&= \frac{16\gamma^2}{(1-\gamma)^4} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \\
&\quad \times \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\} \\
&+ \frac{4}{(1-\gamma)^4}\Bigg(\gamma \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\} \\
&\quad + \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\}\Bigg)^2 \\
&\leqslant -\frac{4}{(1-\gamma)^4}\Bigg(\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\Big(U^{\pi,P}(s,a,\cdot)\Big)\right\} \\
&\quad - \gamma \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\Big(Q^{\pi,P}(s,\cdot)\Big)\right\}\Bigg)^2,
\end{aligned}
$$

where we simply bounded $16\gamma^2 \leqslant 16\gamma$. When $\overline{P} \neq P$ and $\overline{\pi} \neq \pi$ we observe that the Hessian matrix is indefinite since both the trace and the determinant are negative. This means that the unique stationary point is a saddle point which is uninteresting for optimization purposes. By the way, $B(\alpha,\beta)$ is a quadratic function, therefore it is continuous on the compact set $[0,1]^2$ and therefore, from Weierstrass theorem, it admits a global maximum (and minimum). Since such point is not a stationary point it must lie on the boundary of $[0,1]^2$.

Then, by setting to zero the equations $\frac{\partial B}{\partial \alpha}\big|_{\beta=0}$, $\frac{\partial B}{\partial \alpha}\big|_{\beta=1}$, $\frac{\partial B}{\partial \beta}\big|_{\alpha=0}$, $\frac{\partial B}{\partial \beta}\big|_{\alpha=1}$ we can obtain the following optimal values (which are clipped to lie in the interval $[0,1]$):

$$
\begin{aligned}
\alpha_0^* &= \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{4\gamma \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\right\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}, \\
\alpha_1^* &= \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\overline{\pi},P}}{4\gamma \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\right\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} \\
&\quad - \frac{\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}{2\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} - \frac{\sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}\big(U^{\pi,P}(s,a,\cdot)\big)\right\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}}{2\gamma \sup_{s\in\mathcal{S}} \left\{\mathrm{sp}(Q^{\pi,P}(s,\cdot))\right\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}}, \\
\beta_0^* &= \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\pi,\overline{P}}}{4\gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\right\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}, \\
\beta_1^* &= \frac{(1-\gamma)\mathbb{A}_{\pi,P}^{\pi,\overline{P}}}{4\gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\right\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty} \left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} \\
&\quad - \frac{\left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}}{2\gamma\left\|\overline{P} - P\right\|_{\mathrm{TV},\mu_\gamma^{\pi,P}}} - \frac{\sup_{s\in\mathcal{S}} \left\{\mathrm{sp}\big(Q^{\pi,P}(s,\cdot)\big)\right\} \left\|\overline{\pi} - \pi\right\|_{\mathrm{TV},\infty}}{2\sup_{s\in\mathcal{S},a\in\mathcal{A}} \left\{\mathrm{sp}(U^{\pi,P}(s,a,\cdot))\right\} \left\|\overline{P} - P\right\|_{\mathrm{TV},\infty}}.
\end{aligned}
$$

Instead, for $\gamma \in (0,1)$, the Hessian is singular when either the target policy or the target model are equal to the current one. Those cases can be treated separately and clearly yield maxima points. When $\overline{P} = P$ then we have $\alpha^* = \alpha_0^*$, when $\overline{\pi} = \pi$ we have $\beta^* = \beta_0^*$. □

A.2 Additional Results and Proofs of Chapter 7

In this appendix, we report the proof of Theorem 7.9. For sake of brevity, we will denote with $\mathcal{X} = \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and with $x = (s,a,s')$ a state-action-state triple. In order to make

the presentation clearer, we revise in the following the formulation of the optimization problems involved in REMPS.

A.2.1 Formulation of the Optimization Problems

The REMPS problem takes as input a stationary distribution $\mu^{\pi,P} \in \mathcal{D}_{\Theta,\Omega}$ and a KL–divergence threshold $\kappa \geqslant 0$ and provides as output a new stationary distribution in the space $\mathcal{D}_{\Theta,\Omega}$. This process is dived into two consecutive phases: *optimization* and *projection*.

Optimization In the *optimization* phase, given a KL–divergence threshold $\kappa > 0$, let $(\pi, P) \in \Pi_\Theta \times \mathcal{P}_\Omega$ be the current policy-configuration pair inducing a stationary distribution $\mu^{\pi,P}$, we seek for a new stationary distribution μ' that solves the following optimization problem PRIMAL$_\kappa$:

$$\max_{\mu' \in \mathscr{P}(\mathcal{X})} J^{\mu'} = \int \mu'(x) r(x) \mathrm{d}x$$
$$\text{s.t.} \quad D_{\mathrm{KL}}(\mu' \| \mu^{\pi,P}) = \int \mu'(x) \log \frac{\mu'(x)}{\mu^{\pi,P}(x)} \mathrm{d}x \leqslant \kappa.$$

This problem, yields to the solution for all $x \in \mathcal{X}$:

$$\mu'(x) = \frac{\mu^{\pi,P}(x) \exp\left(\frac{1}{\eta} r(x)\right)}{\int_{\mathcal{X}} \mu^{\pi,P}(x) \exp\left(\frac{1}{\eta} r(x)\right) \mathrm{d}x}, \tag{A.1}$$

where η is the unique solution of the dual problem DUAL$_\kappa$:

$$\min_{\eta \in [0,\infty)} \eta \log \int_{\mathcal{X}} \mu^{\pi,P}(x) \exp\left(\frac{1}{\eta} r(x) + \kappa\right) \mathrm{d}x. \tag{A.2}$$

In practice, we have no access to $\mu^{\pi,P}$. Therefore, we need to estimate the expectations from samples using a dataset $\{(S_i, A_i, S'_i, R_i)\}_{i=1}^n = \{(X_i, R_i)\}_{i=1}^n$ of n samples collected with $\mu^{\pi,P}$. Notice that we have only access to an empirical estimate of $\mu^{\pi,p}$, which is $\widehat{\mu}^{\pi,P}(x) = \frac{1}{n} \sum_{i=1}^n \delta(x - X_i)$ uniform on the observed samples. Using $\widehat{\mu}^{\pi,P}$ we want to evaluate the performance of a candidate distribution μ' defined over the observed samples. For this purpose, we perform an importance weighting procedure. We define the weight $w(X_i) = \frac{\mu'(X_i)}{\widehat{\mu}^{\pi,P}(X_i)} = n\mu'(X_i)$. The problem we aim to solve becomes $\widetilde{\text{PRIMAL}}_\kappa$:

$$\max_{\mu' \in \mathscr{P}(\{X_i : i \in \{1,2,\ldots,n\}\})} \widetilde{J}^{\mu'} = \frac{1}{n} \sum_{i=1}^n w(X_i) R_i = \sum_{i=1}^n \mu'(X_i) r(X_i)$$
$$\text{s.t.} \quad \widetilde{D}_{\mathrm{KL}}(\mu' \| \mu^{\pi,P}) = \frac{1}{n} \sum_{i=1}^n w(X_i) \log w(X_i)$$
$$= \sum_{i=1}^n \mu'(X_i) \left(\log \mu'(X_i) + \log n\right) \leqslant \kappa.$$

This problem yields a solution which is defined only over the seen state-action-next-state triples $i \in \{1, 2, \dots, n\}$:

$$\mu'(X_i) = \frac{\exp\left(\frac{1}{\widetilde{\eta}} R_i\right)}{\frac{1}{n}\sum_{j=1}^{n} \exp\left(\frac{1}{\widetilde{\eta}} r(X_j)\right)}, \tag{A.3}$$

where $\widetilde{\eta}$ is the unique solution of the dual problem $\widetilde{\text{DUAL}}_\kappa$:

$$\min_{\widetilde{\eta} \in [0,\infty)} \widetilde{\eta} \log \frac{1}{n} \sum_{i=1}^{n} \exp\left(\frac{1}{\widetilde{\eta}} R_i + \kappa\right). \tag{A.4}$$

Once we solved this problem, the new distribution over the whole $\mathcal{X}$ is characterized by just the Lagrange multiplier $\widetilde{\eta}$, for all $x \in \mathcal{X}$:

$$\widetilde{\mu}'(x) = \frac{\mu^{\pi,P}(x) \exp\left(\frac{1}{\widetilde{\eta}} r(x)\right)}{\int_{\mathcal{X}} \mu^{\pi,P}(x) \exp\left(\frac{1}{\widetilde{\eta}} r(x)\right) \mathrm{d}x}. \tag{A.5}$$

We denote the performance of the new distribution $\widetilde{\mu}'$ with $J^{\widetilde{\mu}'} = \int_{\mathcal{X}} \widetilde{\mu}'(x) r(x) \mathrm{d}x$.

Projection In the *projection* phase we aim at finding the best representation of the stationary distribution we got from the optimization phase in a given hypothesis space $\mathcal{D}_{\Theta,\Omega}$. Let μ' be the solution of PRIMAL_κ, the projection problem PROJ can be stated as the moment-projection of μ' onto $\mathcal{D}_{\Theta,\Omega}$. According to the three projections presented in Section 7.3.2, we have:

$$\text{PROJ}_\mu \qquad \max_{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega} H(\mu' \| \mu^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}) = \mathop{\mathbb{E}}_{X \sim \mu'} \left[\log \mu^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}(X)\right] + \text{const},$$

$$\text{PROJ}_{P^\pi} \qquad \begin{aligned} \max_{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega} \mathbb{H}((P')^{\pi'} \| P_{\boldsymbol{\omega}}{}^{\pi_{\boldsymbol{\theta}}}) &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[H((P')^{\pi'}(\cdot|S) \| P_{\boldsymbol{\omega}}{}^{\pi_{\boldsymbol{\theta}}}(\cdot|S))\right] \\ &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[\log p_{\boldsymbol{\omega}}{}^{\pi_{\boldsymbol{\theta}}}(\cdot|S)\right] + \text{const} \end{aligned},$$

$$\text{PROJ}_{\pi,P} \qquad \begin{aligned} \max_{\boldsymbol{\theta} \in \Theta} \mathbb{H}(\pi' \| \pi_{\boldsymbol{\theta}}) &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[H(\pi'(\cdot|S) \| \pi_{\boldsymbol{\theta}}(\cdot|S))\right] \\ &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[\log \pi_{\boldsymbol{\theta}}(\cdot|S)\right] + \text{const} \end{aligned}$$

$$\begin{aligned} \max_{\boldsymbol{\omega} \in \Omega} \mathbb{H}(P \| P_{\boldsymbol{\omega}}) &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[H(P'(\cdot|S,A) \| P_{\boldsymbol{\omega}}(\cdot|S,A))\right] \\ &= \mathop{\mathbb{E}}_{S,A,S' \sim \mu'} \left[\log p_{\boldsymbol{\omega}}(\cdot|S,A)\right] + \text{const}, \end{aligned}$$

where $H(\mu \| \mu')$ is the cross–entropy, since $D_{\mathrm{KL}}(\mu \| \mu') = H(\mu \| \mu') - H(\mu)$, the entropy $H(\mu)$ is independent on μ', and const denotes a constant that does not depend on the quantities we are optimizing on. Clearly, also in this case we need to consider the Monte Carlo estimates obtained from the very same samples $\{X_i\}_{i=1}^n$ collected with $\mu^{\pi,P}$. Let $\widetilde{\mu}'$ be the solution of $\widetilde{\text{PRIMAL}}_\kappa$, the projection problem $\widetilde{\text{PROJ}}$ becomes:

$$\widetilde{\text{PROJ}}_{\mu} \qquad \max_{(\boldsymbol{\theta},\boldsymbol{\omega})\in\Theta\times\Omega} \widetilde{H}(\widetilde{\mu}'\|\mu^{\pi_{\boldsymbol{\theta}},P_{\boldsymbol{\omega}}}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)\log\mu^{\pi_{\boldsymbol{\theta}},P_{\boldsymbol{\omega}}}(X_i) + \text{const},$$

$$\widetilde{\text{PROJ}}_{P^{\pi}} \qquad \max_{(\boldsymbol{\theta},\boldsymbol{\omega})\in\Theta\times\Omega} \widetilde{\mathbb{H}}(\widetilde{(P')^{\pi'}}\|P_{\boldsymbol{\omega}}{}^{\pi_{\boldsymbol{\theta}}}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)\log p_{\boldsymbol{\omega}}{}^{\pi_{\boldsymbol{\theta}}}(S_i'|S_i) + \text{const},$$

$$\widetilde{\text{PROJ}}_{\pi,P} \qquad \max_{\boldsymbol{\theta}\in\Theta} \widetilde{\mathbb{H}}(\widetilde{\pi}'\|\pi_{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)\log\pi_{\boldsymbol{\theta}}(A_i|S_i) + \text{const}$$

$$\max_{\boldsymbol{\omega}\in\Omega} \widetilde{\mathbb{H}}(\widetilde{P'}\|P_{\boldsymbol{\omega}}) = \frac{1}{n}\sum_{i=1}^{n} w(X_i)\log p_{\boldsymbol{\omega}}(S_i'|S_i,A_i) + \text{const},$$

A.2.2 Off-distribution estimation

Given a value of the Lagrange multiplier η inducing μ, let us define the ratio importance weight $\widehat{w}(x)$ and the self-normalized importance weight $\widetilde{w}(x)$ as:

$$\widehat{w}(x) = \frac{\mu(x)}{\mu^{\pi,P}(x)} = \frac{\exp\left(\frac{1}{\eta}r(x)\right)}{\int_{\mathcal{X}}\mu^{\pi,P}(x)\exp\left(\frac{1}{\eta}r(x)\right)\mathrm{d}x},$$

$$\widetilde{w}(x) = \frac{\widehat{w}(x)}{\sum_{i=1}^{n}\widehat{w}(X_i)} = \frac{\exp\left(\frac{1}{\eta}r(x)\right)}{\sum_{i=1}^{n}\exp\left(\frac{1}{\eta}r(X_i)\right)}.$$

Thus, the off-distribution estimator $\widetilde{J}^{\mu}$ which is optimized by $\widetilde{\text{PRIMAL}}_{\kappa}$ is actually a *self–normalized importance weighting* estimate, opposed to the *ratio importance weighting* estimate $\widehat{J}^{\mu}$ which does not appear in the optimization problems, but will be useful in the following:

$$\widehat{J}^{\mu} = \frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)R_i,$$

$$\widetilde{J}^{\mu} = \sum_{i=1}^{n}\widetilde{w}(X_i)R_i.$$

Analogously we can define the KL divergence estimators:

$$\widehat{D}_{KL}(\mu\|\mu^{\pi,P}) = \frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\log\widehat{w}(X_i),$$

$$\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) = \sum_{i=1}^{n}\widetilde{w}(X_i)\log\left(n\widetilde{w}(X_i)\right),$$

and, given a $\mu' \in \mathcal{D}_{\Theta,\Omega}$, we define the cross–entropy estimators:

$$\widehat{H}(\mu\|\mu') = \frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\log\mu'(X_i),$$

$$\widetilde{H}(\mu\|\mu') = \sum_{i=1}^{n} \widetilde{w}(X_i) \log \mu'(X_i).$$

It is well known that the ratio estimation is unbiased while the self-normalized estimator is biased but consistent Owen (2013).

A.2.3 Error Analysis

We have seen in the previous section that we need to solve both phases of the REMPS problem using the samples. Starting with $\mu^{\pi,P}$, PRIMAL$_\kappa$ yields the solution μ' whereas $\widetilde{\text{REMPS}}_\kappa$ provides the solution $\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}$ which is in terms derived from the $\widetilde{\text{PRIMAL}}_\kappa$ problem yielding $\widetilde{\mu}'$ and the $\widetilde{\text{PROJ}}$ problem. There are two sources of error in this process. First of all, $\widetilde{\mu}'$ is obtained from a finite sample and thus it may differ from μ' (*estimation error*). Secondly, we limit to a hypothesis space $\mathcal{D}_{\Theta,\Omega}$ that may not be able to represent $\widetilde{\mu}'$ (*approximation error*). Furthermore, the projection is performed from samples as well (another source of estimation error). The goal of this analysis is to provide a bound to the quantity $J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}')$. To this end, we consider the following decomposition to isolate the contribution of the two phases:

$$J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') = \underbrace{J^{\mu'} - J^{\widetilde{\mu}'}}_{\text{(i)}} + \underbrace{J^{\widetilde{\mu}'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}')}_{\text{(ii)}}.$$

Recall, finally, that $J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') = J^{\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}}$.

Term (i) A typical approach, from Empirical Risk Minimization (ERM), to bound the estimation error is to add and subtract the empirical risk of the empirical risk minimizer $\widetilde{J}^{\widetilde{\mu}'}$ and exploit the fact that this quantity is larger (smaller in supervised learning) than the empirical risk of any other hypothesis in the hypothesis space (being ERM), in particular μ'. However, in our framework, the hypothesis space changes since the constraint on the KL–divergence is estimated from samples and, in principle, it can impose more relaxed/tight conditions. For this purpose, we introduce a new distribution $\overline{\mu}$ which is the optimal solution to the PRIMAL$_\kappa$ problem using the sample constraint. For this reason, $\widetilde{\mu}'$ and $\overline{\mu}$ are searched in the same hypothesis space and thus we can apply the theory from ERM. Clearly, we need to manage the discrepancy between $\overline{\mu}$ and μ'. For this, we use the sensitivity analysis (Section 7.4.2). Let us define the discrepancy in the constraint for a given hypothesis μ:

$$\Delta\kappa(\mu) = D_{\text{KL}}(\mu\|\mu^{\pi,P}) - \widetilde{D}_{KL}(\mu\|\mu^{\pi,P}). \tag{A.6}$$

As a consequence $\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) \leqslant \kappa \iff D_{\text{KL}}(\mu\|\mu^{\pi,P}) \leqslant \kappa + \Delta\kappa(\mu)$. Finally, we define $\Delta\kappa = \sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}} \Delta\kappa(\mu)$. We have the usual two cases. i) If $\Delta\kappa \leqslant 0$ then the exact constraint is always (i.e., for every hypothesis) tighter and thus $J^{\overline{\mu}} \geqslant J^{\mu}$. ii) If $\Delta\kappa > 0$ then there exists at least one hypothesis for which the constraint is looser; thus it might be that $J^{\overline{\mu}} \leqslant J^{\mu}$. In general, the following result holds.

Lemma A.3. *Let μ', $\overline{\mu}$ as defined before. The following bound holds:*

$$J^{\mu'} \leqslant J^{\overline{\mu}} + 2R_{\max} \max\left\{0, \min\left\{\frac{1}{2}, \frac{\Delta\kappa}{\kappa}\right\}\right\}. \tag{A.7}$$

Proof. If $J^{\mu'} - J^{\overline{\mu}} \leqslant 0$ then the theorem holds. Otherwise, it must be that $\Delta\kappa(\mu') \geqslant 0$ (this is because we defined $\overline{\mu}$ as the optimal solution under the sample-based constraint). We define μ_α as in Proposition 7.8, so we get:

$$\begin{aligned}
J^{\mu'} - J^{\overline{\mu}} &\leqslant J^{\mu'} - J^{\mu_\alpha} \\
&\leqslant R_{\max}\left(1 - \frac{\kappa}{\kappa + \Delta\kappa(\mu')}\right)\|\mu' - \mu^{\pi,P}\|_1 \\
&\leqslant R_{\max}\frac{\Delta\kappa(\mu')}{\kappa + \Delta\kappa(\mu')}\|\mu' - \mu^{\pi,P}\|_1 \\
&\leqslant 2R_{\max}\min\left\{\frac{1}{2}, \frac{\Delta\kappa(\mu')}{\kappa}\right\} \\
&\leqslant 2R_{\max}\min\left\{\frac{1}{2}, \frac{\Delta\kappa}{\kappa}\right\},
\end{aligned}$$

where we exploited the fact that $\|\mu' - \mu^{\pi,P}\|_1 \leqslant 2$, $\frac{\Delta\kappa(\mu')}{\kappa+\Delta\kappa(\mu')} \leqslant \frac{\Delta\kappa(\mu')}{\kappa}$, being $\Delta\kappa(\mu') \geqslant 0$, and $\frac{\Delta\kappa(\mu')}{\kappa+\Delta\kappa(\mu')} \leqslant \frac{1}{2}$ being $\Delta\kappa(\mu') \leqslant \kappa$ and finally $\Delta\kappa(\mu') \leqslant \Delta\kappa$. Taking the max between the two cases we get the result. □

Notice that:

$$\max\left\{0, \min\left\{\frac{1}{2}, \frac{\Delta\kappa}{\kappa}\right\}\right\} \leqslant \frac{|\Delta\kappa|}{\kappa} = \frac{1}{\kappa}\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right|,$$

which is a convenient term for using ERM theory. Now we are ready to bound $J^{\mu'} - J^{\widetilde{\mu}'}$.

Lemma A.4. *Let μ' and $\widetilde{\mu}'$ be the solutions of the PRIMAL$_\kappa$ and $\widetilde{\text{PRIMAL}}_\kappa$ problems, the latter using $n > 0$ i.i.d. samples collected with $\mu^{\pi,P}$. Let $\kappa > 0$ be the KL–divergence threshold. Then, it holds that:*

$$J^{\mu'} - J^{\widetilde{\mu}'} \leqslant 2\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\left|J^{\mu} - \widetilde{J}^{\mu}\right| + \frac{2R_{\max}}{\kappa}\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right|. \tag{A.8}$$

Proof. We use a very simple argument of ERM combined with the previous result. Let $\overline{\mu}$ be defined as before, we have:

$$\begin{aligned}
J^{\mu'} - J^{\widetilde{\mu}'} &\leqslant J^{\overline{\mu}} - J^{\widetilde{\mu}'} + 2R_{\max}\max\left\{0, \min\left\{\frac{1}{2}, \frac{\Delta\kappa}{\kappa}\right\}\right\} \\
&\leqslant J^{\overline{\mu}} - J^{\widetilde{\mu}'} + \frac{2R_{\max}}{\kappa}|\Delta\kappa| \\
&= J^{\overline{\mu}} - J^{\widetilde{\mu}'} + \frac{2R_{\max}}{\kappa}|\Delta\kappa| \pm \widetilde{J}^{\widetilde{\mu}'} \\
&\leqslant J^{\overline{\mu}} - \widetilde{J}^{\overline{\mu}} + \widetilde{J}^{\widetilde{\mu}'} - J^{\widetilde{\mu}'} + \frac{2R_{\max}}{\kappa}|\Delta\kappa| \\
&\leqslant 2\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\left|J^{\mu} - \widetilde{J}^{\mu}\right| + \frac{2R_{\max}}{\kappa}\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right|,
\end{aligned}$$

where we exploited the fact that $\widetilde{J}^{\overline{\mu}} \leqslant \widetilde{J}^{\widetilde{\mu}'}$, being $\widetilde{\mu}'$ the ERM over the same hypothesis space. □

Term (ii) To bound this second term it is useful to recall the property of the KL–divergence $D_{\mathrm{KL}}(\mu\|\mu') = H(\mu\|\mu') - H(\mu)$, where $H(\mu\|\mu')$ is the cross-entropy between μ and μ' and $H(\mu)$ is the entropy of μ. When performing the projection, we are minimizing the term $H(\mu\|\mu')$ since $H(\mu)$ does not depend on μ'. We can state the following result for $\widetilde{\mathrm{PROJ}}_{\mu}$.

Lemma A.5. *Let $\widetilde{\mu}'$ and $\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}$ be the solutions of the $\widetilde{\mathrm{PRIMAL}}_{\kappa}$ and $\widetilde{\mathrm{PROJ}}_{\mu}$ problems using $n > 0$ i.i.d. samples collected with $\mu^{\pi,P}$. Let $\kappa > 0$ be the KL–divergence threshold. Then, it holds that:*

$$\begin{aligned} J^{\widetilde{\mu}'} - J^{\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}} \leqslant\ & R_{\max}\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi},P}} \inf_{\mu' \in \mathcal{D}_{\Theta,\Omega}} D_{\mathrm{KL}}(\mu\|\mu')} \\ & + R_{\max}\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi},P}} \sup_{\mu' \in \mathcal{D}_{\Theta,\Omega}} \left|\widehat{H}(\mu\|\mu') - H(\mu\|\mu')\right|}. \end{aligned} \tag{A.9}$$

Proof. Let us define:

$$\epsilon_2 = \sup_{\mu \in \mathcal{D}_{\mu^{\pi},P}} \sup_{\mu' \in \mathcal{D}_{\Theta,\Omega}} \left|\widehat{H}(\mu\|\mu') - H(\mu\|\mu')\right|. \tag{P.6}$$

Consider the best approximation of $\widetilde{\mu}'$ contained in $\mathcal{D}_{\Theta,\Omega}$, let us denote it with μ^*, i.e., $\mu^* \in \arg\min_{\mu \in \mathcal{D}_{\Theta,\Omega}} H(\widetilde{\mu}'\|\mu)$. Then we can state the following inequalities:

$$\begin{aligned} J^{\widetilde{\mu}'} - J^{\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}} &\leqslant R_{\max}\left\|\widetilde{\mu}' - \mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}\right\|_1 \\ &\leqslant R_{\max}\sqrt{2D_{\mathrm{KL}}(\widetilde{\mu}'\|\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}})} && \text{(P.7)} \\ &= R_{\max}\sqrt{2H(\widetilde{\mu}'\|\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}) - 2H(\widetilde{\mu}')} \\ &\leqslant R_{\max}\sqrt{2\widehat{H}(\widetilde{\mu}'\|\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}) - 2H(\widetilde{\mu}') + \epsilon_2} && \text{(P.8)} \\ &= R_{\max}\sqrt{2\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\widetilde{H}(\widetilde{\mu}'\|\mu^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}) - 2H(\widetilde{\mu}') + \epsilon_2} && \text{(P.9)} \\ &\leqslant R_{\max}\sqrt{2\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\widetilde{H}(\widetilde{\mu}'\|\mu^*) - 2H(\widetilde{\mu}') + \epsilon_2} && \text{(P.10)} \\ &= R_{\max}\sqrt{2\widehat{H}(\widetilde{\mu}'\|\mu^*) - 2H(\widetilde{\mu}') + \epsilon_2} \\ &\leqslant R_{\max}\sqrt{2H(\widetilde{\mu}'\|\mu^*) - 2H(\widetilde{\mu}') + 2\epsilon_2} && \text{(P.11)} \\ &= R_{\max}\sqrt{2D_{\mathrm{KL}}(\widetilde{\mu}'\|\mu^*) + 2\epsilon_2} && \text{(P.12)} \\ &\leqslant R_{\max}\sqrt{2D_{\mathrm{KL}}(\widetilde{\mu}'\|\mu^*)} + R_{\max}\sqrt{2\epsilon_2} \\ &\leqslant R_{\max}\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi},P}} \inf_{\mu' \in \mathcal{D}_{\Theta,\Omega}} D_{\mathrm{KL}}(\mu\|\mu')} + R_{\max}\sqrt{2\epsilon_2}, && \text{(P.13)} \end{aligned}$$

where line (P.7) follows from Pinsker inequality, lines (P.8) and (P.11) follow from the hypothesis, line (P.10) follows from the fact that $\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}$ is ERM, line (P.12) follows from the inequality $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ and lines (P.9) and (P.11) follow from the fact that:

$$\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\widetilde{H}(\mu'\|\mu^{\pi_{\widehat{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}) = \widehat{H}(\widetilde{\mu}'\|\mu^{\pi_{\boldsymbol{\theta}}, P_{\boldsymbol{\omega}}}).$$

□

It is pretty straightforward to extent the previous result to the other two projections.

Corollary A.6. *Let $\widetilde{\mu}'$ and $\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}$ be the solutions of the $\widetilde{\text{PRIMAL}}_\kappa$ and $\widetilde{\text{PROJ}}_{P^\pi}$ problems using $n > 0$ i.i.d. samples collected with $\mu^{\pi,P}$. Let $\kappa > 0$ be the KL–divergence threshold. Then, it holds that:*

$$J^{\widetilde{\mu}'} - J^{\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}} \leqslant R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \inf_{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega} \mathop{\mathbb{E}}_{S \sim \mu} \left[D_{\mathrm{KL}} \left((P')^{\pi'}(\cdot|S) \| P_{\boldsymbol{\omega}}^{\pi_{\boldsymbol{\theta}}}(\cdot|S) \right) \right]} + R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \sup_{(\boldsymbol{\theta},\boldsymbol{\omega}) \in \Theta \times \Omega} \left| \widehat{\mathbb{H}}((P')^{\pi'} \| P_{\boldsymbol{\omega}}^{\pi_{\boldsymbol{\theta}}}) - \mathbb{H}((P')^{\pi'} \| P_{\boldsymbol{\omega}}^{\pi_{\boldsymbol{\theta}}}) \right|}.$$

Let $\widetilde{\mu}'$ and $\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}$ be the solutions of the $\widetilde{\text{PRIMAL}}_\kappa$ and $\widetilde{\text{PROJ}}_{\pi,P}$ problems using $n > 0$ i.i.d. samples collected with $\mu^{\pi,P}$. Let $\kappa > 0$ be the KL–divergence threshold. Then, it holds that:

$$\begin{aligned} J^{\widetilde{\mu}'} - J^{\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}} \leqslant & R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \inf_{\boldsymbol{\theta} \in \Theta} \mathop{\mathbb{E}}_{S \sim \mu} \left[D_{\mathrm{KL}} \left(\pi'(\cdot|S) \| \pi_{\boldsymbol{\theta}}(\cdot|S) \right) \right]} \\ & + R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \sup_{\boldsymbol{\theta} \in \Theta} \left| \widehat{\mathbb{H}}(\pi' \| \pi_{\boldsymbol{\theta}}) - \mathbb{H}(\pi' \| \pi_{\boldsymbol{\theta}}) \right|} \\ & + R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \inf_{\boldsymbol{\omega} \in \Omega} \mathop{\mathbb{E}}_{S,A \sim \mu} \left[D_{\mathrm{KL}} \left(P'(\cdot|S,A) \| P_{\boldsymbol{\omega}}(\cdot|S,A) \right) \right]} \\ & + R_{\max}\rho\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \sup_{\boldsymbol{\omega} \in \Omega} \left| \widehat{\mathbb{H}}(P' \| P_{\boldsymbol{\omega}}) - \mathbb{H}(P' \| P_{\boldsymbol{\omega}}) \right|}. \end{aligned}$$

Proof. The result is obtained using an approach analogous to that of Lemma A.5, using Corollary 7.5 and Lemma 7.6. □

From now on we will limit our attention to the case of PROJ_μ. Putting all together we get the following result.

Theorem A.7. *(Error Decomposition) Let $\mu^{\pi,P}$ be the sampling distribution. Let $\kappa > 0$ be the KL–divergence threshold. Let $\mu' \in \mathcal{D}_{\mu^{\pi,P}}$ be the solution of the PRIMAL_κ problem and $(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \in \Theta \times \Omega$ be the solution of the $\widetilde{\text{REMPS}}_\kappa$ problem computed with $n > 0$ i.i.d. samples collected with $\mu^{\pi,P}$. Then, it holds that:*

$$\begin{aligned} J^{\mu'} - J^{\mu^{\pi_{\widetilde{\boldsymbol{\theta}}'}, P_{\widetilde{\boldsymbol{\omega}}'}}} \leqslant & \, 2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left| J^{\mu} - \widetilde{J}^{\mu} \right| \\ & + \frac{2R_{\max}}{\kappa} \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left| \widetilde{D}_{KL}(\mu \| \mu^{\pi,P}) - D_{\mathrm{KL}}(\mu \| \mu^{\pi,P}) \right| \\ & + R_{\max}\sqrt{2 \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \inf_{\mu' \in \mathcal{D}_{\Theta,\Omega}} D_{\mathrm{KL}}(\mu \| \mu')} \\ & + R_{\max}\sqrt{2 \sup_{\mu \in \mathcal{D}^{\mu^{\pi,P}}} \sup_{\mu' \in \mathcal{D}_{\Theta,\Omega}} \left| \widehat{H}(\mu \| \mu') - H(\mu \| \mu') \right|}. \end{aligned}$$

Proof. Just sum together Lemma A.4 and Lemma A.5. □

A.2.4 Finite–Sample Analysis for finite β-moments

In the following, we provide the finite-sample analysis under Assumption 7.3. Since we are not guaranteed that the involved loss functions have finite supremum. This problem can be tackled by resorting to learning bounds that are applicable to unbounded loss functions with bounded moments (Cortes et al., 2019). The main theoretical tool we are going to use in the following comes from Cortes et al. (2019).

Theorem A.8. *Let $\mathcal{H}$ be a family real-valued functions and let $\mathcal{G} = \{L_h(x) : h \in \mathcal{H}\}$ be the family of loss functions associated to $\mathcal{H}$. Assume that $\mathrm{Pdim}(\mathcal{G}) = v$ and that there exists $\alpha \in (1,2)$ such that $\sup_{h\in\mathcal{H}} L_\alpha(h) = \mathbb{E}_X\left[|L_h(X)|^\alpha\right] < +\infty$. Let $\widehat{L}_\alpha(h) = \frac{1}{n}\sum_{i=1}^n |L_h(X_i)|^\alpha$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, for all $h \in \mathcal{H}$ it holds that:*

$$\mathop{\mathbb{E}}_X\left[L_h(X)\right] \leqslant \frac{1}{n}\sum_{i=1}^n L_h(X_i) + 2^{\frac{\alpha+2}{2\alpha}}\sqrt[\alpha]{L_\alpha(h)}\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{4}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\Gamma\left(\alpha,\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{4}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right),$$

and also, with probability at least $1-\delta$, for all $h \in \mathcal{H}$ it holds that:

$$\frac{1}{n}\sum_{i=1}^n L_h(X_i) \leqslant \mathop{\mathbb{E}}_X\left[L_h(X)\right] + 2^{\frac{\alpha+2}{2\alpha}}\sqrt[\alpha]{\widehat{L}_\alpha(h)}\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{4}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\Gamma\left(\alpha,\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{4}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right),$$

where $\Gamma(\alpha,\epsilon) = \frac{\alpha-1}{\alpha} + \frac{1}{\alpha}\left(\frac{\alpha}{\alpha-1}\right)^{\alpha-1}\left(1+\left(\frac{\alpha-1}{\alpha}\right)^{\alpha-1}\log\frac{1}{\epsilon}\right)^{\frac{\alpha-1}{\alpha}}$.

In the following statements, we make use of the Rényi divergence between probability distributions, and its exponentiated version, that we have introduced in Section 3.3.2. We start by showing a trivial application of Theorem A.8 for bounding in probability several deviations of interest.

Lemma A.9. *Let us define $\epsilon = 2^{\frac{\beta+2}{2\beta}}\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{8}{\delta}}{n^{\frac{2(\beta-1)}{\beta}}}}\Gamma\left(\beta,\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{8}{\delta}}{n^{\frac{2(\beta-1)}{\beta}}}}\right)$. Under Assumption 7.3, each of these events holds with probability at least $1-\delta$:*

$$(\mathcal{E}_1) \qquad \forall \mu \in \mathcal{D}_{\mu^{\pi,P}} : \left|\frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i) - 1\right| \leqslant \max\left\{\sqrt[\beta]{d_\beta(\mu\|\mu^{\pi,P})}, \sqrt[\beta]{\widehat{d}_\beta(\mu\|\mu^{\pi,P})}\right\}\epsilon;$$

$$(\mathcal{E}_2) \qquad \forall \mu \in \mathcal{D}_{\mu^{\pi,P}} : \left|\widehat{J}^\mu - J^\mu\right| \leqslant R_{\max}\left\{\sqrt[\beta]{d_\beta(\mu\|\mu^{\pi,P})}, \sqrt[\beta]{\widehat{d}_\beta(\mu\|\mu^{\pi,P})}\right\}\epsilon;$$

$$(\mathcal{E}_4) \qquad \forall \mu \in \mathcal{D}_{\mu^{\pi,P}}, \forall \mu' \in \mathcal{D}_{\Theta,\Omega} : \quad \left|\widehat{H}(\mu\|\mu') - H(\mu\|\mu')\right|$$

$$\leqslant \max\left\{\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\log\mu'(X)\right|^\beta\right]^{1/\beta}, \left(\frac{1}{n}\sum_{i=1}^n\left|\frac{\mu(X_i)}{\mu^{\pi,P}(X_i)}\log\mu'(X_i)\right|^\beta\right)^{1/\beta}\right\}\epsilon.$$

Proof. It is a simple application of Theorem A.8, using Assumption 7.3 and applying the definition of Rényi divergence. □

Concerning the KL–divergence, the derivation is a bit more complicated. We first need the following technical lemma.

Lemma A.10. *Under Assumption 7.3, for any $\alpha \in (1, \beta)$, the following inequality holds:*

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,p}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\log\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^{\alpha}\right]^{1/\alpha} &\leqslant \frac{1}{e}+\frac{\alpha}{\beta-\alpha}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^{\beta}\right]^{1/\alpha}\\ &= \frac{1}{e}+\frac{\alpha}{\beta-\alpha}d_{\beta}(\mu\|\mu^{\pi,P})^{\beta/\alpha}.\end{aligned} \tag{A.10}$$

Proof. Let $y = \mu(x)/\mu^{\pi,P}(x)$. We start proving that the following inequality hold for all $\alpha > 1$:

$$|y\log y| \leqslant \max\left\{\frac{1}{e}, \frac{y^{\alpha}}{\alpha-1}\right\}. \tag{P.14}$$

Let $g(y) = |y\log y|$. For $y \in [0,1]$ we know that $y\log y$ is negative, thus $g(y) = -y\log y$ that has $1/e$ as maximum. Just take the derivative $\partial g/\partial y = -\log y - 1 = 0 \implies y = 1/e \implies g(1/e) = 1/e$. Clearly the second derivative is negative, thus $1/e$ is a maximum and at the extremes $g(0) = g(1) = 0 < 1/e$. We prove that for $y \in [1,\infty)$, $g(y) = y\log y \leqslant \frac{y^{\alpha}}{\alpha-1}$. It suffices to prove that $\log y \leqslant \frac{y^{\alpha-1}}{\alpha-1}$. Consider the function $h(y) = \log y - \frac{y^{\alpha-1}}{\alpha-1}$, it is enough to prove that $h(y) \leqslant 0$ for all $y \in [1,\infty)$. We know that $h(1) = -\frac{1}{\alpha-1} < 0$ and $h(\infty) = -\infty$ and continuous. Therefore we consider the derivative:

$$\frac{\partial h}{\partial y} = \frac{1}{y} - y^{\alpha-2} \leqslant 0 \implies y \geqslant 1. \tag{P.15}$$

Thus $h(y)$ is monotonically decreasing in $[1,\infty)$ and therefore the statement holds. Now we observe that $\max\{x,y\} \leqslant x + y$ for $x, y \geqslant 0$ and we get using Minkowski's inequality:

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\log\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^{\alpha}\right]^{1/\alpha} &\leqslant \mathop{\mathbb{E}}_{X\sim\mu^{\pi,p}}\left[\left(\frac{1}{e}+\frac{1}{\gamma-1}\left(\frac{\mu(X)}{\mu^{\pi,P}(X)}\right)^{\gamma}\right)^{\alpha}\right]^{1/\alpha}\\ &\leqslant \frac{1}{e}+\frac{1}{\gamma-1}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left(\frac{\mu(X)}{\mu^{\pi,P}(X)}\right)^{\gamma\alpha}\right]^{1/\alpha}.\end{aligned}$$

By taking $\gamma\alpha = \beta$ we get the result. □

The following result is an immediate consequence.

Lemma A.11. *For any $\alpha \in (1,2)$, let $\epsilon = 2^{\frac{\alpha+2}{2\alpha}}\sqrt{\frac{v\log\frac{2en}{v}+\log\frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\,\Gamma\left(\alpha, \sqrt{\frac{v\log\frac{2en}{v}+\log\frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right)$. For any $\alpha \in (1,\beta)$, under Assumption 7.3, the following inequality holds with probability $1-\delta$:*

$$\begin{aligned}(\mathcal{E}_3)\qquad &\forall \mu \in \mathcal{D}_{\mu^{\pi,P}}: \quad \left|\widehat{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right|\\ &\leqslant \max\left\{\frac{1}{e}+\frac{\alpha}{\beta-\alpha}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^{\beta}\right]^{1/\alpha}, \left(\frac{1}{n}\sum_{i=1}^{n}|\widehat{w}(X_i)\log\widehat{w}(X_i)|^{\beta}\right)^{1/\beta}\right\}\epsilon.\end{aligned}$$

Proof. It is a simple application of Theorem A.8, using Assumption 7.3 and Lemma A.10. □

Finally, we need the following result to relate the KL–divergence estimated with and without the self–normalized estimator.

Lemma A.12. *For any $\mu \in \mathcal{D}_{\mu^{\pi,P}}$, the following inequality holds:*

$$\left|\widehat{D}_{KL}(\mu\|\mu^{\pi,P}) - \widetilde{D}_{KL}(\mu\|\mu^{\pi,P})\right| \leqslant \left|\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\log\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\right| + 2\log n\left|\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i) - 1\right|.$$

Proof. We perform some algebraic manipulation of the expression:

$$\begin{aligned}
\widehat{D}_{KL}(\mu\|\mu^{\pi,P}) - \widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) &= \frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\log\widehat{w}(X_i) \\
&\quad - \frac{1}{n}\sum_{i=1}^{n}\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)}\log\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\log\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)} + \left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\log\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right) \\
&\quad - \frac{1}{n}\sum_{i=1}^{n}\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)}\log\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)}\log\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)}\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i) - 1\right) \\
&\quad + \left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right)\log\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{w}(X_i)\right).
\end{aligned}$$

Now, consider the term:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)}\log\frac{\widehat{w}(X_i)n}{\sum_{i=1}^{n}\widehat{w}(X_i)} = \sum_{i=1}^{n}\widetilde{w}(X_i)\log\widetilde{w}(X_i) + \log n.$$

Since the $\widetilde{w}(X_i)$ sum up to 1, the summation $\sum_{i=1}^{n}\widetilde{w}(X_i)\log\widetilde{w}(X_i)$ is maximized in absolute value when all $\widehat{w}(X_i)$ are equal, thus $|\sum_{i=1}^{n}\widetilde{w}(X_i)\log\widetilde{w}(X_i)| \leqslant \log n$. By taking the absolute value of the full expression, we get the result. □

Now we can put all together.

Theorem 7.9. *(Finite–Sample Bound) Let $\mu^{\pi,P} \in \mathscr{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{S})$ be the sampling distribution, $\kappa > 0$ be the KL–divergence threshold, $\mu' \in \mathcal{D}_\mu$ be the solution of the PRIMAL$_\kappa$ problem and $(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \in \Theta\times\Omega$ be the solution of the $\widetilde{\text{REMPS}}_\kappa$ problem with PROJ$_\mu$ computed with $n > 0$ samples collected with μ. Then, under Assumptions 4.1, 7.2 and 7.3, for any $\alpha \in (1, \beta)$, there exist two constants χ, ξ and a function $\zeta(n) = \mathcal{O}(\log n)$ depending on α, and on the samples, such that for any $\delta \in (0,1)$, with probability at least $1 - 4\delta$ it holds that:*

$$J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') \leqslant \underbrace{\sqrt{2}R_{\max}\sup_{\mu\in\mathcal{D}_{\mu^{\pi,P}}}\inf_{\overline{\mu}\in\mathcal{D}_{\Theta,\Omega}}\left\{\sqrt{D_{\mathrm{KL}}(\mu\|\overline{\mu})}\right\}}_{\textit{approximation error}}$$

$$+ \underbrace{R_{\max}\chi\sqrt{\epsilon} + R_{\max}\zeta(n)\epsilon + R_{\max}\xi\epsilon^2}_{\textit{estimation error}},$$

where $\epsilon = 2^{\frac{\alpha+2}{2\alpha}} \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}} \Gamma\left(\alpha, \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right)$*, which depend on the pseudo-dimension bound* $v < +\infty$ *and* $\Gamma(\alpha, \tau) = \frac{\alpha-1}{\alpha} + \frac{1}{\alpha}\left(\frac{\alpha}{\alpha-1}\right)^{\alpha-1}\left(1 + \left(\frac{\alpha-1}{\alpha}\right)^{\alpha-1} \log \frac{1}{\tau}\right)^{\frac{\alpha-1}{\alpha}}$.

Proof. We start from Theorem A.7 and we bound each term using Lemma A.9 and Lemma A.11. For brevity, we define $\epsilon = 2^{\frac{\alpha+2}{2\alpha}} \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}} \Gamma\left(\alpha, \sqrt{\frac{v \log \frac{2en}{v} + \log \frac{8}{\delta}}{n^{\frac{2(\alpha-1)}{\alpha}}}}\right)$. Let us start with $\sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} |J^\mu - \widetilde{J}^\mu|$:

$$\begin{aligned}
\sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|J^\mu - \widetilde{J}^\mu\right| &= \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|J^\mu - \widetilde{J}^\mu \pm \widehat{J}^\mu\right| \\
&\leqslant \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|J^\mu - \widehat{J}^\mu\right| + R_{\max} \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|\frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i) - 1\right| \\
&\leqslant 2R_{\max} \max\left\{\sqrt[\alpha]{d_\alpha(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\beta(\mu\|\mu^{\pi,P})}\right\}\epsilon,
\end{aligned}$$

where we exploited events $(\mathcal{E}_1)$ and $(\mathcal{E}_2)$ and simply observed that $\alpha < \beta$ and thus Lemma A.11 holds as well. Consider $\sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right|$:

$$\begin{aligned}
&\sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right| \\
&\quad= \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P}) \pm \widehat{D}_{KL}(\mu\|\mu^{\pi,P})\right| \\
&\quad\leqslant \sup_{\mu \in \mathcal{D}_{\mu^{\pi,P}}} \left|D_{KL}(\mu\|\mu^{\pi,P}) - \widehat{D}_{KL}(\mu\|\mu^{\pi,P})\right| \\
&\qquad+ \left|\left(\frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i)\right)\log\left(\frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i)\right)\right| + 2\log n \left|\frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i) - 1\right|.
\end{aligned}$$

To complete the derivation we have to analyze the term $z \log z$ with $z = \frac{1}{n}\sum_{i=1}^n \widehat{w}(X_i)$. Now using Lemma A.9 and defining $\tau = \max\left\{\sqrt[\alpha]{d_\alpha(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\beta(\mu\|\mu^{\pi,P})}\right\}\epsilon$ we know that $\max\{0, 1-\tau\} \leqslant z \leqslant 1+\tau$ as $z \geqslant 0$. Consider a value of $\tau \in [0,1]$ it is simple to prove that $(1+\tau)\log(1+\tau) \geqslant -(1-\tau)\log(1-\tau)$, therefore $|z \log z| \leqslant (1+\tau)\log(1+\tau)$. Therefore, we have:

$$\begin{aligned}
&\sup_{\mu \in \mathcal{D}_{\mu^{\pi,p}}} \left|\widetilde{D}_{KL}(\mu\|\mu^{\pi,P}) - D_{\mathrm{KL}}(\mu\|\mu^{\pi,P})\right| \leqslant \\
&\qquad\max\left\{\frac{1}{e} + \frac{\alpha}{\beta-\alpha} \mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^\beta\right]^{1/\alpha}, \left(\frac{1}{n}\sum_{i=1}^n |\widehat{w}(X_i)\log\widehat{w}(X_i)|^\beta\right)^{1/\beta}\right\}\epsilon \\
&\qquad+ \left(1 + \max\left\{\sqrt[\alpha]{d_\alpha(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\alpha(\mu\|\mu^{\pi,P})}\right\}\epsilon\right) \\
&\qquad\times \log\left(1 + \max\left\{\sqrt[\alpha]{d_\alpha(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\alpha(\mu\|\mu^{\pi,P})}\right\}\epsilon\right)
\end{aligned}$$

$$+ 2\log n \max\left\{\sqrt[\alpha]{d_\beta(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\alpha(\mu\|\mu^{\pi,P})}\right\}\epsilon.$$

Finally, the term $\sup_{\mu\in\mathcal{D}_{\mu^{\pi},P}} \sup_{\mu'\in\mathcal{D}_{\Theta,\Omega}} \left|\widehat{H}(\mu\|\mu') - H(\mu\|\mu')\right|$ can be bounded using Lemma A.9. We define:

$$f(\alpha) = \max\left\{\frac{1}{e} + \frac{\alpha}{\beta-\alpha}\mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\right|^\beta\right]^{1/\alpha}, \left(\frac{1}{n}\sum_{i=1}^n |\widehat{w}(X_i)\log\widehat{w}(X_i)|^\beta\right)^{1/\beta},\right.$$
$$\sqrt[\alpha]{d_\alpha(\mu\|\mu^{\pi,P})}, \sqrt[\alpha]{\widehat{d}_\alpha(\mu\|d^{\pi,P})}, \mathop{\mathbb{E}}_{X\sim\mu^{\pi,P}}\left[\left|\frac{\mu(X)}{\mu^{\pi,P}(X)}\log\mu'(X)\right|^\beta\right]^{1/\beta},$$
$$\left.\left(\frac{1}{n}\sum_{i=1}^n\left|\frac{\mu(X_i)}{\mu^{\pi,P}(X_i)}\log\mu'(X_i)\right|^\beta\right)^{1/\beta}\right\}.$$

Finally,

$$\begin{aligned}
J^{\mu'} - J(\widetilde{\boldsymbol{\theta}}', \widetilde{\boldsymbol{\omega}}') &\leqslant 4R_{\max}f(\alpha)\epsilon \\
&\quad + \frac{2R_{\max}}{\kappa}\left[f(\alpha)\epsilon + (1+f(\alpha)\epsilon)\log(1+f(\alpha)\epsilon) + 2\log n f(\alpha)\epsilon\right] \\
&\quad + R_{\max}\sqrt{2}\sup_{\mu\in\mathcal{D}_{\mu^{\pi},P}}\inf_{\overline{\mu}\in\mathcal{D}_{\Theta,\Omega}}\sqrt{D_{\mathrm{KL}}(\mu\|\overline{\mu})} + R_{\max}\sqrt{2f(\alpha)\epsilon} \\
&\leqslant 4R_{\max}f(\alpha)\epsilon + \frac{2R_{\max}}{\kappa}(1+2\log n + f(\alpha)\epsilon)f(\alpha)\epsilon \\
&\quad + R_{\max}\sqrt{2}\sup_{\mu\in\mathcal{D}_{\mu^{\pi},P}}\inf_{\overline{\mu}\in\mathcal{D}_{\Theta,\Omega}}\sqrt{D_{\mathrm{KL}}(\mu\|\overline{\mu})} + R_{\max}\sqrt{2f(\alpha)\epsilon} \\
&= \sqrt{2}R_{\max}\sup_{\mu\in\mathcal{D}_{\mu^{\pi},P}}\inf_{\overline{\mu}\in\mathcal{D}_{\Theta,\Omega}}\sqrt{D_{\mathrm{KL}}(\mu\|\overline{\mu})} + R_{\max}\chi\sqrt{\epsilon} + R_{\max}\zeta(n)\epsilon + R_{\max}\xi\epsilon^2,
\end{aligned}$$

where we exploited the fact that $\log(1+x)\leqslant x$ and $\chi = \sqrt{2f(\alpha)}$, $\zeta(n) = 4 + \frac{2}{\kappa}(1+2\log n)f(\alpha)$ and $\xi = \frac{2}{\kappa}$. Since we made a union bound over the events $(\mathcal{E}_1)$, $(\mathcal{E}_2)$, $(\mathcal{E}_3)$ and $(\mathcal{E}_4)$, the statement holds with probability $1-4\delta$. □

A.3 Additional Results and Proofs of Chapter 8

A.3.1 Concentration Result

The goal of this appendix is to provide a probabilistic bound to the differences $\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})$ and $\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})$. To this purpose, we start with a technical lemma (Lemma A.13) which provides a concentration result involving a quantity that will be used later, under Assumption 8.2. Then, we use this result to obtain the concentration of the parameters, i.e., bounding the distance $\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2$ (Theorem A.14), under suitable well–conditioning properties of the involved quantities. Finally, we employ the latter result to prove the concentration of the negative log–likelihood (Corollary A.15). Some parts of the derivation are inspired to Li et al. (2017).

Lemma A.13. *Under Assumption 8.1 and Assumption 8.2, let $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ be a dataset of $n > 0$ independent samples, where $S_i \sim \nu$ and $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$. For any*

$\boldsymbol{\theta} \in \Theta$, let $\mathbf{g}(\boldsymbol{\theta})$ be defined as:

$$\mathbf{g}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right] - \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right]\right). \tag{A.11}$$

Let $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\widehat{\ell}(\boldsymbol{\theta})\} = \frac{1}{n}\sum_{i=1}^{n}\log\pi_{\boldsymbol{\theta}}(A_i|S_i)$. Then, under Assumption 8.2, for any $\delta \in [0,1]$, with probability at least $1-\delta$, it holds that:

$$\left\|\mathbf{g}(\widehat{\boldsymbol{\theta}})\right\|_2 \leqslant \sigma\sqrt{\frac{2d}{n}\log\frac{2d}{\delta}}. \tag{A.12}$$

Proof. The negative log–likelihood of a policy complying with Definition 8.2 is $\mathcal{C}^2(\mathbb{R}^d)$. Thus, since $\widehat{\boldsymbol{\theta}}$ is a minimizer of the negative log–likelihood function $\widehat{\ell}(\boldsymbol{\theta})$, it must fulfill the following first–order condition:

$$\nabla_{\boldsymbol{\theta}}\widehat{\ell}(\widehat{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}}\log\pi_{\widehat{\boldsymbol{\theta}}}(A_i|S_i) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{t}(S_i,A_i) - \mathop{\mathbb{E}}_{A\sim\pi_{\widehat{\boldsymbol{\theta}}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right]\right) = \mathbf{0}. \tag{P.16}$$

As a consequence, we can rewrite the expression of $\mathbf{g}(\widehat{\boldsymbol{\theta}})$ exploiting this condition:

$$\begin{aligned}\mathbf{g}(\widehat{\boldsymbol{\theta}}) &= \frac{1}{n}\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{A\sim\pi_{\widehat{\boldsymbol{\theta}}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right] - \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right]\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{t}(S_i,A_i) - \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)}\left[\mathbf{t}(S_i,A)\right]\right) = \frac{1}{n}\sum_{i=1}^{n}\overline{\mathbf{t}}(S_i,A_i,\boldsymbol{\theta}^{\mathsf{Ag}}).\end{aligned}$$

By recalling that $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$ it immediately follows that $\mathbf{g}(\widehat{\boldsymbol{\theta}})$ is a zero-mean random vector, i.e., $\mathbb{E}_{\substack{S_i\sim\nu\\ A_i\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)}}\left[\mathbf{g}(\widehat{\boldsymbol{\theta}})\right] = \mathbf{0}$. Moreover, under Assumption 8.2, $\mathbf{g}(\widehat{\boldsymbol{\theta}})$ is the sample mean of subgaussian random vectors. Our goal is to bound the probability $\mathbb{P}\left(\left\|\mathbf{g}(\widehat{\boldsymbol{\theta}})\right\|_2 > \epsilon\right)$; to this purpose we consider the following derivation:

$$\begin{aligned}\mathbb{P}\left(\left\|\mathbf{g}(\widehat{\boldsymbol{\theta}})\right\|_2 > \epsilon\right) &= \mathbb{P}\left(\sqrt{\sum_{j=1}^{d} g_j(\widehat{\boldsymbol{\theta}})^2} > \epsilon\right)\\ &\leqslant \mathbb{P}\left(\bigvee_{j=1}^{d}\left|g_j(\widehat{\boldsymbol{\theta}})\right| > \frac{\epsilon}{\sqrt{d}}\right) &\text{(P.17)}\\ &\leqslant \sum_{j=1}^{d}\mathbb{P}\left(\left|g_j(\widehat{\boldsymbol{\theta}})\right| > \frac{\epsilon}{\sqrt{d}}\right), &\text{(P.18)}\end{aligned}$$

where we exploited in line (P.17) the fact that for a d-dimensional vector $\mathbf{x}$ if $\|\mathbf{x}\|_2 > \epsilon$ it must be that at least one component $j = 1,...,d$ satisfy $x_j^2 > \frac{\epsilon^2}{d}$ and we used a union bound over the d dimensions to get line (P.18). Since for each $j = 1,...,d$ we have that $g_j(\widehat{\boldsymbol{\theta}})$ is a zero-mean subgaussian random variable we can bound the deviation using standard results (Boucheron et al., 2013):

$$\mathbb{P}\left(\left|g_j(\widehat{\boldsymbol{\theta}})\right| > \frac{\epsilon}{\sqrt{d}}\right) \leqslant 2\exp\left\{-\frac{\epsilon^2 n}{2d\sigma^2}\right\}. \tag{P.19}$$

Putting all together we get:

$$\mathbb{P}\left(\left\|\mathbf{g}(\widehat{\boldsymbol{\theta}})\right\|_2 > \epsilon\right) \leqslant 2d \exp\left\{-\frac{\epsilon^2 n}{2d\sigma^2}\right\}. \tag{P.20}$$

By setting $\delta = 2d \exp\left\{-\frac{\epsilon^2 n}{2d\sigma^2}\right\}$ and solving for ϵ we get the result.

□

We can now use the previous result to derive the concentration of the parameters, i.e., bounding the deviation $\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2$.

Theorem A.14 (Parameter concentration)**.** *Under Assumption 8.1 and Assumption 8.2, let $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ be a dataset of $n > 0$ independent samples, where $S_i \sim \nu$ and $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$. Let $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\widehat{\ell}(\boldsymbol{\theta})\}$. If the empirical FIM $\widehat{\mathcal{F}}(\boldsymbol{\theta})$ has a positive minimum eigenvalue $\widehat{\lambda}_{\min} > 0$ for all $\boldsymbol{\theta} \in \Theta$, for any $\delta \in [0, 1]$, with probability at least $1 - \delta$, it holds that:*

$$\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2 \leqslant \frac{\sigma}{\widehat{\lambda}_{\min}} \sqrt{\frac{2d}{n} \log \frac{2d}{\delta}}. \tag{A.13}$$

Proof. Recalling that $\mathbf{g}(\boldsymbol{\theta}^{\mathsf{Ag}}) = \mathbf{0}$, we employ the mean value theorem to rewrite $\mathbf{g}(\widehat{\boldsymbol{\theta}})$ centered in $\boldsymbol{\theta}^{\mathsf{Ag}}$:

$$\mathbf{g}(\widehat{\boldsymbol{\theta}}) = \mathbf{g}(\widehat{\boldsymbol{\theta}}) - \mathbf{g}(\boldsymbol{\theta}^{\mathsf{Ag}}) = \widehat{\mathcal{F}}(\overline{\boldsymbol{\theta}}) \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right), \tag{P.21}$$

where $\overline{\boldsymbol{\theta}} = t\widehat{\boldsymbol{\theta}} + (1-t)\boldsymbol{\theta}^{\mathsf{Ag}}$ for some $t \in [0, 1]$ and $\widehat{\mathcal{F}}(\overline{\boldsymbol{\theta}})$ is defined as:

$$\begin{aligned}
\widehat{\mathcal{F}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\theta}) &= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|S_i)} \left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A|S_i) \mathbf{t}(S_i, A)\right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|S_i)} \left[\left(\mathbf{t}(S_i, A) - \mathop{\mathbb{E}}_{\overline{A} \sim \pi_{\boldsymbol{\theta}}(\cdot|S)} \left[\mathbf{t}(S_i, \overline{A})\right]\right) \mathbf{t}(S_i, A)\right] = \widehat{\mathcal{F}}(\boldsymbol{\theta}),
\end{aligned}$$

where we exploited the expression of $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)$ and the definition of Fisher information matrix given in Equation (B.3). Under the hypothesis of the statement, we can derive the following lower bound:

$$\left\|\mathbf{g}(\widehat{\boldsymbol{\theta}})\right\|_2^2 = \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right)^T \widehat{\mathcal{F}}(\overline{\boldsymbol{\theta}})^T \widehat{\mathcal{F}}(\overline{\boldsymbol{\theta}}) \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right) \geqslant \widehat{\lambda}_{\min}^2 \left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2^2. \tag{P.22}$$

By solving for $\left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2$ and applying Lemma A.13 we get the result. □

Finally, we can get the concentration result for the negative log–likelihood.

Corollary A.15 (Negative log–likelihood concentration)**.** *Under Assumption 8.1 and Assumption 8.2, let $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ be a dataset of $n > 0$ independent samples, where $S_i \sim \nu$ and $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$. Let $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\{\widehat{\ell}(\boldsymbol{\theta})\}$. If $\lambda_{\min}(\widehat{\mathcal{F}}(\boldsymbol{\theta})) = \widehat{\lambda}_{\min} > 0$ for all $\boldsymbol{\theta} \in \Theta$, for any $\delta \in [0, 1]$, with probability at least $1 - \delta$, it holds that:*

$$\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) \leqslant \frac{d^2\sigma^4}{\widehat{\lambda}_{\min}^2 n} \log \frac{2d}{\delta}, \tag{A.14}$$

and also:

$$\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) \leqslant \frac{d^2\sigma^4}{\widehat{\lambda}_{\min}^2 n} \log \frac{2d}{\delta}. \tag{A.15}$$

Proof. Let us start with $\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})$. We consider the first order Taylor expansion of the negative log–likelihood centered in $\boldsymbol{\theta}^{\mathsf{Ag}}$:

$$\ell(\widehat{\boldsymbol{\theta}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}^{\mathsf{Ag}})^T \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right) + \frac{1}{2} \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right)^T \mathcal{H}_{\boldsymbol{\theta}} \ell(\overline{\boldsymbol{\theta}}) \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right), \tag{P.23}$$

where $\overline{\boldsymbol{\theta}} = t\widehat{\boldsymbol{\theta}} + (1-t)\boldsymbol{\theta}^{\mathsf{Ag}}$ for some $t \in [0,1]$. We first observe that $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) = \mathbf{0}$ being $\boldsymbol{\theta}^{\mathsf{Ag}}$ the true parameter and we develop $\mathcal{H}_{\boldsymbol{\theta}} \ell(\overline{\boldsymbol{\theta}})$:

$$\begin{aligned}
\mathcal{H}_{\boldsymbol{\theta}} \ell(\overline{\boldsymbol{\theta}}) &= \mathop{\mathbb{E}}_{\substack{S \sim \nu \\ A \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\mathcal{H}_{\boldsymbol{\theta}} \log \pi_{\overline{\boldsymbol{\theta}}}(A|S)\right] \\
&= \mathop{\mathbb{E}}_{\substack{S \sim \nu \\ A \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\nabla_{\boldsymbol{\theta}} \left(\mathbf{t}(S,A) - \mathop{\mathbb{E}}_{\overline{A} \sim \pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)} \left[\mathbf{t}(S,\overline{A})\right]\right)\right] \\
&= \mathop{\mathbb{E}}_{S \sim \nu} \left[\nabla_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\overline{A} \sim \pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)} \left[\mathbf{t}(S,\overline{A})\right]\right] \\
&= \mathop{\mathbb{E}}_{S \sim \nu} \left[\mathop{\mathbb{E}}_{\overline{A} \sim \pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)} \left[\left(\mathbf{t}(S,\overline{A}) - \mathop{\mathbb{E}}_{\widetilde{A} \sim \pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)} \left[\mathbf{t}(S,\widetilde{A})\right]\right) \mathbf{t}(S,\overline{A})^T\right]\right] = \mathop{\mathbb{E}}_{S \sim \nu} \left[\mathcal{F}(\overline{\boldsymbol{\theta}}, S)\right].
\end{aligned}$$

By using Lemma B.3 to bound the maximum eigenvalue of $\mathcal{F}(\overline{\boldsymbol{\theta}}, S)$, we can state the inequality:

$$\frac{1}{2} \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right)^T \mathcal{H}_{\boldsymbol{\theta}} \ell(\overline{\boldsymbol{\theta}}) \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right) \leqslant \frac{d\sigma^2}{2} \left\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2^2. \tag{P.24}$$

Using the concentration result of Theorem A.14, we get the result. Concerning $\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})$, the derivation is analogous with the only difference that the Taylor expansion has to be centered in $\widehat{\boldsymbol{\theta}}$ instead of $\boldsymbol{\theta}^{\mathsf{Ag}}$. □

To conclude this section, we present the following technical lemma.

Theorem A.16. *Under Assumption 8.1 and Assumption 8.2, let $\mathcal{D} = \{(S_i, A_i)\}_{i=1}^n$ be a dataset of $n > 0$ independent samples, where $S_i \sim \nu$ and $A_i \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S_i)$. Let $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, then for any $\epsilon > 0$, it holds that:*

$$\mathbb{P}\left(\left[\ell(\boldsymbol{\theta}) - \widehat{\ell}(\boldsymbol{\theta})\right] - \left[\ell(\boldsymbol{\theta}') - \widehat{\ell}(\boldsymbol{\theta}')\right] > \epsilon\right) \leqslant \exp\left\{-\frac{\epsilon^2 n}{2\left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|_2^2 \sigma^2}\right\}.$$

Proof. We write explicitly the involved expression, using Definition 8.2 and perform some algebraic manipulations:

$$\begin{aligned}
\left[\ell(\boldsymbol{\theta}) - \widehat{\ell}(\boldsymbol{\theta})\right] - \left[\ell(\boldsymbol{\theta}') - \widehat{\ell}(\boldsymbol{\theta}')\right] &= \mathop{\mathbb{E}}_{\substack{S \sim \nu \\ A \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\boldsymbol{\theta}^T \mathbf{t}(S,A) - A(\boldsymbol{\theta}, s)\right] \\
&\quad - \frac{1}{n} \sum_{i=1}^n \left(\boldsymbol{\theta}^T \mathbf{t}(S_i, A_i) - A(\boldsymbol{\theta}, S_i)\right) \\
&\quad - \mathop{\mathbb{E}}_{\substack{S \sim \nu \\ A \sim \pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\left(\boldsymbol{\theta}'\right)^T \mathbf{t}(S,A) - A(\boldsymbol{\theta}', S)\right] \\
&\quad + \frac{1}{n} \sum_{i=1}^n \left(\left(\boldsymbol{\theta}'\right)^T \mathbf{t}(S_i, A_i) - A(\boldsymbol{\theta}', S_i)\right)
\end{aligned}$$

$$= \mathop{\mathbb{E}}_{\substack{S\sim\nu \\ A\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)^T \mathbf{t}(S,A) - \left(A(\boldsymbol{\theta},S) - A(\boldsymbol{\theta}',S)\right) \right] - \frac{1}{n}\sum_{i=1}^{n} \left(\left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)^T \mathbf{t}(S_i,A_i) - \left(A(\boldsymbol{\theta},S_i) - A(\boldsymbol{\theta}',S_i)\right) \right).$$

We are comparing the mean and the sample mean of the random variable $\left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)^T \mathbf{t}(S,A) - \left(A(\boldsymbol{\theta},S) - A(\boldsymbol{\theta}',S)\right)$. Let us now focus on $A(\boldsymbol{\theta},S) - A(\boldsymbol{\theta}',S)$. From the mean value theorem we know that, for some $t \in [0,1]$ and $\overline{\boldsymbol{\theta}} = t\boldsymbol{\theta} + (1-t)\boldsymbol{\theta}'$, we have:

$$A(\boldsymbol{\theta},S) - A(\boldsymbol{\theta}',S) = \nabla_{\boldsymbol{\theta}} A(\overline{\boldsymbol{\theta}},S)^T \left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right). \tag{P.25}$$

From Equation (P.1), we know that $\nabla_{\boldsymbol{\theta}} A(\overline{\boldsymbol{\theta}},S) = \mathbb{E}_{\overline{A}\sim\pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)}\left[\mathbf{t}(S,\overline{A})\right]$. The random variable $\overline{\mathbf{t}}(S,A,\overline{\boldsymbol{\theta}}) = \mathbf{t}(S,A) - \mathbb{E}_{\overline{A}\sim\pi_{\overline{\boldsymbol{\theta}}}(\cdot|S)}\left[\mathbf{t}(S,\overline{A})\right]$ is a subgaussian random variable for any $\overline{\boldsymbol{\theta}} \in \Theta$. Thus, under Assumption 8.2 we have:

$$\left[\ell(\boldsymbol{\theta}) - \widehat{\ell}(\boldsymbol{\theta})\right] - \left[\ell(\boldsymbol{\theta}') - \widehat{\ell}(\boldsymbol{\theta}')\right] = \left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)^T \left(\mathop{\mathbb{E}}_{\substack{S\sim\nu \\ A\sim\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}(\cdot|S)}} \left[\overline{\mathbf{t}}(S,A,\overline{\boldsymbol{\theta}})\right] - \frac{1}{n}\sum_{i=1}^{n} \overline{\mathbf{t}}(S_i,A_i,\overline{\boldsymbol{\theta}}) \right).$$

If we apply Proposition B.5, we get the result. □

A.3.2 Results on Significance and Power of the Tests

Theorem 8.5. *Let $\widehat{I}_c$ be the set of parameter indexes selected by the Identification Rule 8.2 obtained using $n > 0$ i.i.d. samples collected with $\pi_{\boldsymbol{\theta}^{\mathsf{Ag}}}$, with $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta$. Then, under Assumption 8.1 and Assumption 8.2, let $\boldsymbol{\theta}_i^{\mathsf{Ag}} = \arg\min_{\boldsymbol{\theta}\in\Theta_i}\{\ell(\boldsymbol{\theta})\}$ for all $i \in \{1,...,d\}$ and $\xi = \min\left\{1, \frac{\lambda_{\min}}{\sigma^2}\right\}$. If $\widehat{\lambda}_{\min} \geqslant \frac{\lambda_{\min}}{2\sqrt{2}}$ and $\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) \geqslant c_1$, it holds that:*

$$\alpha \leqslant 2d\exp\left\{-\frac{c_1\lambda_{\min}^2 n}{16d^2\sigma^4}\right\},$$

$$\beta \leqslant (2d-1)\sum_{i\in I^{\mathsf{Ag}}} \exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - c_1\right)\lambda_{\min}\xi n}{16(d-1)^2\sigma^2}\right\}.$$

Proof. We start considering $\alpha = \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : i \in \widehat{I}_c\right)$. We employ an argument analogous to that of (Garivier and Kaufmann, 2019):

$$\begin{aligned}
\mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : i \in \widehat{I}_c\right) &= \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : \lambda_i > c_1\right) \\
&= \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : \widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) > \frac{c_1}{2}\right) \\
&\leqslant \mathbb{P}\left(\exists i \notin I^{\mathsf{Ag}} : \widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) > \frac{c_1}{2}\right) \\
&= \mathbb{P}\left(\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) > \frac{c_1}{2}\right) \leqslant 2d\exp\left\{-\frac{c_1\lambda_{\min}^2 n}{16d^2\sigma^4}\right\},
\end{aligned}$$

where we observed that $\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) \geqslant \widehat{\ell}(\widehat{\boldsymbol{\theta}_i})$ as $\boldsymbol{\theta}^{\mathsf{Ag}} \in \Theta_i$ under $\mathcal{H}_0$ and we applied Corollary A.15 in the last line, recalling that $\widehat{\lambda}_{\min} \geqslant \frac{\lambda_{\min}}{2\sqrt{2}}$. For the second inequality, the derivation is a little more

articulated. Concerning $\beta = \mathbb{P}\left(i \in I^{\mathsf{Ag}} : i \notin \widehat{I}\right)$, we first perform a union bound:

$$\mathbb{P}\left(\exists i \in I^{\mathsf{Ag}} : i \notin \widehat{I}_c\right) = \mathbb{P}\left(\bigvee_{i \in I^{\mathsf{Ag}}} i \notin \widehat{I}_c\right) \leqslant \sum_{i \in I^{\mathsf{Ag}}} \mathbb{P}\left(i \notin \widehat{I}_c\right).$$

Let us now focus on the single terms $\mathbb{P}\left(i \notin \widehat{I}_c\right)$. We now perform the following manipulations:

$$\begin{aligned}
\mathbb{P}\left(i \notin \widehat{I}_c\right) &= \mathbb{P}\left(\widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) \leqslant \frac{c_1}{2}\right) \\
&= \mathbb{P}\left(\left[\widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) - \widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \leqslant \frac{c_1}{2}\right) && \text{(P.26)} \\
&\leqslant \mathbb{P}\left(\left[\widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) - \widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \leqslant \frac{c_1}{2}\right) && \text{(P.27)} \\
&= \mathbb{P}\Big(\left[\widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) - \widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right] + \left[\ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \leqslant \\
&\qquad \frac{c_1}{2} + \left[\ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right]\Big) \\
&= \mathbb{P}\Big(\left[\widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}_i})\right] + \left[\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}_i^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \geqslant \\
&\qquad \left[\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] - \frac{c_1}{2}\Big).
\end{aligned}$$

where line (P.27) is obtained by observing that $\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}}) \geqslant 0$. Thus, we have:

$$\begin{aligned}
\mathbb{P}\left(i \notin \widehat{I}_c\right) &\leqslant \mathbb{P}\left(\widehat{\ell}(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) \geqslant \frac{1}{2}\left[\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] - \frac{c_1}{2}\right) \\
&\quad + \mathbb{P}\Bigg(\left[\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \geqslant \\
&\qquad \frac{1}{2}\left[\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right]\Bigg) && \text{(P.28)} \\
&\leqslant \mathbb{P}\left(\widehat{\ell}(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \widehat{\ell}(\widehat{\boldsymbol{\theta}_i}) \geqslant \frac{1}{2}\left[\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] - \frac{c_1}{2}\right) \\
&\quad + \mathbb{P}\Bigg(\left[\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \widehat{\ell}(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})\right] + \left[\widehat{\ell}(\boldsymbol{\theta}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right] \geqslant \\
&\qquad \frac{1}{2}\left[\frac{1}{2}\lambda_{\min}\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right)\left\|\boldsymbol{\theta}_i^{\mathsf{Ag}} - \boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2^2\right]^{\frac{1}{2}}\Bigg) && \text{(P.29)} \\
&\leqslant 2(d-1)\exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - c_1\right)\lambda_{\min}^2 n}{16(d-1)^2\sigma^4}\right\} \\
&\quad + \exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right)\lambda_{\min} n}{16\sigma^2}\right\} && \text{(P.30)} \\
&\leqslant 2(d-1)\exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - c_1\right)\lambda_{\min} n\xi}{16(d-1)^2\sigma^2}\right\} \\
&\quad + \exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}) - \ell(\boldsymbol{\theta}^{\mathsf{Ag}}) - c_1\right)\lambda_{\min} n\xi}{16(d-1)^2\sigma^2}\right\} && \text{(P.31)}
\end{aligned}$$

$$\leqslant (2d-1)\exp\left\{-\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})-\ell(\boldsymbol{\theta}^{\mathsf{Ag}})-c_1\right)\lambda_{\min}n\xi}{16(d-1)^2\sigma^2}\right\}.$$

where line (P.28) derives from the inequality $\mathbb{P}(X+Y\geqslant c)\leqslant\mathbb{P}(X\geqslant a)+\mathbb{P}(Y\geqslant b)$ with $c=a+b$, line (P.29) is obtained by the following second order Taylor expansion, recalling that $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}^{\mathsf{Ag}})=\mathbf{0}$:

$$\begin{aligned}\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})-\ell(\boldsymbol{\theta}^{\mathsf{Ag}}) &= \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}^{\mathsf{Ag}})^T\left(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}-\boldsymbol{\theta}^{\mathsf{Ag}}\right)+\frac{1}{2}\left(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}-\boldsymbol{\theta}^{\mathsf{Ag}}\right)^T\mathcal{H}_{\boldsymbol{\theta}}\ell(\overline{\boldsymbol{\theta}})\left(\boldsymbol{\theta}_i{}^{\mathsf{Ag}}-\boldsymbol{\theta}^{\mathsf{Ag}}\right)\\ &\geqslant \frac{\lambda_{\min}}{2}\left\|\boldsymbol{\theta}_i{}^{\mathsf{Ag}}-\boldsymbol{\theta}^{\mathsf{Ag}}\right\|_2^2,\end{aligned}$$

where $\overline{\boldsymbol{\theta}}=t\boldsymbol{\theta}^{\mathsf{Ag}}+(1-t)\boldsymbol{\theta}_i^{\mathsf{Ag}}$ for some $t\in[0,1]$. Line (P.30) is obtained by applying Corollary A.15, recalling that $\widehat{\lambda}_{\min}\geqslant\frac{\lambda_{\min}}{2\sqrt{2}}$ and Theorem A.16. Finally, line (P.31) derives by introducing the term $\xi=\min\left\{1,\frac{\lambda_{\min}}{\sigma^2}\right\}$ and observing that:

$$\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})-\ell(\boldsymbol{\theta}^{\mathsf{Ag}})-c_1\right)\xi}{(d-1)^2}\leqslant\frac{\left(\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})-\ell(\boldsymbol{\theta}^{\mathsf{Ag}})\right)n}{16}.$$

Clearly, this result is meaningful as long as $\ell(\boldsymbol{\theta}_i{}^{\mathsf{Ag}})-\ell(\boldsymbol{\theta}^{\mathsf{Ag}})-c_1\geqslant 0$.

□

A.4 Additional Results and Proofs of Chapter 9

Lemma A.17. *Let $\mathcal{M}$ be an MDP and $\pi\in\Pi^{\mathrm{SR}}$ be a Markovian stationary policy, then for any $k\in\mathbb{N}_{\geqslant 1}$ the following two identities hold:*

$$\begin{aligned}Q^\pi-Q_k^\pi &= \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}}-\gamma^k\left(P^\pi\right)^k\right)^{-1}\left(\left(T^\pi\right)^kQ_k^\pi-\left(T^\delta\right)^{k-1}T^\pi Q_k^\pi\right)\\ &= \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}}-\gamma^k\left(P^\delta\right)^{k-1}P^\pi\right)^{-1}\left(\left(T^\pi\right)^kQ^\pi-\left(T^\delta\right)^{k-1}T^\pi Q^\pi\right).\end{aligned}$$

Proof. We prove the equalities by exploiting the facts that Q^π and Q_k^π are the fixed points of T^π and T_k^π:

$$\begin{aligned}Q^\pi-Q_k^\pi &= T^\pi Q^\pi-T_k^\pi Q_k^\pi\\ &= \left(T^\pi\right)^kQ^\pi-\left(T^\delta\right)^{k-1}T^\pi Q_k^\pi &\text{(P.32)}\\ &= \left(T^\pi\right)^kQ^\pi-\left(T^\delta\right)^{k-1}T^\pi Q_k^\pi\pm\left(T^\pi\right)^kQ_k^\pi &\text{(P.33)}\\ &= \gamma^k\left(P^\pi\right)^k\left(Q^\pi-Q_k^\pi\right)+\left(\left(T^\pi\right)^kQ_k^\pi-\left(T^\delta\right)^{k-1}T^\pi Q_k^\pi\right), &\text{(P.34)}\end{aligned}$$

where line (P.32) derives from recalling that $Q^\pi=T^\pi Q^\pi$ and exploiting Theorem 9.1, line (P.34) is obtained by exploiting the identity that holds for two generic bounded measurable functions $f,g\in\mathscr{B}(\mathcal{S}\times\mathcal{A})$:

$$\left(T^\pi\right)^kf-\left(T^\pi\right)^kg=\gamma^k\left(P^\pi\right)^k(f-g). \tag{P.35}$$

We prove this identity by induction. For $k=1$ the identity clearly holds. Suppose Equation (P.35) holds for all integers $h<k$, we prove that it holds for k too:

$$\left(T^\pi\right)^kf-\left(T^\pi\right)^kg=T^\pi\left(T^\pi\right)^{k-1}f-T^\pi\left(T^\pi\right)^{k-1}g$$

$$
\begin{aligned}
&= r + \gamma P^\pi \left(T^\pi\right)^{k-1} f - r - P^\pi \gamma \left(T^\pi\right)^{k-1} g \\
&= \gamma P^\pi \left(\left(T^\pi\right)^{k-1} f - \left(T^\pi\right)^{k-1} g\right) && \text{(P.36)} \\
&= \gamma P^\pi \gamma^{k-1} \left(P^\pi\right)^{k-1} (f - g) && \text{(P.37)} \\
&= \gamma^k \left(P^\pi\right)^k (f - g),
\end{aligned}
$$

where line (P.36) derives from the linearity of operator P^π and line (P.37) follows from the inductive hypothesis. From line (P.34) the result follows immediately, recalling that since $\gamma < 1$ the inversion of the operator is well-defined:

$$
\begin{aligned}
&Q^\pi - Q_k^\pi = \gamma^k \left(P^\pi\right)^k (Q^\pi - Q_k^\pi) + \left(\left(T^\pi\right)^k Q_k^\pi - \left(T^\delta\right)^{k-1} T^\pi Q_k^\pi\right) \implies \\
&\left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k \left(P^\pi\right)^k\right)(Q^\pi - Q_k^\pi) = \left(\left(T^\pi\right)^k Q_k^\pi - \left(T^\delta\right)^{k-1} T^\pi Q_k^\pi\right) \implies \\
&Q^\pi - Q_k^\pi = \left(\mathrm{Id}_{\mathcal{S}\times\mathcal{A}} - \gamma^k \left(P^\pi\right)^k\right)^{-1} \left(\left(T^\pi\right)^k Q_k^\pi - \left(T^\delta\right)^{k-1} T^\pi Q_k^\pi\right).
\end{aligned}
$$

The second identity of the statement is obtained with an analogous derivation, in which at line (P.33) we sum and subtract $\left(T^\delta\right)^{k-1} T^\pi Q^\pi$ and we exploit the identity for two bounded measurable functions $f, g \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$:

$$
\left(T^\delta\right)^{k-1} T^\pi Q f - \left(T^\delta\right)^{k-1} T^\pi Q g = \gamma^k \left(P^\delta\right)^{k-1} P^\pi (f - g). \tag{P.38}
$$

□

Lemma A.18. *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy, then for any $k \in \mathbb{N}_{\geqslant 1}$ and any bounded measurable function $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ the following two identities hold:*

$$
\begin{aligned}
\left(T^\pi\right)^{k-1} f - \left(T^\delta\right)^{k-1} f &= \sum_{i=0}^{k-2} \gamma^{i+1} \left(P^\pi\right)^i \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-i} f \\
&= \sum_{i=0}^{k-2} \gamma^{i+1} \left(P^\delta\right)^i \left(P^\pi - P^\delta\right) \left(T^\pi\right)^{k-2-i} f.
\end{aligned}
$$

Proof. We start with the first identity and we prove it by induction on k. For $k = 1$, we have that the left hand side is zero and the summation on the right hand side has no terms. Suppose that the statement holds for every $h < k$, we prove the statement for k:

$$
\begin{aligned}
\left(T^\pi\right)^{k-1} f - \left(T^\delta\right)^{k-1} f &= \left(T^\pi\right)^{k-1} f - \left(T^\delta\right)^{k-1} f \pm \left(T^\pi\right)^{k-2} T^\delta f && \text{(P.39)} \\
&= \left(\left(T^\pi\right)^{k-2} T^\pi f - \left(T^\pi\right)^{k-2} T^\delta f\right) + \left(\left(T^\pi\right)^{k-2} T^\delta f - \left(T^\delta\right)^{k-2} T^\delta f\right) \\
&= \gamma^{k-2} \left(P^\pi\right)^{k-2} \left(T^\pi f - T^\delta f\right) + \left(\left(T^\pi\right)^{k-2} T^\delta f - \left(T^\delta\right)^{k-2} T^\delta f\right) && \text{(P.40)} \\
&= \gamma^{k-1} \left(P^\pi\right)^{k-2} \left(P^\pi - P^\delta\right) f + \sum_{i=0}^{k-3} \gamma^{i+1} \left(P^\pi\right)^i \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-3-i} T^\delta f && \text{(P.41)}
\end{aligned}
$$

$$= \sum_{i=0}^{k-2} \gamma^{i+1} \left(P^\pi\right)^i \left(P^\pi - P^\delta\right) \left(T^\delta\right)^{k-2-i} f, \tag{P.42}$$

where in line (P.40) we exploited the identity at Equation (P.35), line (P.41) derives from observing that $T^\pi f - T^\delta f = \gamma \left(P^\pi - P^\delta\right) f$ and by inductive hypothesis applied on $T^\delta f$ which is a bounded measurable function as well. Finally, line (P.42) follows from observing that the first term completes the summation up to $k-2$. The second identity in the statement can be obtained by an analogous derivation in which at line (P.39) we sum and subtract $\left(T^\delta\right)^{k-2} T^\pi f$ and, later, exploit the identity at Equation (P.38). □

Lemma A.19. *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ that is L_f–LC. Then, under Assumptions 9.1 and 9.2, the following statements hold:*

1. *$T^\pi f$ is $(L_r + \gamma L_P (L_\pi + 1) L_f)$–LC;*
2. *$T^\delta f$ is $(L_r + \gamma (L_P + 1) L_f)$–LC;*
3. *$T^* f$ is $(L_r + \gamma L_P L_f)$–LC.*

Proof. Let $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ be L_f-LC. Consider an application of T^π and $(s,a), (\overline{s},\overline{a}) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
\left|(T^\pi f)(s,a) - (T^\pi f)(\overline{s},\overline{a})\right| &= \Bigg| r(s,a) + \gamma \int_{\mathcal{S}} \int_{\mathcal{A}} P(\mathrm{d}s'|s,a)\pi(\mathrm{d}a'|s') f(s',a') \\
&\quad - r(\overline{s},\overline{a}) - \gamma \int_{\mathcal{S}} \int_{\mathcal{A}} P(\mathrm{d}s'|\overline{s},\overline{a})\pi(\mathrm{d}a'|s') f(s',a') \Bigg| \\
&\leqslant |r(s,a) - r(\overline{s},\overline{a})| && \text{(P.43)} \\
&\quad + \gamma \left| \int_{\mathcal{S}} \left(P(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|\overline{s},\overline{a})\right) \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') f(s',a') \right| && \text{(P.44)} \\
&\leqslant |r(s,a) - r(\overline{s},\overline{a})| + \gamma (L_\pi + 1) L_f \sup_{f:\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} \left(P(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|\overline{s},\overline{a})\right) f(s') \right| && \text{(P.45)} \\
&\leqslant \left(L_r + \gamma L_P (L_\pi + 1) L_f\right) d_{\mathcal{S}\times\mathcal{A}}\left((s,a),(\overline{s},\overline{a})\right), && \text{(P.46)}
\end{aligned}$$

where line (P.44) follows from triangular inequality, line (P.45) is obtained from observing that the function $g_f(s') = \int_{\mathcal{A}} \pi(\mathrm{d}a'|s') f(s',a')$ is $(L_\pi + 1)L_f$–LC, since for any $s, \overline{s} \in \mathcal{S}$:

$$\begin{aligned}
|g_f(s) - g_f(\overline{s})| &= \left| \int_{\mathcal{A}} \pi(\mathrm{d}a|s) f(s,a) - \int_{\mathcal{A}} \pi(\mathrm{d}a|\overline{s}) f(\overline{s},a) \right| \\
&= \left| \int_{\mathcal{A}} \pi(\mathrm{d}a|s) f(s,a) - \int_{\mathcal{A}} \pi(\mathrm{d}a|\overline{s}) f(\overline{s},a) \pm \int_{\mathcal{A}} \pi(\mathrm{d}a|\overline{s}) f(s,a) \right| \\
&\leqslant \left| \int_{\mathcal{A}} \left(\pi(\mathrm{d}a|s) - \pi(\mathrm{d}a|\overline{s})\right) f(s,a) \right| + \left| \int_{\mathcal{A}} \pi(\mathrm{d}a|\overline{s}) \left(f(\overline{s},a) - f(s,a)\right) \right| \\
&\leqslant L_f \sup_{f:\|f\|_L \leqslant 1} \left| \int_{\mathcal{A}} \left(\pi(\mathrm{d}a|s) - \pi(\mathrm{d}a|\overline{s})\right) f(a) \right| \\
&\quad + \left| \int_{\mathcal{A}} \pi(\mathrm{d}a|\overline{s}) \left(f(\overline{s},a) - f(s,a)\right) \right| \\
&\leqslant L_f L_\pi d_{\mathcal{S}}(s,\overline{s}) + L_f d_{\mathcal{S}}(s,\overline{s}),
\end{aligned}$$

where we exploited the fact that L_π–LC. Finally, line (P.46) is obtained by recalling that the reward function is L_r–LC and the transition model is L_P–LC. The derivations are analogous for T^δ and T^*. Concerning T^δ we have:

$$\begin{aligned}
\left|(T^\delta f)(s,a) - (T^\delta f)(\overline{s},\overline{a})\right| &\leqslant |r(s,a) - r(\overline{s},\overline{a})| \\
&\quad + \gamma \left| \int_{\mathcal{S}} \int_{\mathcal{A}} \left(\delta_a(\mathrm{d}a') P(\mathrm{d}s'|s,a) - \delta_{\overline{a}}(\mathrm{d}a') P(\mathrm{d}s'|\overline{s},\overline{a})\right) f(s',a') \right| \\
&\leqslant L_r d_{\mathcal{S}\times\mathcal{A}}\left((s,a),(\overline{s},\overline{a})\right) \\
&\quad + \gamma \left| \int_{\mathcal{S}} \left(P(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|\overline{s},\overline{a})\right) \int_{\mathcal{A}} \delta_a(\mathrm{d}a') f(s',a') \right| \\
&\quad + \gamma \int_{\mathcal{S}} P(\mathrm{d}s'|\overline{s},\overline{a}) \left| \int_{\mathcal{A}} \left(\delta_a(\mathrm{d}a') - \delta_{\overline{a}}(\mathrm{d}a')\right) f(s',a') \right| \\
&\leqslant \left(L_r + \gamma L_f L_P + \gamma L_f\right) d_{\mathcal{S}\times\mathcal{A}}\left((s,a),(\overline{s},\overline{a})\right),
\end{aligned}$$

where we observed that $\int_{\mathcal{A}} \delta_a(\mathrm{d}a') f(s',a') = f(s',a)$ is L_f–LC and exploited the inequality $\int_{\mathcal{A}} \left|\delta_a(\mathrm{d}a') - \delta_{\overline{a}}(\mathrm{d}a')\right| f(s',a') \leqslant L_f d_{\mathcal{A}}(a,\overline{a}) \leqslant L_f d_{\mathcal{S}\times\mathcal{A}}((s,a),(\overline{a},\overline{a}))$. Finally, considering T^*, we have:

$$\begin{aligned}
\left|(T^* f)(s,a) - (T^* f)(\overline{s},\overline{a})\right| &\leqslant |r(s,a) - r(\overline{s},\overline{a})| \\
&\quad + \gamma \left| \int_{\mathcal{S}} \left(P(\mathrm{d}s'|s,a) - P(\mathrm{d}s'|\overline{s},\overline{a})\right) \sup_{a'\in\mathcal{A}} f(s',a') \right| \\
&\leqslant \left(L_r + \gamma L_f L_P\right) d_{\mathcal{S}\times\mathcal{A}}\left((s,a),(\overline{s},\overline{a})\right),
\end{aligned}$$

where we observed that the function $h_f(s') = \sup_{a'\in As} f(s',a')$ is L_f–LC, since:

$$\begin{aligned}
|h_f(s) - h_f(\overline{s})| &= \left| \sup_{a'\in\mathcal{A}} f(s,a') - \sup_{a'\in\mathcal{A}} f(\overline{s},a') \right| \\
&\leqslant \sup_{a'\in\mathcal{A}} \left| f(s,a') - f(\overline{s},a') \right| \\
&\leqslant L_f d_{\mathcal{S}}(s,\overline{s}).
\end{aligned}$$

□

Lemma A.20. *Let $\mathcal{M}$ be an MDP and $\pi \in \Pi^{\mathrm{SR}}$ be a Markovian stationary policy. Then, under Assumptions 9.1 and 9.2, if $\gamma \max\{L_P + 1, L_P(L_\pi + 1)\} < 1$, the functions $f \in \mathcal{Q}_k$ are $L_{\mathcal{Q}_k}$–LC, where:*

$$L_{\mathcal{Q}_k} \leqslant \frac{L_r}{1 - \gamma \max\{L_P + 1, L_P(L_\pi + 1)\}}. \tag{A.16}$$

Furthermore, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ it holds that:

$$d^\pi_{\mathcal{Q}_k}(s,a) \leqslant L_{\mathcal{Q}_k} \mathcal{W}_1\left(P^\pi(\cdot|s,a), P^\delta(\cdot|s,a)\right). \tag{A.17}$$

Proof. First of all consider the action-value function of the k–persistent MDP Q^π_k, which is the fixed point of the operator T^π_k that decomposes into $(T^\delta)^{k-1} T^\pi$ according to Theorem 9.1. It follows that for any $f \in \mathscr{B}(\mathcal{S} \times \mathcal{A})$ we have:

$$Q^\pi_k = \lim_{j\to+\infty} (T^\pi_k)^j f = \lim_{j\to+\infty} \left((T^\delta)^{k-1} T^\pi\right)^j f.$$

We now want to bound the Lipschitz constant of Q_k^π. To this purpose, let us first compute the Lipschitz constant of $T_k^\pi f = ((T^\delta)^{k-1}T^\pi)f$ for $f \in \mathscr{B}(\mathcal{S}\times\mathcal{A})$ being an L_f–LC function. From Lemma A.19 we can bound the Lipschitz constant a_h of $(T^\delta)^h T^\pi f$ for $h \in \{0,...k-1\}$, leading to the sequence:

$$a_h = \begin{cases} L_r + \gamma L_P(L_\pi+1)L_f & \text{if } h = 0 \\ L_r + \gamma(L_P+1)a_{h-1} & \text{if } h \in \{1,...k-1\} \end{cases}.$$

Thus, the Lipschitz constant of $((T^\delta)^{k-1}T^\pi)f$ is a_{k-1}. By unrolling the recursion we have:

$$\begin{aligned} a_{k-1} &= L_r \sum_{i=0}^{k-1} \gamma^i (L_P+1)^i + \gamma^k L_P(L_\pi+1)(L_P+1)^{k-1}L_f \\ &= L_r \frac{1-\gamma^k(L_P+1)^k}{1-\gamma(L_P+1)} + \gamma^k L_P(L_\pi+1)(L_P+1)^{k-1}L_f. \end{aligned}$$

Let us now consider the sequence b_j of the Lipschitz constants of $(T_k^\pi)^j f$ for $j \in \mathbb{N}$:

$$b_j = \begin{cases} L_f & \text{if } j = 0 \\ L_r \frac{1-\gamma^k(L_P+1)^k}{1-\gamma(L_P+1)} + \gamma^k L_P(L_\pi+1)(L_P+1)^{k-1} b_{j-1} & \text{if } j \in \mathbb{N}_{\geqslant 1} \end{cases}.$$

The sequence b_j converges to a finite limit as long as $\gamma^k L_P(L_\pi+1)(L_P+1)^{k-1} < 1$. In such case, the limit b_∞ can be computed solving the fixed point equation:

$$\begin{aligned} b_\infty &= L_r \frac{1-\gamma^k(L_P+1)^k}{1-\gamma(L_P+1)} + \gamma^k L_P(L_\pi+1)(L_P+1)^{k-1} b_\infty \\ \implies \quad b_\infty &= \frac{L_r\left(1-\gamma^k(L_P+1)^k\right)}{(1-\gamma(L_P+1))\left(1-\gamma^k L_P(L_\pi+1)(L_P+1)^{k-1}\right)}. \end{aligned}$$

Thus, b_∞ represents the Lipschitz constant of Q_k^π.

It is worth noting that when setting $k = 1$ we recover the Lipschitz constant of the Q^π as in (Rachelson and Lagoudakis, 2010). To get a bound that is independent on k we define $L = \max\{L_P(L_\pi+1), L_P+1\}$, assuming that $\gamma L < 1$ so that:

$$b_\infty = \frac{L_r\left(1-\gamma^k(L_P+1)^k\right)}{(1-\gamma(L_P+1))\left(1-\gamma^k L_P(L_\pi+1)(L_P+1)^{k-1}\right)} \leqslant \frac{L_r}{1-\gamma L},$$

having observed that $\frac{1-\gamma^k(L_P+1)^k}{1-\gamma(L_P+1)} \leqslant \frac{1-\gamma^k L^k}{1-\gamma L}$. Thus, we conclude that Q_k^π is also $\frac{L_r}{1-\gamma L}$–LC for any $k \in \mathbb{N}_{\geqslant 1}$. Consider now the application of the operator T^π to Q_k^π, we have that the corresponding Lipschitz constant can be bounded by:

$$L_{T^\pi Q_k^\pi} \leqslant L_r + \gamma L_P(L_\pi+1)\frac{L_r}{1-\gamma L} \leqslant L_r + \gamma L \frac{L_r}{1-\gamma L} = \frac{L_r}{1-\gamma L}. \tag{P.47}$$

A similar derivation holds for the application of T^δ. As a consequence, any arbitrary sequence of applications of T^π and T^δ to Q_k^π generates a sequence of $\frac{L_r}{1-\gamma L}$–LC functions. Even more so for the functions in the set $\mathcal{Q}_k = \{\left(T^\delta\right)^{k-2-l} T^\pi Q_k^\pi : l \in \{0,\dots,k-2\}\}$. As a consequence, we can rephrase the dissimilarity term $d^\pi_{\mathcal{Q}_k^\pi}(s,a)$ as a Kantorovich distance:

$$d^\pi_{\mathcal{Q}_k^\pi}(s,a) = \sup_{f\in\mathcal{Q}_k} \left| \int_{\mathcal{S}}\int_{\mathcal{A}} \left(P^\pi(\mathrm{d}s',\mathrm{d}a'|s,a) - P^\delta(\mathrm{d}s',\mathrm{d}a'|s,a)\right) f(s',a') \right|$$

$$\leqslant L_{\mathcal{Q}_k^\pi} \sup_{f:\|f\|_L \leqslant 1} \left| \int_{\mathcal{S}} \int_{\mathcal{A}} \left(P^\pi(\mathrm{d}s', \mathrm{d}a'|s,a) - P^\delta(\mathrm{d}s', \mathrm{d}a'|s,a) \right) f(s',a') \right|$$
$$= L_{\mathcal{Q}_k^\pi} \mathcal{W}_1 \left(P^\pi(\cdot|s,a), P^\delta(\cdot|s,a) \right).$$

□

APPENDIX $\mathcal{B}$

Exponential Family Policies

In this appendix, we report some results about policies that belong to the exponential family.

B.1 Gaussian and Boltzmann Linear Policies as Exponential Family distributions

We show how a multivariate Gaussian with fixed covariance and a Boltzmann policy, both linear in the state features $\boldsymbol{\phi}(s)$ can be cast into Definition 8.2. We are going to make use of the following identities regarding the Kronecker product (Petersen and Pedersen, 2008):

$$\mathrm{vec}(\mathbf{AXB}) = \left(\mathbf{B}^T \otimes \mathbf{A}\right) \mathrm{vec}(\mathbf{X}) \tag{B.1}$$

$$\mathbf{a}^T\mathbf{XBX}^T\mathbf{c} = \mathrm{vec}(\mathbf{X})^T \left(\mathbf{B} \otimes \mathbf{ca}^T\right) \mathrm{vec}(\mathbf{X}), \tag{B.2}$$

where $\mathrm{vec}(\mathbf{X})$ is the *vectorization* of matrix $\mathbf{X}$ obtained by stacking the columns of $\mathbf{X}$ into a single column vector.

B.1.1 Multivariate Linear Gaussian Policy with fixed covariance

The typical representation of a multivariate linear Gaussian policy is given by the following probability density function:

$$\pi_{\widetilde{\boldsymbol{\theta}}}(\mathbf{a}|s) = \frac{1}{(2\pi)^{\frac{k}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(\mathbf{a}-\widetilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s))^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\widetilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s))\right\},$$

where $\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^{k\times q}$ is a properly sized matrix. Recalling Definition 8.2, we rephrase the previous equation as:

$$\pi_{\widetilde{\boldsymbol{\theta}}}(\mathbf{a}|s) = \frac{1}{(2\pi)^{\frac{k}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\mathbf{a}^T\boldsymbol{\Sigma}^{-1}\mathbf{a}\right\}\exp\left\{\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\mathbf{a}-\frac{1}{2}\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\widetilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s)\right\}.$$

Recalling the identities at Equation (B.1) and (B.2) and observing that $\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\mathbf{a}$ and $\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\widetilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s)$ are scalar, we can rewrite:

$$\begin{aligned}\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\mathbf{a} &= \operatorname{vec}\left(\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\mathbf{a}\right)\\ &= \left(\mathbf{a}^T\boldsymbol{\Sigma}^{-1}\otimes\boldsymbol{\phi}(s)^T\right)\operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)\\ &= \operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)^T\left(\boldsymbol{\Sigma}^{-1}\mathbf{a}\otimes\boldsymbol{\phi}(s)\right),\end{aligned}$$

$$\boldsymbol{\phi}(s)^T\widetilde{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\widetilde{\boldsymbol{\theta}}\boldsymbol{\phi}(s) = \operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)^T\left(\boldsymbol{\Sigma}^{-1}\otimes\boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T\right)\operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right).$$

Now, by redefining the parameter of the exponential family distribution $\boldsymbol{\theta} = \operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)$ we state the following definitions to comply with Definition 8.2:

$$\begin{aligned}&\mathbf{t}(s,\mathbf{a}) = \boldsymbol{\Sigma}^{-1}\mathbf{a}\otimes\boldsymbol{\phi}(s),\\ &h(\mathbf{a}) = \frac{1}{(2\pi)^{\frac{k}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\mathbf{a}^T\boldsymbol{\Sigma}^{-1}\mathbf{a}\right\},\\ &A(\boldsymbol{\theta},s) = \boldsymbol{\theta}^T\left(\boldsymbol{\Sigma}^{-1}\otimes\boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T\right)\boldsymbol{\theta}.\end{aligned}$$

B.1.2 Boltzmann Linear Policy

The Boltzmann policy on a finite set of actions $\{a_1,...,a_{k+1}\}$ is typically represented by means of a matrix of parameters $\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^{k\times q}$:[1]

$$\pi_{\widetilde{\boldsymbol{\theta}}}(a_i|s) = \begin{cases}\frac{\exp\left\{\widetilde{\boldsymbol{\theta}}_i^T\boldsymbol{\phi}(s)\right\}}{1+\sum_{j=1}^{k}\exp\left\{\widetilde{\boldsymbol{\theta}}_j^T\boldsymbol{\phi}(s)\right\}} & \text{if } i\leqslant k\\ \frac{1}{1+\sum_{j=1}^{k}\exp\left\{\widetilde{\boldsymbol{\theta}}_j^T\boldsymbol{\phi}(s)\right\}} & \text{if } i=k+1\end{cases},$$

[1]Notice that we are considering a set made of $k+1$ actions but the matrix $\widetilde{\boldsymbol{\theta}}$ has only k rows. This allows enforcing the identifiability property, otherwise if we had a row for each of the $k+1$ actions we would have multiple representation for the same policy (rescaling the rows by the same amount).

where with $\widetilde{\boldsymbol{\theta}}_i$ we denote the i-th row of matrix $\widetilde{\boldsymbol{\theta}}$. In order to comply to Definition 8.2, we rewrite the density function in the following form:

$$\pi_{\widetilde{\boldsymbol{\theta}}}(a_i|s) = \begin{cases} \exp\left\{\widetilde{\boldsymbol{\theta}}_i^T \boldsymbol{\phi}(s) - \log\left(\exp\{0\} + \sum_{j=1}^k \exp\left\{\widetilde{\boldsymbol{\theta}}_j^T \boldsymbol{\phi}(s)\right\}\right)\right\} & \text{if } i \leqslant k \\ \exp\left\{0 - \log\left(\exp\{0\} + \sum_{j=1}^k \exp\left\{\widetilde{\boldsymbol{\theta}}_j^T \boldsymbol{\phi}(s)\right\}\right)\right\} & \text{if } i = k+1 \end{cases}.$$

By introducing the vector $\mathbf{e}_i$ as the i–th vector of the canonical basis of $\mathbb{R}^k$, i.e., the vector having 1 in the i–th component and 0 elsewhere, and recalling the definition of Kronecker product, we can derive the following identity for $i \leqslant k$:

$$\widetilde{\boldsymbol{\theta}}_i^T \boldsymbol{\phi}(s) = \operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)^T (\mathbf{e}_i \otimes \boldsymbol{\phi}(s)).$$

In the case $i = k$ it is sufficient to replace the previous term with the zero vector $\mathbf{0}$. Therefore, by renaming $\boldsymbol{\theta} = \operatorname{vec}\left(\widetilde{\boldsymbol{\theta}}^T\right)$ we can make the following assignments in order to get the relevant quantities in Definition 8.2:

$$\begin{aligned} &\mathbf{t}(s, a_i) = \begin{cases} \mathbf{e}_i \otimes \boldsymbol{\phi}(s) & \text{if } i \leqslant k \\ \mathbf{0} & \text{if } i = k+1 \end{cases}, \\ &h(a_i) = 1, \\ &A(\boldsymbol{\theta}, s) = \log\left(1 + \sum_{j=1}^k \exp\left\{\boldsymbol{\theta}^T (\mathbf{e}_j \otimes \boldsymbol{\phi}(s))\right\}\right). \end{aligned}$$

B.2 Fisher Information Matrix

We start by providing an expression of the Fisher Information matrix (FIM) for the specific case of the exponential family, that we are going to use extensively in the derivation. We first define the FIM for a fixed state and then we provide its expectation under the state distribution ν. For any state $s \in \mathcal{S}$, we define the FIM induced by $\pi_{\boldsymbol{\theta}}(\cdot|s)$ as:

$$\mathcal{F}(\boldsymbol{\theta}, s) = \mathop{\mathbb{E}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A|s) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A|s)^T\right]. \tag{B.3}$$

We can derive the following immediate result.

Lemma B.1. *For a policy $\pi_{\boldsymbol{\theta}}$ belonging to the exponential family, as in Definition 8.2, the FIM for state $s \in \mathcal{S}$ is given by the covariance matrix of the sufficient statistic:*

$$\mathcal{F}(\boldsymbol{\theta}, s) = \mathop{\mathbb{E}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\bar{\mathbf{t}}(s, A, \boldsymbol{\theta}) \bar{\mathbf{t}}(s, A, \boldsymbol{\theta})^T\right] = \mathop{\mathbb{C}\mathrm{ov}}_{A \sim \pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\mathbf{t}(s, A)\right].$$

Proof. Let us first compute the gradient log-policy for the exponential family:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) &= \mathbf{t}(s, a) - \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}, s) \\ &= \mathbf{t}(s, a) - \frac{\int_{\mathcal{A}} \mathbf{t}(s, \overline{a}) h(\overline{a}) \exp\left\{\boldsymbol{\theta}^T \mathbf{t}(s, \overline{a})\right\} \mathrm{d}\overline{a}}{\int_{\mathcal{A}} h(\overline{a}) \exp\left\{\boldsymbol{\theta}^T \mathbf{t}(s, \overline{a})\right\} \mathrm{d}\overline{a}} \end{aligned} \tag{P.1}$$

$$= \mathbf{t}(s,a) - \mathop{\mathbb{E}}_{\overline{A}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s,\overline{A})\right] = \overline{\mathbf{t}}(s,a,\boldsymbol{\theta}).$$

Now, we just need to apply the definition given in Equation (B.3) and to recall the definition of covariance matrix:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\theta},s) &= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\overline{\mathbf{t}}(s,A,\boldsymbol{\theta})\overline{\mathbf{t}}(s,A,\boldsymbol{\theta})^T\right] \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left(\mathbf{t}(s,A) - \mathop{\mathbb{E}}_{\overline{A}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s,\overline{A})\right]\right)\left(\mathbf{t}(s,A) - \mathop{\mathbb{E}}_{\overline{A}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s,\overline{A})\right]\right)^T\right] \\
&= \mathop{\mathbb{C}\mathrm{ov}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s,A)\right].
\end{aligned}$$

□

We now define the expected FIM $\mathcal{F}(\boldsymbol{\theta})$ and its corresponding estimator $\widehat{\mathcal{F}}(\boldsymbol{\theta})$ under the sampling distribution ν:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{S\sim\nu}\left[\mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|S)}\left[\overline{\mathbf{t}}(S,A)\overline{\mathbf{t}}(s,a)^T\right]\right], \\
\widehat{\mathcal{F}}(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n}\mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\overline{\mathbf{t}}(S_i,A)\overline{\mathbf{t}}(S_i,A)^T\right].
\end{aligned}$$

Finally, we provide a sufficient condition to ensure that the FIM $\mathcal{F}(\boldsymbol{\theta})$ is non singular in the case of Gaussian and Boltzmann linear policies.

Proposition B.2. *If the second moment matrix of the feature vector $\mathbb{E}_{S\sim\nu}\left[\boldsymbol{\phi}(S)\boldsymbol{\phi}(S)^T\right]$ is non–singular, the identifiability condition of Lemma 8.3 is fulfilled by the Gaussian and Boltzmann linear policies for all $\boldsymbol{\theta}\in\Theta$, provided that each action is played with non–zero probability for the Boltzmann policy.*

Proof. Let us start with the Boltzmann policy and consider the expression of $\overline{\mathbf{t}}(s,a_i)$ with $i\in\{1,...,k\}$:

$$\begin{aligned}
\overline{\mathbf{t}}(s,a_i,\boldsymbol{\theta}) &= \mathbf{t}(s,a_i) - \mathop{\mathbb{E}}_{\overline{A}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{t}(s,\overline{A})\right] \\
&= \mathbf{e}_i\otimes\boldsymbol{\phi}(s) - \sum_{j=1}^{k}\pi_{\boldsymbol{\theta}}(a_i|s)\mathbf{e}_i\otimes\boldsymbol{\phi}(s) \\
&= (\mathbf{e}_i - \boldsymbol{\pi})\otimes\boldsymbol{\phi}(s),
\end{aligned}$$

where $\boldsymbol{\pi}$ is a vector defined as $\boldsymbol{\pi} = (\pi_{\boldsymbol{\theta}}(a_1|s),...,\pi_{\boldsymbol{\theta}}(a_k|s))^T$ and we exploited the distributivity of the Kronecker product. While for $i = k+1$, we have $(\mathbf{0}-\boldsymbol{\pi})\otimes\boldsymbol{\phi}(s)$. For the sake of the proof, let us define $\widetilde{\mathbf{e}}_i = \mathbf{e}_i$ if $i\leqslant k$ and $\widetilde{\mathbf{e}}_{k+1} = \mathbf{0}$. Let us compute the FIM:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\overline{\mathbf{t}}(s,A,\boldsymbol{\theta})\overline{\mathbf{t}}(s,A,\boldsymbol{\theta})^T\right] \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left((\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})\otimes\boldsymbol{\phi}(s)\right)\left((\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})\otimes\boldsymbol{\phi}(s)\right)^T\right] \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[(\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})(\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})^T\otimes\boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T\right] \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[(\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})(\widetilde{\mathbf{e}}_i-\boldsymbol{\pi})^T\right]\otimes\boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T
\end{aligned}$$

$$= \left(\mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\widetilde{\mathbf{e}}_i \widetilde{\mathbf{e}}_i^{\,T} \right] - \boldsymbol{\pi}\boldsymbol{\pi}^T \right) \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T$$
$$= \left(\mathrm{diag}(\boldsymbol{\pi}) - \boldsymbol{\pi}\boldsymbol{\pi}^T \right) \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T,$$

where we exploited the distributivity of the Kroneker product, observed that $\mathbb{E}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\widetilde{\mathbf{e}}_i\right] = \boldsymbol{\pi}$ and $\mathbb{E}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\widetilde{\mathbf{e}}_i \widetilde{\mathbf{e}}_i^{\,T} \right] = \mathrm{diag}(\boldsymbol{\pi})$. Let us now consider the matrix:

$$\mathrm{diag}(\boldsymbol{\pi}) - \boldsymbol{\pi}\boldsymbol{\pi}^T = \begin{pmatrix} \pi_{\boldsymbol{\theta}}(a_1|s) - \pi_{\boldsymbol{\theta}}(a_1|s)^2 & -\pi_{\boldsymbol{\theta}}(a_1|s)\pi_{\boldsymbol{\theta}}(a_2|s) & \dots & -\pi_{\boldsymbol{\theta}}(a_1|s)\pi_{\boldsymbol{\theta}}(a_k|s) \\ -\pi_{\boldsymbol{\theta}}(a_1|s)\pi_{\boldsymbol{\theta}}(a_2|s) & \pi_{\boldsymbol{\theta}}(a_2|s) - \pi_{\boldsymbol{\theta}}(a_2|s)^2 & \dots & -\pi_{\boldsymbol{\theta}}(a_2|s)\pi_{\boldsymbol{\theta}}(a_k|s) \\ \vdots & \vdots & \ddots & \vdots \\ -\pi_{\boldsymbol{\theta}}(a_1|s)\pi_{\boldsymbol{\theta}}(a_k|s) & -\pi_{\boldsymbol{\theta}}(a_2|s)\pi_{\boldsymbol{\theta}}(a_k|s) & \dots & \pi_{\boldsymbol{\theta}}(a_k|s) - \pi_{\boldsymbol{\theta}}(a_k|s)^2 \end{pmatrix}.$$

Consider a generic row $i \in \{1, ..., k\}$. The element on the diagonal is $\pi_{\boldsymbol{\theta}}(a_i|s) - \pi_{\boldsymbol{\theta}}(a_i|s)^2 = \pi_{\boldsymbol{\theta}}(a_i|s)\left(1 - \pi_{\boldsymbol{\theta}}(a_i|s)\right)$, while the absolute sum of the elements out of the diagonal is:

$$\pi_{\boldsymbol{\theta}}(a_i|s) \sum_{j\in\{1,\dots k\}\,\wedge\, j\neq i} \pi_{\boldsymbol{\theta}}(a_j|s) = \pi_{\boldsymbol{\theta}}(a_i|s)\left(1 - \pi_{\boldsymbol{\theta}}(a_i|s) - \pi_{\boldsymbol{\theta}}(a_{k+1}|s)\right).$$

Therefore, if all actions are played with non–zero probability, i.e., $\pi_{\boldsymbol{\theta}}(a_i|s) > 0$ for all $i \in \{1, ..., k+1\}$ it follows that the matrix is strictly diagonally dominant by rows and thus it is positive definite. If also $\mathbb{E}_{S\sim\nu}\left[\boldsymbol{\phi}(S)\boldsymbol{\phi}(S)^T\right]$ is positive definite, for the properties of the Kroneker product, the FIM is positive definite.

Let us now focus on the Gaussian policy. Let $\mathbf{a} \in \mathbb{R}^d$ and denote $\boldsymbol{\mu}(s) = \mathbb{E}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}[\mathbf{a}]$:

$$\bar{\mathbf{t}}(s, \mathbf{a}, \boldsymbol{\theta}) = \mathbf{t}(s, \mathbf{a}) - \mathop{\mathbb{E}}_{\bar{\mathbf{a}}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} [\mathbf{t}(s, \bar{\mathbf{a}})] = \boldsymbol{\Sigma}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}(s)\right) \otimes \boldsymbol{\phi}(s).$$

Let us compute the FIM:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\bar{\mathbf{t}}(s, A, \boldsymbol{\theta}) \bar{\mathbf{t}}(s, A, \boldsymbol{\theta})^T \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}(s)\right) \otimes \boldsymbol{\phi}(s) \right) \left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}(s)\right) \otimes \boldsymbol{\phi}(s) \right)^T \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\boldsymbol{\Sigma}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}(s)\right)\left(\mathbf{a} - \boldsymbol{\mu}(s)\right)^T \boldsymbol{\Sigma}^{-1} \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T \right] \\
&= \boldsymbol{\Sigma}^{-1} \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)} \left[\left(\mathbf{a} - \boldsymbol{\mu}(s)\right)\left(\mathbf{a} - \boldsymbol{\mu}(s)\right)^T \right] \boldsymbol{\Sigma}^{-1} \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T \\
&= \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1} \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T = \boldsymbol{\Sigma}^{-1} \otimes \boldsymbol{\phi}(s)\boldsymbol{\phi}(s)^T.
\end{aligned}$$

If $\boldsymbol{\Sigma}$ has finite values, then $\boldsymbol{\Sigma}^{-1}$ will be positive definite and, considering that $\mathbb{E}_{S\sim\nu}\left[\boldsymbol{\phi}(S)\boldsymbol{\phi}(S)^T\right]$ is positive definite, we have that the FIM is positive definite. □

B.3 Subgaussianity Assumption

From Assumption 8.2, we can prove the following result that upper bounds the maximum eigenvalue $\lambda_{\max}$ of the Fisher information matrix with the subgaussianity parameter σ.

Lemma B.3. *Under Assumption 8.2, for any $\boldsymbol{\theta} \in \Theta$ and for any $s \in \mathcal{S}$ the maximum eigenvalue of the Fisher Information matrix $\mathcal{F}(\boldsymbol{\theta}, s)$ is upper bounded by $d\sigma^2$.*

Proof. Recall that the maximum eigenvalue of a matrix $\mathbf{A}$ can be computed as $\sup_{\mathbf{x}:\|\mathbf{x}\|_2\leqslant 1} \mathbf{x}^T\mathbf{A}\mathbf{x}$ and the norm of a vector $\mathbf{y}$ can be computed as $\sup_{\mathbf{x}:\|\mathbf{x}\|_2\leqslant 1} \mathbf{x}^T\mathbf{y}$. Consider now the derivation for a generic $\mathbf{x} \in \mathbb{R}^d$ such that $\|\mathbf{x}\|_2 \leqslant 1$:

$$\begin{aligned}
\mathbf{x}^T\mathcal{F}(\boldsymbol{\theta}, s)\mathbf{x} &= \mathbf{x}^T \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})^T\right]\mathbf{x} \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\mathbf{x}^T\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})^T\mathbf{x}\right] \\
&= \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left(\mathbf{x}^T\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\right)^2\right] \\
&\leqslant \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left(\sup_{\mathbf{x}:\|\mathbf{x}\|_2\leqslant 1} \mathbf{x}^T\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\right)^2\right] = \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left\|\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\right\|_2^2\right],
\end{aligned}$$

where we employed Lemma B.1 and upper bounded the right hand side. By taking the supremum over $\mathbf{x} \in \mathbb{R}^d$ such that $\|\mathbf{x}\|_2 \leqslant 1$ we get:

$$\lambda_{\max}\left(\mathcal{F}(\boldsymbol{\theta}, s)\right) = \sup_{\mathbf{x}:\|\mathbf{x}\|_2\leqslant 1} \mathbf{x}^T\mathcal{F}(\boldsymbol{\theta}, s)\mathbf{x} \leqslant \mathop{\mathbb{E}}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left\|\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\right\|_2^2\right]. \tag{P.2}$$

By applying the first inequality in Remark 2.2 of Hsu et al. (2011) and setting $\mathbf{A} = \mathbf{I}$ we get that $\mathbb{E}_{A\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\left\|\bar{\mathbf{t}}(s, A, \boldsymbol{\theta})\right\|_2^2\right] \leqslant d\sigma^2$. □

We now show that the subgaussianity assumption is satisfied by the Boltzmann and Gaussian policies, as defined in Table 8.1, under mild assumptions.

Proposition B.4. *If the features $\boldsymbol{\phi}$ are uniformly bounded in norm over the state space, i.e., $\Phi_{\max} = \sup_{s\in\mathcal{S}} \|\boldsymbol{\phi}(s)\|_2$, then Assumption 8.2 is fulfilled by the Boltzmann linear policy with parameter $\sigma = 2\Phi_{\max}$ and Gaussian linear policy with parameter $\sigma = \frac{\Phi_{\max}}{\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})}}$.*

Proof. Let us start with the Boltzmann policy. From the definition of subgaussianity given in Assumption 8.2, requiring that the random vector $\bar{\mathbf{t}}(s, a_i, \boldsymbol{\theta})$ is subgaussian with parameter σ is equivalent to require that the random (scalar) variable $\frac{1}{\|\boldsymbol{\alpha}\|_2}\boldsymbol{\alpha}^T\bar{\mathbf{t}}(s, a_i, \boldsymbol{\theta})$ is subgaussian with parameter σ for any $\boldsymbol{\alpha} \in \mathbb{R}^d$. Thus, we now bound the term:

$$\begin{aligned}
\left|\boldsymbol{\alpha}^T\bar{\mathbf{t}}(s, a, \boldsymbol{\theta})\right| &= \left|\boldsymbol{\alpha}^T\left((\widetilde{\mathbf{e}}_i - \boldsymbol{\pi}) \otimes \boldsymbol{\phi}(s)\right)\right| \\
&= \|\boldsymbol{\alpha}\|_2 \left\|(\widetilde{\mathbf{e}}_i - \boldsymbol{\pi}) \otimes \boldsymbol{\phi}(s)\right\|_2 \\
&= \|\boldsymbol{\alpha}\|_2 \left\|\widetilde{\mathbf{e}}_i - \boldsymbol{\pi}\right\|_2 \left\|\boldsymbol{\phi}(s)\right\|_2 \\
&\leqslant 2\|\boldsymbol{\alpha}\|_2 \Phi_{\max},
\end{aligned}$$

where we used Cauchy–Swartz inequality, the identity $\|\mathbf{x}\otimes\mathbf{y}\|_2^2 = (\mathbf{x}\otimes\mathbf{y})^T(\mathbf{x}\otimes\mathbf{y}) = \left(\mathbf{x}^T\mathbf{x}\right)\otimes\left(\mathbf{y}^T\mathbf{y}\right) = \|\mathbf{x}\|_2^2\|\mathbf{y}\|_2^2$ and the inequality $\|\widetilde{\mathbf{e}}_i - \boldsymbol{\pi}\|_2^2 \leqslant 2$. Therefore, we have that the random variable $\frac{1}{\|\boldsymbol{\alpha}\|_2}\boldsymbol{\alpha}^T\bar{\mathbf{t}}(s, a_i, \boldsymbol{\theta}) \leqslant 2\Phi_{\max}$ is bounded. Thanks to Hoeffding's lemma we have that the subgaussianity parameter is $\sigma = 2\Phi_{\max}$.

Let us now consider the Gaussian policy. Let $\mathbf{a} \in \mathbb{R}^d$ and denote with $\boldsymbol{\mu}(s) = \mathbb{E}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}[\mathbf{a}]$:

$$\bar{\mathbf{t}}(s, \mathbf{a}, \boldsymbol{\theta}) = \mathbf{t}(s, \mathbf{a}) - \mathop{\mathbb{E}}_{\bar{\mathbf{a}}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}[\mathbf{t}(s, \bar{\mathbf{a}})] = \boldsymbol{\Sigma}^{-1}\left(\mathbf{a} - \boldsymbol{\mu}(s)\right) \otimes \boldsymbol{\phi}(s).$$

Let us first observe that we can rewrite:

$$\begin{aligned}\boldsymbol{\alpha}^T \left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\otimes\boldsymbol{\phi}(s)\right) &= \sum_{i=1}^{k}\sum_{j=1}^{q}\alpha_{ij}\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\right)_i\phi(s)_j \\ &= \sum_{i=1}^{k}\sum_{j=1}^{q}\alpha_{ij}\phi(s)_j\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\right)_i \\ &= \boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right),\end{aligned}$$

where $\beta_i = \sum_j \alpha_{ij}\phi(s)_j$ for $i \in \{1, ..., k\}$. We now proceed with explicit computations:

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\exp\left\{\boldsymbol{\alpha}^T\bar{\mathbf{t}}(s,\mathbf{a},\boldsymbol{\theta})\right\}\right] &= \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\exp\left\{\boldsymbol{\alpha}^T\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\otimes\boldsymbol{\phi}(s)\right)\right\}\right] \\ &= \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\exp\left\{\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\right\}\right] \\ &= \int_{\mathbb{R}^d}\frac{\exp\left\{-\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu}(s))^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu}(s))\right\}}{(2\pi)^{\frac{k}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\exp\left\{\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{a}-\boldsymbol{\mu}(s)\right)\right\}\mathrm{d}\mathbf{a}.\end{aligned}$$

Now we complete the square:

$$\begin{aligned}&-\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu}(s))^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu}(s)) + \boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu}(s)) \\ &\qquad = -\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu}(s)-\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu}(s)-\boldsymbol{\beta}) + \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}.\end{aligned}$$

Thus, we have:

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathbf{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[\exp\left\{\boldsymbol{\alpha}^T\bar{\mathbf{t}}(s,\mathbf{a},\boldsymbol{\theta})\right\}\right] \\ &\qquad = \exp\left\{\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}\right\}\int_{\mathbb{R}^d}\frac{\exp\left\{-\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu}(s)-\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\mathbf{a}-\boldsymbol{\mu}(s)-\boldsymbol{\beta})\right\}}{(2\pi)^{\frac{k}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}}\mathrm{d}\mathbf{a} \\ &\qquad = \exp\left\{\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}\right\}.\end{aligned}$$

Now, we observe that:

$$\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta} \leqslant \|\boldsymbol{\beta}\|_2^2\left\|\boldsymbol{\Sigma}^{-1}\right\|_2 \leqslant \|\boldsymbol{\alpha}\|_2^2\|\boldsymbol{\phi}(s)\|_2^2\left\|\boldsymbol{\Sigma}^{-1}\right\|_2,$$

having derived from Cauchy–Swartz inequality:

$$\begin{aligned}\|\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^{k}\left(\sum_{j=1}^{q}\alpha_{ij}\phi(s)_j\right)^2 &\leqslant \sum_{i=1}^{k}\sum_{j=1}^{q}\alpha_{ij}^2\sum_{l=1}^{q}\phi(s)_l^2 \\ &= \left(\sum_{i=1}^{k}\sum_{j=1}^{q}\alpha_{ij}^2\right)\sum_{l=1}^{q}\phi(s)_l^2 \\ &= \|\boldsymbol{\alpha}\|_2^2\|\boldsymbol{\phi}(s)\|_2^2.\end{aligned}$$

We get the result by setting $\sigma = \Phi_{\max}\sqrt{\left\|\boldsymbol{\Sigma}^{-1}\right\|_2} = \frac{\Phi_{\max}}{\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})}}$. □

Furthermore, we report for completeness the standard Hoeffding concentration inequality for subgaussian random vectors.

Proposition B.5. *Let $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n$ be n i.i.d. zero–mean subgaussian d–dimensional random vectors with parameter $\sigma \geqslant 0$, then for any $\boldsymbol{\alpha} \in \mathbb{R}^d$ and $\epsilon > 0$ it holds that:*

$$\mathbb{P}\left(\boldsymbol{\alpha}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\right) \geqslant \epsilon\right) \leqslant \exp\left\{-\frac{\epsilon^2 n}{2\left\|\boldsymbol{\alpha}\right\|_2^2 \sigma^2}\right\}.$$

Proof. The proof is analogous to that of the Hoeffding inequality for bounded random variables. Let $s \geqslant 0$:

$$\begin{aligned}
\mathbb{P}\left(\boldsymbol{\alpha}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\right) \geqslant \epsilon\right) &= \mathbb{P}\left(\exp\left\{s\boldsymbol{\alpha}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\right)\right\} \geqslant e^{s\epsilon}\right) \\
&\leqslant e^{-s\epsilon}\,\mathbb{E}\left[\exp\left\{s\boldsymbol{\alpha}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\right)\right\}\right] \\
&= e^{-s\epsilon}\prod_{i=1}^{n}\mathbb{E}\left[\exp\left\{\frac{s}{n}\boldsymbol{\alpha}^T\mathbf{X}_i\right\}\right] \\
&\leqslant e^{-s\epsilon}\exp\left\{\frac{s^2}{2n}\left\|\boldsymbol{\alpha}\right\|_2^2\sigma^2\right\} = \exp\left\{-s\epsilon + \frac{s^2}{2n}\left\|\boldsymbol{\alpha}\right\|_2^2\sigma^2\right\},
\end{aligned}$$

where we employed Markov inequality, exploited the subgaussianity assumption and the independence. We minimize the last expression over s, getting the optimal $s = \frac{\epsilon n}{\left\|\boldsymbol{\alpha}\right\|_2^2\sigma^2}$, from which we get the result:

$$\mathbb{P}\left(\boldsymbol{\alpha}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i\right) \geqslant \epsilon\right) \leqslant \exp\left\{-\frac{\epsilon^2 n}{2\left\|\boldsymbol{\alpha}\right\|_2^2 \sigma^2}\right\}.$$

□

Under the Assumption 8.2, we provide the following concentration inequality for the minimum eigenvalue of the empirical FIM.

Proposition B.6. *Let $\mathcal{F}(\boldsymbol{\theta})$ and $\widehat{\mathcal{F}}(\boldsymbol{\theta})$ be the FIM and its estimate obtained with $n > 0$ independent samples. Then, under Assumption 8.2, for any $\epsilon > 0$ it holds that:*

$$\mathbb{P}\left(\left|\lambda_{\min}\left(\widehat{\mathcal{F}}(\boldsymbol{\theta})\right) - \lambda_{\min}\left(\mathcal{F}(\boldsymbol{\theta})\right)\right| > \epsilon\right) \leqslant 2\exp\left\{-\frac{\epsilon^2 n}{\psi_\sigma d^2 \sigma^4}\right\},$$

where $\psi_\sigma > 0$ is a constant depending only on the subgaussianity parameter σ. In particular, under the following condition on n we have that, for any $\delta \in [0,1]$, $\lambda_{\min}(\widehat{\mathcal{F}}(\boldsymbol{\theta})) > 0$ with probability at least $1-\delta$:

$$n > \frac{d^2\sigma^4\psi_\sigma \log\frac{2}{\delta}}{\lambda_{\min}(\mathcal{F}(\boldsymbol{\theta}))^2}.$$

Proof. Let us recall that $\widehat{\mathcal{F}}(\boldsymbol{\theta})$ and $\mathcal{F}(\boldsymbol{\theta})$ are both symmetric positive semidefinite matrices, thus their eigenvalues λ_j correspond to their singular values σ_j. Let us consider the following sequence of inequalities:

$$\left|\lambda_{\min}\left(\widehat{\mathcal{F}}(\boldsymbol{\theta})\right) - \lambda_{\min}\left(\mathcal{F}(\boldsymbol{\theta})\right)\right| = \left|\sigma_{\min}\left(\widehat{\mathcal{F}}(\boldsymbol{\theta})\right) - \sigma_{\min}\left(\mathcal{F}(\boldsymbol{\theta})\right)\right|$$

$$\leqslant \max_{j\in\{1,\dots,d\}} \left|\sigma_j\left(\widehat{\mathcal{F}}(\boldsymbol{\theta})\right) - \sigma_j\left(\mathcal{F}(\boldsymbol{\theta})\right)\right|$$
$$\leqslant \left\|\widehat{\mathcal{F}}(\boldsymbol{\theta}) - \mathcal{F}(\boldsymbol{\theta})\right\|_2,$$

where last inequality follows from Ben-Israel and Greville (2003). Therefore, all it takes is to bound the norm of the difference. For this purpose, we employ Corollary 5.50 and Remark 5.51 of Vershynin (2012), having observed that the FIM is indeed a covariance matrix and its estimate is a sample covariance matrix. We obtain that with probability at least $1-\delta$:

$$\left\|\widehat{\mathcal{F}}(\boldsymbol{\theta}) - \mathcal{F}(\boldsymbol{\theta})\right\|_2 \leqslant \|\mathcal{F}(\boldsymbol{\theta})\|_2 \sqrt{\frac{\psi_\sigma \log\frac{2}{\delta}}{n}}, \tag{P.3}$$

where $\psi_\sigma \geqslant 0$ is a constant depending on the subgaussianity parameter σ. Recalling, from Lemma B.3, that $\|\mathcal{F}(\boldsymbol{\theta})\| = \lambda_{\max}\left(\mathcal{F}(\boldsymbol{\theta})\right) \leqslant d\sigma^2$, we can rewrite the previous inequality as:

$$\left\|\widehat{\mathcal{F}}(\boldsymbol{\theta}) - \mathcal{F}(\boldsymbol{\theta})\right\|_2 \leqslant d\sigma^2 \sqrt{\frac{\psi_\sigma \log\frac{2}{\delta}}{n}}. \tag{P.4}$$

By setting the right hand side equal to ϵ and solving for δ, we get the first result. The value of n can be obtained by setting the right hand side equal to $\lambda_{\min}(\mathcal{F}(\boldsymbol{\theta}))$. □

Bibliography

Yasin Abbasi-Yadkori, Peter L. Bartlett, and Stephen J. Wright. A fast and reliable policy improvement algorithm. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 1338–1346. JMLR.org, 2016. (Cited on page 90.)

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 2017. (Cited on page 95.)

Alekh Agarwal, Nan Jiang, and Sham M Kakade. Reinforcement learning: Theory and algorithms. Technical report, Technical Report, Department of Computer Science, University of Washington, 2019. (Cited on page 11.)

András Antos, Rémi Munos, and Csaba Szepesvári. Fitted q-iteration in continuous action-space mdps. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 9–16. Curran Associates, Inc., 2007. (Cited on page 38.)

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Mach. Learn.*, 71(1): 89–129, 2008. doi: 10.1007/s10994-007-5038-2. (Cited on pages 38, 39, 188, 192, and 197.)

Søren Asmussen. Further topics in renewal theory and regenerative processes. *Applied Probability and Queues*, pages 186–219, 2003. (Cited on page 18.)

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002. doi: 10.1023/A:1013689704352. (Cited on page 35.)

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1726–1734. AAAI Press, 2017. (Cited on page 198.)

J Andrew Bagnell, Andrew Y Ng, and Jeff G Schneider. Solving uncertain markov decision processes. 2001. (Cited on page 62.)

Leemon C Baird. Reinforcement learning in continuous time: Advantage updating. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 4, pages 2448–2453. IEEE, 1994. (Cited on page 198.)

Leemon C Baird III. Advantage updating. Technical report, Wright Lab Wright-Patterson Afb Oh, 1993. (Cited on page 22.)

Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. math*, 3(1):133–181, 1922. (Cited on pages 23, 58, 73, 82, and 84.)

Bita Banihashemi, Giuseppe De Giacomo, and Yves Lespérance. Online situation-determined agents and their supervision. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pages 517–520. AAAI Press, 2016. (Cited on page 51.)

Bita Banihashemi, Giuseppe De Giacomo, and Yves Lespérance. Hierarchical agent supervision. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 1432–1440. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. (Cited on page 51.)

George A Barnard. Control charts and stochastic processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1959. (Cited on pages 6 and 149.)

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.*, 13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077. (Cited on pages 139 and 199.)

Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998. (Cited on page 80.)

Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. *J. Artif. Intell. Res.*, 15:319–350, 2001. doi: 10.1613/jair.806. (Cited on pages 32, 40, 42, 122, and 123.)

Carolyn L. Beck and R. Srikant. Error bounds for constant step-size q-learning. *Syst. Control. Lett.*, 61(12):1203–1208, 2012. doi: 10.1016/j.sysconle.2012.08.014. (Cited on page 36.)

Richard Bellman. The theory of dynamic programming. Technical report, Rand corp santa monica ca, 1954. (Cited on page 13.)

Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN 9780486428093. (Cited on pages 7, 12, 13, 23, and 27.)

Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003. (Cited on page 251.)

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM, 2009. doi: 10.1145/1553374.1553380. (Cited on pages 50 and 66.)

Dimitir P Bertsekas and Steven Shreve. *Stochastic optimal control: the discrete-time case*. 2004. (Cited on page 26.)

Dimitri P. Bertsekas. *Dynamic programming and optimal control, 3rd Edition*. Athena Scientific, 2005. ISBN 1886529264. (Cited on pages 13, 37, and 198.)

Dimitri P Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011. (Cited on page 39.)

Dimitri P Bertsekas and Sergey Ioffe. Temporal differences-based policy iteration and applications in neuro-dynamic programming. *Lab. for Info. and Decision Systems Report LIDS-P-2349, MIT, Cambridge, MA*, 14, 1996. (Cited on page 37.)

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-dynamic programming*, volume 3 of *Optimization and neural computation series*. Athena Scientific, 1996. ISBN 1886529108. (Cited on pages 13, 23, and 38.)

Dimitri P Bertsekas, Vivek S Borkar, and Angelia Nedic. Improved temporal difference methods with linear function approximation. *Learning and Approximate Dynamic Programming*, pages 231–255, 2004. (Cited on page 37.)

Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. ISBN 9780387310732. (Cited on pages 2, 11, and 37.)

Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4583–4589. ijcai.org, 2020. doi: 10.24963/ijcai.2020/632. (Cited on page 20.)

David Blackwell. Discounted dynamic programming. *The Annals of Mathematical Statistics*, 36(1): 226–235, 1965. (Cited on pages 13 and 26.)

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowl. Based Syst.*, 46:109–132, 2013. doi: 10.1016/j.knosys.2013.03.012. (Cited on page 2.)

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN 978-0-19-953525-5. doi: 10.1093/acprof:oso/9780199535255.001.0001. (Cited on page 232.)

Bruce Lee Bowerman. Nonstationary markov decision processes and related topics in nonstationary markov chains. 1974. (Cited on pages 49, 61, and 63.)

Justin A. Boyan. Technical update: Least-squares temporal difference learning. *Mach. Learn.*, 49 (2-3):233–246, 2002. doi: 10.1023/A:1017936530646. (Cited on page 37.)

Justin A. Boyan and Andrew W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 369–376. MIT Press, 1994. (Cited on page 37.)

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. (Cited on page 127.)

Steven J. Bradtke and Andrew G. Barto. Linear least-squares algorithms for temporal difference learning. *Mach. Learn.*, 22(1-3):33–57, 1996. doi: 10.1023/A:1018056104778. (Cited on pages 37 and 39.)

Steven J. Bradtke and Michael O. Duff. Reinforcement learning methods for continuous-time markov decision problems. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 393–400. MIT Press, 1994. (Cited on pages 171 and 198.)

Ronen I. Brafman and Moshe Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3:213–231, 2002. (Cited on page 35.)

Kianté Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. (Cited on page 162.)

Michele Breton, Abderrahmane Alj, and Alain Haurie. Sequential stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications*, 59(1):71–97, 1988. (Cited on page 83.)

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. (Cited on page 199.)

Lawrence D Brown. Fundamentals of statistical exponential families: with applications in statistical decision theory. Ims, 1986. (Cited on page 155.)

Thiago P. Bueno, Denis Deratani Mauá, Leliane N. de Barros, and Fábio Gagliardi Cozman. Modeling markov decision processes with imprecise probabilities using probabilistic logic programming. In Alessandro Antonucci, Giorgio Corani, Inés Couso, and Sébastien Destercke, editors, *Proceedings of the Tenth International Symposium on Imprecise Probability: Theories and Applications, Lugano, Switzerland, 10-14 July 2017*, volume 62 of *Proceedings of Machine Learning Research*, pages 49–60. PMLR, 2017. (Cited on page 61.)

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*, 2008. (Cited on pages 81 and 84.)

George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002. (Cited on pages 6, 149, 150, and 159.)

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. ISBN 978-0-521-84108-5. doi: 10.1017/CBO9780511546921. (Cited on page 11.)

Nicolò Cesa-Bianchi, Claudio Gentile, Gergely Neu, and Gábor Lugosi. Boltzmann exploration done right. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6284–6293, 2017. (Cited on page 35.)

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *CoRR*, abs/1810.00069, 2018. (Cited on page 66.)

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009. doi: 10.1145/1541880.1541882. (Cited on page 2.)

Samuel P. M. Choi, Dit-Yan Yeung, and Nevin Lianwen Zhang. Hidden-mode markov decision processes for nonstationary sequential decision making. In Ron Sun and C. Lee Giles, editors, *Sequence Learning - Paradigms, Algorithms, and Applications*, volume 1828 of *Lecture Notes in Computer Science*, pages 264–287. Springer, 2001. doi: 10.1007/3-540-44565-X_12. (Cited on page 64.)

Kamil Andrzej Ciosek and Shimon Whiteson. OFFER: off-environment reinforcement learning. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1819–1825. AAAI Press, 2017. (Cited on pages 50 and 66.)

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In Joan Feigenbaum, John C.-I. Chuang, and David M. Pennock, editors, *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, pages 82–90. ACM, 2006. doi: 10.1145/1134707.1134717. (Cited on page 82.)

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 442–450. Curran Associates, Inc., 2010. (Cited on pages 44 and 135.)

Corinna Cortes, Spencer Greenberg, and Mehryar Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *Ann. Math. Artif. Intell.*, 85(1):45–70, 2019. doi: 10.1007/s10472-018-9613-y. (Cited on pages 135 and 227.)

Rémi Coulom. *Reinforcement Learning Using Neural Networks, with Applications to Motor Control. (Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur)*. PhD thesis, Grenoble Institute of Technology, France, 2002. (Cited on page 199.)

Christian Daniel, Gerhard Neumann, and Jan Peters. Hierarchical relative entropy policy search. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 273–281. JMLR.org, 2012. (Cited on pages 43, 44, 45, and 125.)

Peter Dayan and Satinder P. Singh. Improving policies without measuring merits. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pages 1059–1065. MIT Press, 1995. (Cited on page 198.)

Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 465–472. Omnipress, 2011. (Cited on pages 31, 136, and 147.)

Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013. doi: 10.1561/2300000021. (Cited on pages 7, 40, 147, and 156.)

Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Oper. Res.*, 58(1):203–213, 2010. doi: 10.1287/opre.1080.0685. (Cited on page 63.)

Karina Valdivia Delgado, Leliane Nunes de Barros, Fabio Gagliardi Cozman, and Ricardo Shirota. Representing and solving factored markov decision processes with imprecise probabilities. *Proceedings ISIPTA, Durham, United Kingdom*, 18, 2009. (Cited on page 61.)

Thomas G. Dietterich. The MAXQ method for hierarchical reinforcement learning. In Jude W. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 118–126. Morgan Kaufmann, 1998. (Cited on page 198.)

Pierluca D'Oro, Alberto Maria Metelli, Andrea Tirinzoni, Matteo Papini, and Marcello Restelli. Gradient-aware model-based policy search. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3801–3808. AAAI Press, 2020. (Cited on pages 19 and 32.)

Kenji Doya. Temporal difference learning in continuous time and space. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pages 1073–1079. MIT Press, 1995. (Cited on page 12.)

Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1): 219–245, 2000. doi: 10.1162/089976600300015961. (Cited on pages 171 and 198.)

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1329–1338. JMLR.org, 2016. (Cited on page 32.)

Lester E Dubins, Leonard J Savage, William Sudderth, and David Gilat. *How to gamble if you must: Inequalities for stochastic processes*. Courier Corporation, 2014. (Cited on page 13.)

Rick Durrett. *Probability: Theory and Examples, 4th Edition*. Cambridge University Press, 2010. ISBN 9780511779398. doi: 10.1017/CBO9780511779398. (Cited on page 14.)

Evgeniui Borisovich Dynkin, Aleksandr Adol'fovich IUshkevich, and AA Yushkevich. *Controlled markov processes*, volume 235. Springer, 1979. (Cited on page 26.)

Zackory M. Erickson, Vamsee Gangaram, Ariel Kapusta, C. Karen Liu, and Charles C. Kemp. Assistive gym: A physics simulation framework for assistive robotics. *CoRR*, abs/1910.04700, 2019. (Cited on page 199.)

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, 2005. (Cited on pages 6, 32, 38, 173, and 186.)

Eyal Even-Dar and Yishay Mansour. Learning rates for q-learning. *J. Mach. Learn. Res.*, 5:1–25, 2003. (Cited on page 36.)

Amir Massoud Farahmand. *Regularization in Reinforcement Learning*. PhD thesis, University of Alberta, 2011. (Cited on pages 38, 39, 186, 188, 189, 192, and 194.)

Amir Massoud Farahmand and Csaba Szepesvári. Model selection in reinforcement learning. *Mach. Learn.*, 85(3):299–332, 2011. doi: 10.1007/s10994-011-5254-7. (Cited on page 197.)

Amir Massoud Farahmand, Daniel Nikolaev Nikovski, Yuji Igarashi, and Hiroki Konaka. Truncated approximate dynamic programming with task-dependent terminal value. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3123–3129. AAAI Press, 2016. (Cited on page 175.)

Tanner Fiez, Benjamin Chasnov, and Lillian J. Ratliff. Convergence of learning dynamics in stackelberg games. *CoRR*, abs/1906.01217, 2019. (Cited on page 210.)

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. (Cited on page 160.)

Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922. (Cited on page 41.)

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pages 482–495. PMLR, 2017. (Cited on page 50.)

Thomas Furmston and David Barber. A unifying perspective of parametric policy search methods for markov decision processes. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2726–2734, 2012. (Cited on page 41.)

Víctor Gallego, Roi Naveiro, and David Ríos Insua. Reinforcement learning under threats. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9939–9940. AAAI Press, 2019a. doi: 10.1609/aaai.v33i01.33019939. (Cited on page 66.)

Víctor Gallego, Roi Naveiro, David Ríos Insua, and David Gómez-Ullate. Opponent aware reinforcement learning. *CoRR*, abs/1908.08773, 2019b. (Cited on page 66.)

Alfredo Garcia and Robert L. Smith. Solving nonstationary infinite horizon stochastic production planning problems. *Oper. Res. Lett.*, 27(3):135–141, 2000. doi: 10.1016/S0167-6377(00)00049-3. (Cited on page 49.)

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16:1437–1480, 2015. (Cited on pages 40 and 89.)

Aurélien Garivier and Emilie Kaufmann. Non-asymptotic sequential tests for overlapping hypotheses and application to near optimal arm identification in bandit models. *arXiv preprint arXiv:1905.03495*, 2019. (Cited on pages 159 and 235.)

Izrail Moiseevitch Gelfand and Richard A Silverman. *Calculus of variations*. Courier Corporation, 2000. (Cited on page 127.)

Alborz Geramifard, Christoph Dann, Robert H. Klein, William Dabney, and Jonathan P. How. Rlpy: a value-function-based reinforcement learning framework for education and research. *J. Mach. Learn. Res.*, 16:1573–1578, 2015. (Cited on page 199.)

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1. (Cited on page 199.)

Zoubin Ghahramani. Unsupervised learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176 of *Lecture Notes in Computer Science*, pages 72–112. Springer, 2003. doi: 10.1007/978-3-540-28650-9_5. (Cited on page 2.)

Mohammad Ghavamzadeh, Sridhar Mahadevan, and Rajbala Makar. Hierarchical multi-agent reinforcement learning. *Auton. Agents Multi Agent Syst.*, 13(2):197–229, 2006. doi: 10.1007/s10458-006-7035-4. (Cited on page 4.)

Mohammad Ghavamzadeh, Marek Petrik, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2298–2306, 2016. (Cited on page 90.)

Giuseppe De Giacomo, Yves Lespérance, and Christian J. Muise. On supervising agents in situation-determined congolog. In Wiebe van der Hoek, Lin Padgham, Vincent Conitzer, and Michael Winikoff, editors, *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pages 1031–1038. IFAAMAS, 2012. (Cited on page 51.)

Robert Givan, Sonia M. Leach, and Thomas L. Dean. Bounded parameter markov decision processes. In Sam Steel and Rachid Alami, editors, *Recent Advances in AI Planning, 4th European Conference on Planning, ECP'97, Toulouse, France, September 24-26, 1997, Proceedings*, volume 1348 of *Lecture Notes in Computer Science*, pages 234–246. Springer, 1997. doi: 10.1007/3-540-63912-8_89. (Cited on pages 49 and 62.)

Gene H Golub and Charles F Van Loan. Matrix computations johns hopkins university press. *Baltimore and London*, 1996. (Cited on page 28.)

Steven N Goodman. Toward evidence-based medical statistics. 1: The p value fallacy. *Annals of internal medicine*, 130(12):995–1004, 1999. (Cited on page 155.)

Geoffrey J. Gordon. Stable function approximation in dynamic programming. In Armand Prieditis and Stuart J. Russell, editors, *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 261–268. Morgan Kaufmann, 1995. doi: 10.1016/b978-1-55860-377-6.50040-2. (Cited on page 37.)

Vineet Goyal and Julien Grand-Clement. Robust markov decision process: Beyond rectangularity. *arXiv preprint arXiv:1811.00215*, 2018. (Cited on page 63.)

Shixiang Gu, Ethan Holly, Timothy P. Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3389–3396. IEEE, 2017. doi: 10.1109/ICRA.2017.7989385. (Cited on page 3.)

László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer series in statistics. Springer, 2002. ISBN 978-0-387-95441-7. doi: 10.1007/b97848. (Cited on pages 37 and 197.)

Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. In Antonio Bicchi, Hadas Kress-Gazit, and Seth Hutchinson, editors, *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, 2019. doi: 10.15607/RSS.2019.XV.011. (Cited on page 3.)

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848570. doi: 10.1007/978-0-387-84858-7. (Cited on page 2.)

Moshe Haviv and Ludo Van der Heyden. Perturbation bounds for the stationary probabilities of a finite markov chain. *Advances in Applied Probability*, pages 804–818, 1984. (Cited on page 96.)

Verena Heidrich-Meisner and Christian Igel. Neuroevolution strategies for episodic reinforcement learning. *J. Algorithms*, 64(4):152–168, 2009. doi: 10.1016/j.jalgor.2009.04.002. (Cited on page 40.)

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4565–4573, 2016. (Cited on pages 162 and 167.)

John H. Holland and David E. Goldberg. Genetic algorithms in search, optimization and machine learning. *Massachusetts: Addison-Wesley*, 1989. (Cited on page 99.)

Wallace J. Hopp, James C. Bean, and Robert L. Smith. A new optimality criterion for nonhomogeneous markov decision processes. *Oper. Res.*, 35(6):875–883, 1987. doi: 10.1287/opre.35.6.875. (Cited on page 49.)

Ronald A Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960. (Cited on page 27.)

Ronald A Howard. Semi-markov decision-processes. *Bulletin of the International Statistical Institute*, 40(2):625–652, 1963. (Cited on page 198.)

Daniel J. Hsu, Sham M. Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *CoRR*, abs/1110.2842, 2011. (Cited on page 248.)

Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. (Cited on page 66.)

Chelsea C. White III and Hany K. Eldeib. Markov decision processes with imprecise transition probabilities. *Oper. Res.*, 42(4):739–749, 1994. doi: 10.1287/opre.42.4.739. (Cited on page 61.)

Garud N. Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30(2):257–280, 2005. doi: 10.1287/moor.1040.0129. (Cited on pages 49 and 62.)

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, 11:1563–1600, 2010. (Cited on pages 32 and 35.)

Harold Jeffreys. Some tests of significance, treated by the theory of probability. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 203–222. Cambridge University Press, 1935. (Cited on page 154.)

Nan Jiang, Alex Kulesza, Satinder P. Singh, and Richard L. Lewis. The dependence of effective planning horizon on model accuracy. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4180–4189. IJCAI/AAAI Press, 2016. (Cited on page 175.)

Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I. Jordan. Is q-learning provably efficient? In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4868–4878, 2018. (Cited on page 35.)

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*, 2019. (Cited on page 210.)

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 2137–2143. PMLR, 2020. (Cited on page 35.)

Ian T. Jolliffe. Principal component analysis. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*, pages 1094–1096. Springer, 2011. doi: 10.1007/978-3-642-04898-2_455. (Cited on page 157.)

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X. (Cited on page 63.)

Sham M. Kakade. A natural policy gradient. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 1531–1538. MIT Press, 2001. (Cited on page 41.)

Sham M. Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003. (Cited on pages 105 and 107.)

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002. (Cited on pages 5, 39, 90, 91, 92, 96, 97, and 121.)

Michael J. Kearns and Satinder P. Singh. Bias-variance error bounds for temporal difference updates. In Nicolò Cesa-Bianchi and Sally A. Goldman, editors, *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT 2000), June 28 - July 1, 2000, Palo Alto, California, USA*, pages 142–147. Morgan Kaufmann, 2000. (Cited on pages 33 and 34.)

Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Mach. Learn.*, 49(2-3):209–232, 2002. doi: 10.1023/A:1017984413808. (Cited on page 35.)

Sarah Keren, Luis Enrique Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Equi-reward utility maximizing design in stochastic environments. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4353–4360. ijcai.org, 2017. doi: 10.24963/ijcai.2017/608. (Cited on pages 61 and 65.)

Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Relaxed modification heuristics for equi-reward utility maximizing design. *HSDIP 2018*, page 1, 2018. (Cited on page 65.)

Sarah Keren, Luis Enrique Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Efficient heuristic search for optimal environment redesign. In J. Benton, Nir Lipovetzky, Eva Onaindia, David E. Smith, and Siddharth Srivastava, editors, *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2018, Berkeley, CA, USA, July 11-15, 2019*, pages 246–254. AAAI Press, 2019. (Cited on page 65.)

Daniel Kikuti, Fabio G Cozman, and Cassio P De Campos. Partially ordered preferences in decision trees: computing strategies with imprecision in probabilities. In *IJCAI workshop on advances in preference handling*, pages 118–123. Citeseer, 2005. (Cited on page 61.)

Bangalore Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Kumar Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *CoRR*, abs/2002.00444, 2020. (Cited on page 3.)

Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004. (Cited on page 198.)

Jonathan Ko, Daniel J. Klein, Dieter Fox, and Dirk Hähnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*, pages 742–747. IEEE, 2007. doi: 10.1109/ROBOT.2007.363075. (Cited on page 40.)

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 849–856. Curran Associates, Inc., 2008. (Cited on page 40.)

Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *I. J. Robotics Res.*, 32(11):1238–1274, 2013. doi: 10.1177/0278364913495721. (Cited on page 172.)

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1008–1014. The MIT Press, 1999. (Cited on page 32.)

Jan Koutník, Giuseppe Cuccu, Jürgen Schmidhuber, and Faustino Gomez. Evolving large-scale neural networks for vision-based reinforcement learning. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 1061–1068, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1963-8. (Cited on page 142.)

Michail G. Lagoudakis and Ronald Parr. Least-squares policy iteration. *J. Mach. Learn. Res.*, 4: 1107–1149, 2003. (Cited on pages 37 and 39.)

Aravind S. Lakshminarayanan, Sahil Sharma, and Balaraman Ravindran. Dynamic action repetition for deep reinforcement learning. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2133–2139. AAAI Press, 2017. (Cited on page 173.)

Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In Marco A. Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer, 2012. doi: 10.1007/978-3-642-27645-3_2. (Cited on pages 32 and 172.)

Tor Lattimore and Rémi Munos. Bounded regret for finite-armed structured bandits. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 550–558, 2014. (Cited on page 210.)

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. (Cited on pages 12, 209, and 210.)

Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Reinforcement learning in continuous action spaces through sequential monte carlo methods. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 833–840. Curran Associates, Inc., 2007. (Cited on page 167.)

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of classification-based policy iteration algorithms. *J. Mach. Learn. Res.*, 17:19:1–19:30, 2016. (Cited on page 39.)

Yann Le Tallec. *Robust, risk-sensitive, and data-driven control of Markov decision processes*. PhD thesis, Massachusetts Institute of Technology, 2007. (Cited on page 63.)

Erwan Lecarpentier and Emmanuel Rachelson. Non-stationary markov decision processes, a worst-case approach using model-based reinforcement learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7214–7223, 2019. (Cited on pages 63 and 64.)

Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Maximum causal tsallis entropy imitation learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4408–4418, 2018. (Cited on pages 162 and 167.)

Boris Lesner and Bruno Scherrer. Non-stationary approximate modified policy iteration. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1567–1575. JMLR.org, 2015. (Cited on page 195.)

Sergey Levine and Vladlen Koltun. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1–9. JMLR.org, 2013. (Cited on pages 122 and 147.)

Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction. *CoRR*, abs/2006.03041, 2020. (Cited on page 36.)

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2071–2080. PMLR, 2017. (Cited on page 231.)

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. (Cited on pages 32 and 142.)

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3756–3762. ijcai.org, 2017. doi: 10.24963/ijcai.2017/525. (Cited on page 66.)

Mark P Little, Wolfgang F Heidenreich, and Guangquan Li. Parameter identifiability and redundancy: theoretical considerations. *PloS one*, 5(1):e8915, 2010. (Cited on page 157.)

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, pages 157–163. Morgan Kaufmann, 1994. doi: 10.1016/b978-1-55860-335-6.50027-1. (Cited on page 80.)

Michael L. Littman. *Algorithms for sequential decision making*. Brown University Providence, RI, 1996. (Cited on page 29.)

Daniele Loiacono, Alessandro Prete, Pier Luca Lanzi, and Luigi Cardamone. Learning to overtake in TORCS using simple reinforcement learning. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*, pages 1–8. IEEE, 2010. doi: 10.1109/CEC.2010.5586191. (Cited on pages 6 and 142.)

Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. Simulated car racing championship: Competition software manual. *CoRR*, abs/1304.1672, 2013. (Cited on page 142.)

Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007. (Cited on page 2.)

David G Luenberger. Introduction to dynamic systems; theory, models, and applications. Technical report, 1979. (Cited on page 171.)

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 14543–14553, 2019. (Cited on pages 66 and 67.)

Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pages 1204–1212. Curran Associates, Inc., 2009. (Cited on page 37.)

MM Hassan Mahmud, Benjamin Rosman, Subramanian Ramamoorthy, and Pushmeet Kohli. Adapting interaction environments to diverse users through online action set selection. In *Proceedings of the aaai 2014 workshop on machine learning for interactive systems*. Citeseer, 2014. (Cited on page 65.)

Giorgio Manganini, Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Following newton direction in policy gradient with parameter exploration. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–8. IEEE, 2015. doi: 10.1109/IJCNN.2015.7280673. (Cited on page 41.)

Timothy A. Mann, Shie Mannor, and Doina Precup. Approximate value iteration with temporally extended actions. *J. Artif. Intell. Res.*, 53:375–438, 2015. doi: 10.1613/jair.4676. (Cited on page 198.)

Shie Mannor, Ofir Mebel, and Huan Xu. Robust mdps with k-rectangular uncertainty. *Math. Oper. Res.*, 41(4):1484–1509, 2016. doi: 10.1287/moor.2016.0786. (Cited on page 63.)

Yishay Mansour and Satinder P. Singh. On the complexity of policy iteration. In Kathryn B. Laskey and Henri Prade, editors, *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999*, pages 401–408. Morgan Kaufmann, 1999. (Cited on page 28.)

José-Luis Menaldi. Controlled markov processes and viscosity solutions (wendell h. fleeting and h. mete soner). *SIAM Review*, 36(1):133–134, 1994. doi: 10.1137/1036035. (Cited on page 198.)

Alberto Maria Metelli, Matteo Pirotta, and Marcello Restelli. Compatible reward inverse reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2050–2059, 2017. (Cited on page 163.)

Alberto Maria Metelli, Mirco Mutti, and Marcello Restelli. Configurable markov decision processes. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3488–3497. PMLR, 2018a. (Cited on pages 7, 8, 9, 50, 54, 89, 97, 98, 108, 123, 132, and 171.)

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 5447–5459, 2018b. (Cited on pages 32, 43, 44, 45, 46, 90, 95, 160, and 161.)

Alberto Maria Metelli, Emanuele Ghelfi, and Marcello Restelli. Reinforcement learning in configurable continuous environments. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4546–4555. PMLR, 2019a. (Cited on pages 8, 9, 132, 135, and 136.)

Alberto Maria Metelli, Amarildo Likmeta, and Marcello Restelli. Propagating uncertainty in reinforcement learning via wasserstein barycenters. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 4335–4347, 2019b. (Cited on page 35.)

Alberto Maria Metelli, Guglielmo Manneschi, and Marcello Restelli. Policy space identification in configurable environments. *CoRR*, abs/1909.03984, 2019c. (Cited on pages 8, 9, and 163.)

Alberto Maria Metelli, Flavio Mazzolini, Lorenzo Bisi, Luca Sabbioni, and Marcello Restelli. Control frequency adaptation via action persistence in batch reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6862–6873. PMLR, 2020a. (Cited on pages 8, 9, 178, 186, 197, and 199.)

Alberto Maria Metelli, Matteo Papini, Nico Montali, and Marcello Restelli. Importance sampling techniques for policy optimization. *Journal of Machine Learning Research*, 21(141):1–75, 2020b. (Cited on pages 43, 44, 45, and 161.)

Alberto Maria Metelli, Matteo Pirotta, and Marcello Restelli. On the use of the policy gradient and hessian in inverse reinforcement learning. *Intelligenza Artificiale*, 14(1):91–124, 2020c. (Cited on page 163.)

Richard Meyes, Hasan Tercan, Simon Roggendorf, Thomas Thiele, Christian Büscher, Markus Obdenbusch, Christian Brecher, Sabina Jeschke, and Tobias Meisen. Motion planning for industrial robots using reinforcement learning. *Procedia CIRP*, 63:107–112, 2017. (Cited on page 3.)

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Communications and Control Engineering Series. Springer, 1993. ISBN 978-1-4471-3269-1. doi: 10.1007/978-1-4471-3267-7. (Cited on page 17.)

Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997. ISBN 978-0-07-042807-2. (Cited on pages 1 and 11.)

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. (Cited on page 3.)

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. (Cited on pages 3 and 31.)

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org, 2016. (Cited on page 142.)

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012. ISBN 978-0-262-01825-8. (Cited on pages 2 and 135.)

Andrew William Moore. Efficient memory based learning for robot control. *PhD Thesis, Computer Laboratory, University of Cambridge*, 1991. (Cited on page 199.)

Richard D Morey, Jan-Willem Romeijn, and Jeffrey N Rouder. The philosophy of bayes factors and the quantification of statistical evidence. *Journal of Mathematical Psychology*, 72:6–18, 2016. (Cited on page 155.)

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, pages 429–443, 1997. (Cited on page 180.)

Rémi Munos. A convergent reinforcement learning algorithm in the continuous case based on a finite difference method. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, pages 826–831. Morgan Kaufmann, 1997. (Cited on page 198.)

Rémi Munos. Error bounds for approximate policy iteration. In Tom Fawcett and Nina Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 560–567. AAAI Press, 2003. (Cited on page 7.)

Rémi Munos. Error bounds for approximate value iteration. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1006–1011. AAAI Press / The MIT Press, 2005. (Cited on pages 32 and 37.)

Rémi Munos. Performance bounds in lp-norm for approximate value iteration. *SIAM J. Control. Optim.*, 46(2):541–561, 2007. doi: 10.1137/040614384. (Cited on pages 74 and 190.)

Rémi Munos and Paul Bourgine. Reinforcement learning for continuous stochastic control problems. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*, pages 1029–1035. The MIT Press, 1997. (Cited on pages 171 and 198.)

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *J. Mach. Learn. Res.*, 9:815–857, 2008. (Cited on page 38.)

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 7559–7566. IEEE, 2018. doi: 10.1109/ICRA.2018.8463189. (Cited on pages 31 and 136.)

John Nash. Non-cooperative games. *Annals of Mathematics, 54 (2), September, 286-295*, 1951. (Cited on page 52.)

Angelia Nedic and Dimitri P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discret. Event Dyn. Syst.*, 13(1-2):79–110, 2003. doi: 10.1023/A:1022192903948. (Cited on page 37.)

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *CoRR*, abs/1705.07798, 2017. (Cited on page 162.)

Gerhard Neumann. Variational inference for policy search in changing situations. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 817–824. Omnipress, 2011. (Cited on page 40.)

Andrew Y. Ng and Michael I. Jordan. PEGASUS: A policy search method for large mdps and pomdps. In Craig Boutilier and Moisés Goldszmidt, editors, *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA, June 30 - July 3, 2000*, pages 406–415. Morgan Kaufmann, 2000. (Cited on page 40.)

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 663–670. Morgan Kaufmann, 2000. (Cited on page 163.)

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 278–287. Morgan Kaufmann, 1999. (Cited on page 209.)

Yaodong Ni and Zhi-Qiang Liu. Bounded-parameter partially observable markov decision processes. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric A. Hansen, editors, *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, September 14-18, 2008*, pages 240–247. AAAI, 2008. (Cited on page 62.)

Arnab Nilim and Laurent El Ghaoui. Robustness in markov decision problems with uncertain transition matrices. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 839–846. MIT Press, 2003. (Cited on pages 49, 50, 52, 60, 61, and 62.)

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018. doi: 10.1561/2300000053. (Cited on pages 6, 148, and 162.)

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013. (Cited on pages 43, 130, 167, and 223.)

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, 1987. doi: 10.1287/moor.12.3.441. (Cited on page 72.)

Matteo Papini, Matteo Pirotta, and Marcello Restelli. Adaptive batch size for safe policy gradients. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3591–3600, 2017. (Cited on pages 40, 41, and 90.)

Matteo Papini, Alberto Maria Metelli, Lorenzo Lupo, and Marcello Restelli. Optimistic policy optimization via multiple importance sampling. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4989–4999. PMLR, 2019a. (Cited on page 44.)

Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients. *CoRR*, abs/1905.03231, 2019b. (Cited on page 43.)

Matteo Papini, Andrea Battistello, and Marcello Restelli. Balancing learning speed and stability in policy gradient via adaptive exploration. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1188–1199. PMLR, 2020. (Cited on page 90.)

Jing Peng and Ronald J. Williams. Incremental multi-step q-learning. *Mach. Learn.*, 22(1-3):283–290, 1996. doi: 10.1023/A:1018076709321. (Cited on page 36.)

Julien Pérolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning nash equilibrium for general-sum markov games from batch data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 232–241. PMLR, 2017. (Cited on pages 77 and 79.)

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008. doi: 10.1016/j.neunet.2008.02.003. (Cited on pages 41, 42, 43, 122, and 147.)

Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Natural actor-critic. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings*, volume 3720 of *Lecture Notes in Computer Science*, pages 280–291. Springer, 2005. doi: 10.1007/11564096_29. (Cited on page 41.)

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010. (Cited on pages 6, 43, 44, 90, 121, and 125.)

Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008. (Cited on page 243.)

James K Peterson. On-line estimation of the optimal value function: Hjb-estimators. In *Advances in Neural Information Processing Systems*, pages 319–326, 1993. (Cited on page 198.)

Marek Petrik and Bruno Scherrer. Biasing approximate dynamic programming with a lower discount factor. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1265–1272. Curran Associates, Inc., 2008. (Cited on page 175.)

Matteo Pirotta and Marcello Restelli. Inverse reinforcement learning through policy gradient minimization. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1993–1999. AAAI Press, 2016. (Cited on page 163.)

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 1394–1402, 2013a. (Cited on pages 32, 40, 43, 92, 95, and 107.)

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 307–315. JMLR.org, 2013b. (Cited on pages 5, 39, 90, 95, 97, 99, 100, 103, 107, and 121.)

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Mach. Learn.*, 100(2-3):255–283, 2015. doi: 10.1007/s10994-015-5484-1. (Cited on page 182.)

Warren Buckler Powell. *Approximate Dynamic Programming - Solving the Curses of Dimensionality*. Wiley, 2007. ISBN 978-0-470-17155-4. doi: 10.1002/9780470182963. (Cited on page 37.)

Doina Precup. *Temporal abstraction in reinforcement learning.* PhD thesis, University of Massachusetts Amherst, 2001. (Cited on page 198.)

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 759–766. Morgan Kaufmann, 2000. (Cited on page 45.)

Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 67–72. IEEE, 2017. (Cited on page 2.)

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. (Cited on pages 5, 7, 11, 12, 13, 16, 21, 22, 23, 24, 25, 26, 27, 28, 49, 73, and 171.)

Martin L Puterman and Moon Chirl Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978. (Cited on page 28.)

Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and q-learning. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3185–3205. PMLR, 2020. (Cited on page 36.)

Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2010, Fort Lauderdale, Florida, USA, January 6-8, 2010*, 2010. (Cited on pages 182 and 241.)

Anna N. Rafferty, Emma Brunskill, Thomas L. Griffiths, and Patrick Shafto. Faster teaching by POMDP planning. In Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic, editors, *Artificial Intelligence in Education - 15th International Conference, AIED 2011, Auckland, New Zealand, June 28 - July 2011*, volume 6738 of *Lecture Notes in Computer Science*, pages 280–287. Springer, 2011. doi: 10.1007/978-3-642-21869-9_37. (Cited on pages 108 and 109.)

Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. *CoRR*, abs/2003.12909, 2020. (Cited on page 67.)

Giorgia Ramponi, Amarildo Likmeta, Alberto Maria Metelli, Andrea Tirinzoni, and Marcello Restelli. Truly batch model-free inverse reinforcement learning about multiple intentions. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2359–2369, Online, 26–28 Aug 2020. PMLR. (Cited on pages 163 and 167.)

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: imitation learning via regularized behavioral cloning. *CoRR*, abs/1905.11108, 2019. (Cited on page 162.)

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961. (Cited on pages 44, 126, 132, 135, and 161.)

Martin A. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings*, volume 3720 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2005. doi: 10.1007/11564096_32. (Cited on page 38.)

Thomas J Rothenberg. Identification in parametric models. *Econometrica*, 39(3):577–591, 1971. (Cited on page 157.)

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999. (Cited on page 99.)

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994. (Cited on pages 32 and 35.)

Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach, Third International Edition*. Pearson Education, 2010. ISBN 978-0-13-207148-2. (Cited on page 2.)

Jay K. Satia and Roy E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Oper. Res.*, 21(3):728–740, 1973. doi: 10.1287/opre.21.3.728. (Cited on pages 49, 60, and 61.)

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 386–394, 2013. (Cited on page 28.)

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1314–1322. JMLR.org, 2014. (Cited on pages 7, 32, and 38.)

Bruno Scherrer and Boris Lesner. On the use of non-stationary policies for stationary infinite-horizon markov decision processes. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1835–1843, 2012. (Cited on page 195.)

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015. (Cited on pages 6, 32, 43, 90, 95, and 121.)

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. (Cited on pages 32, 43, and 90.)

Wolfram Schultz. Neuronal reward and decision signals: from theories to data. *Physiological reviews*, 95(3):853–951, 2015. (Cited on page 3.)

Teddy Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: γ-maximin versus e-admissibility. *Synthese*, 140(1/2):69–88, 2004. (Cited on page 61.)

Eugene Seneta. Perturbation of the stationary distribution measured by ergodicity coefficients. *Advances in Applied Probability*, 20(1):228–230, 1988. (Cited on pages 131 and 132.)

Richard Serfozo. *Basics of applied stochastic processes*. Springer Science & Business Media, 2009. (Cited on page 18.)

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953. (Cited on pages 5, 50, and 52.)

Rui Silva, Francisco S. Melo, and Manuela Veloso. What if the world were different? gradient-based exploration for new optimal policies. In Daniel D. Lee, Alexander Steen, and Toby Walsh, editors, *GCAI-2018, 4th Global Conference on Artificial Intelligence, Luxembourg, September 18-21, 2018*, volume 55 of *EPiC Series in Computing*, pages 229–242. EasyChair, 2018. (Cited on pages 65 and 208.)

Rui Silva, Gabriele Farina, Francisco S. Melo, and Manuela Veloso. A theoretical and algorithmic analysis of configurable mdps. In J. Benton, Nir Lipovetzky, Eva Onaindia, David E. Smith, and Siddharth Srivastava, editors, *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2018, Berkeley, CA, USA, July 11-15, 2019*, pages 455–463. AAAI Press, 2019. (Cited on pages 72 and 208.)

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 387–395. JMLR.org, 2014. (Cited on pages 32 and 41.)

Satinder P Singh. Reinforcement learning with a hierarchy of abstract models. In *Proceedings of the National Conference on Artificial Intelligence*, number 10, page 202. JOHN WILEY & SONS LTD, 1992a. (Cited on page 198.)

Satinder P Singh. Scaling reinforcement learning algorithms by learning variable temporal resolution models. In *Machine Learning Proceedings 1992*, pages 406–415. Elsevier, 1992b. (Cited on page 198.)

Satinder P. Singh and Richard S. Sutton. Reinforcement learning with replacing eligibility traces. *Mach. Learn.*, 22(1-3):123–158, 1996. doi: 10.1023/A:1018012322525. (Cited on page 34.)

Satinder P. Singh and Richard C. Yee. An upper bound on the loss from approximate optimal-value functions. *Mach. Learn.*, 16(3):227–233, 1994. doi: 10.1007/BF00993308. (Cited on page 37.)

Satinder P. Singh, Tommi S. Jaakkola, and Michael I. Jordan. Reinforcement learning with soft state aggregation. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 361–368. MIT Press, 1994. (Cited on page 36.)

Satinder P. Singh, Tommi S. Jaakkola, Michael L. Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach. Learn.*, 38(3):287–308, 2000. doi: 10.1023/A:1007678930559. (Cited on pages 35 and 36.)

Saumya Sinha and Archis Ghate. Policy iteration for robust nonstationary markov decision processes. *Optim. Lett.*, 10(8):1613–1628, 2016. doi: 10.1007/s11590-016-1040-6. (Cited on page 64.)

Burrhus F Skinner. The behavior of organisms: an experimental analysis. 1938. (Cited on page 3.)

Anders Skrondal and Sophia Rabe-Hesketh. Latent variable modelling: A survey. *Scandinavian Journal of Statistics*, 34(4):712–745, 2007. (Cited on page 2.)

Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision processes. *J. Comput. Syst. Sci.*, 74(8):1309–1331, 2008. doi: 10.1016/j.jcss.2007.08.009. (Cited on page 35.)

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 881–888. ACM, 2006. doi: 10.1145/1143844.1143955. (Cited on page 35.)

Richard S Sutton. Temporal credit assignment in reinforcement learning. 1985. (Cited on pages 33 and 34.)

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3:9–44, 1988. doi: 10.1007/BF00115009. (Cited on page 33.)

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. (Cited on pages 2, 3, 7, 11, 12, 21, 22, 28, 31, 33, 34, 35, 36, 37, and 55.)

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press, 1999a. (Cited on pages 18, 21, 41, 106, 122, 123, 147, 150, 180, and 196.)

Richard S. Sutton, Doina Precup, and Satinder P. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, 1999b. doi: 10.1016/S0004-3702(99)00052-1. (Cited on page 198.)

Richard S. Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1609–1616. Curran Associates, Inc., 2008. (Cited on page 37.)

Richard S. Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 993–1000. ACM, 2009. doi: 10.1145/1553374.1553501. (Cited on page 37.)

Csaba Szepesvári. The asymptotic convergence-rate of q-learning. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*, pages 1064–1070. The MIT Press, 1997. (Cited on page 36.)

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010. doi: 10.2200/S00268ED1V01Y201005AIM009. (Cited on pages 11 and 37.)

Corentin Tallec, Léonard Blier, and Yann Ollivier. Making deep q-learning methods robust to time discretization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6096–6104. PMLR, 2019. (Cited on pages 171 and 198.)

Davide Tateo, Matteo Pirotta, Marcello Restelli, and Andrea Bonarini. Gradient-based minimization for multi-expert inverse reinforcement learning. In *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017, Honolulu, HI, USA, November 27 - Dec. 1, 2017*, pages 1–8. IEEE, 2017. doi: 10.1109/SSCI.2017.8280919. (Cited on page 163.)

Felipe W. Trevizan, Fábio Gagliardi Cozman, and Leliane Nunes de Barros. Planning under risk and knightian uncertainty. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2023–2028, 2007. (Cited on pages 49 and 62.)

Felipe W Trevizan, Fábio G Cozman, and Leliane N De Barros. Mixed probabilistic and nondeterministic factored planning through markov decision processes with set-valued transitions. In *Workshop on A Reality Check for Planning and Scheduling Under Uncertainty at ICAPS*, 2008. (Cited on page 62.)

William T. B. Uther and Manuela M. Veloso. Tree based discretization for continuous state space reinforcement learning. In Jack Mostow and Chuck Rich, editors, *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26-30, 1998, Madison, Wisconsin, USA*, pages 769–774. AAAI Press / The MIT Press, 1998. (Cited on page 36.)

Lev V. Utkin and Thomas Augustin. Powerful algorithms for decision making under partial prior information and general ambiguity attitudes. In Fábio Gagliardi Cozman, Robert Nau, and Teddy Seidenfeld, editors, *ISIPTA '05, Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications, Carnegie Mellon University, Pittsburgh, PA, USA, July 20-23 2005*, pages 349–358. SIPTA, 2005. (Cited on page 61.)

Tim van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. Inf. Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500. (Cited on pages 132, 133, and 135.)

Harm van Seijen, Hado van Hasselt, Shimon Whiteson, and Marco A. Wiering. A theoretical and empirical analysis of expected sarsa. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2009, Nashville, TN, USA, March 31 - April 1, 2009*, pages 177–184. IEEE, 2009. doi: 10.1109/ADPRL.2009.4927542. (Cited on page 35.)

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In Susan G. Mair and Robert Cook, editors, *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6-11, 1995*, pages 419–428. ACM, 1995. (Cited on page 44.)

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and Gitta Kutyniok, editors, *Compressed Sensing*, pages 210–268. Cambridge University Press, 2012. doi: 10.1017/cbo9780511794308.006. (Cited on page 251.)

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Deep conservative policy iteration. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6070–6077. AAAI Press, 2020. (Cited on page 90.)

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. (Cited on page 182.)

Nikos Vlassis, Michael L. Littman, and David Barber. On the computational complexity of stochastic controller optimization in pomdps. *ACM Trans. Comput. Theory*, 4(4):12:1–12:8, 2012. doi: 10.1145/2382559.2382563. (Cited on page 72.)

John Von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928. (Cited on page 81.)

Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934. (Cited on page 52.)

Paul Wagner. A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2573–2581, 2011. (Cited on pages 39 and 103.)

Tao Wang, Michael Bowling, and Dale Schuurmans. Dual representations for dynamic programming and reinforcement learning. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 44–51. IEEE, 2007. (Cited on pages 27 and 29.)

Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *CoRR*, abs/1907.02057, 2019a. (Cited on page 31.)

Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 21. AUAI Press, 2019b. (Cited on page 43.)

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 624–634, 2019c. (Cited on page 43.)

Christopher JCH Watkins. *Learning from delayed rewards*. PhD thesis, 1989. (Cited on page 36.)

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992. (Cited on pages 7, 32, and 36.)

Bernard Widrow and Fred W Smith. Pattern-recognizing control systems, 1964. (Cited on page 139.)

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Math. Oper. Res.*, 38(1):153–183, 2013. doi: 10.1287/moor.1120.0566. (Cited on page 63.)

Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938. (Cited on page 150.)

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. (Cited on pages 32, 40, 42, 122, and 123.)

Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. Torcs, the open racing car simulator. 4:6, 2000. (Cited on page 142.)

Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015. (Cited on page 2.)

Xin Xu, Hangen He, and Dewen Hu. Efficient reinforcement learning using recursive least-squares methods. *J. Artif. Intell. Res.*, 16:259–292, 2002. doi: 10.1613/jair.946. (Cited on page 37.)

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Math. Oper. Res.*, 36(4):593–603, 2011. doi: 10.1287/moor.1110.0516. (Cited on page 28.)

Haoqi Zhang and David C. Parkes. Value-based policy teaching with active indirect elicitation. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 208–214. AAAI Press, 2008. (Cited on page 64.)

Haoqi Zhang, Yiling Chen, and David C. Parkes. A general approach to environment design with one agent. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 2002–2014, 2009a. (Cited on pages 61, 64, and 65.)

Haoqi Zhang, David C. Parkes, and Yiling Chen. Policy teaching through reward function learning. In John Chuang, Lance Fortnow, and Pearl Pu, editors, *Proceedings 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July 6–10, 2009*, pages 295–304. ACM, 2009b. doi: 10.1145/1566374.1566417. (Cited on page 65.)

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 262–270, 2011. (Cited on page 43.)

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1433–1438. AAAI Press, 2008. (Cited on page 162.)

Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1255–1262. Omnipress, 2010. (Cited on page 162.)

List of Acronyms

AI Artificial Intelligence. 1, 3

API Approximate Policy Iteration. 37–39

AVI Approximate Value Iteration. 37, 38, 184

BF Bayes Factor. 153

BRM Bellman Residual Minimization. 38

Conf-MDP Configurable Markov Decision Process. i–iv, xv, 4–9, 14, 50, 53–61, 64–70, 72–83, 87, 88, 90, 91, 93–96, 98, 99, 114–117, 119, 120, 123, 124, 129, 134, 142, 146, 147, 157, 160–162, 165, 167, 170, 173, 205–208

CPI Conservative Policy Iteration. 39, 88, 102

DP Dynamic Programming. 12, 29

ERM Empirical Risk Minimization. 221–223

FIM Fisher Information Matrix. 40, 154–156, 231, 243–245, 248, 249

FQI Fitted Q-Iterations. 37, 171, 184, 201, 202

GLIE Greedy in the Limit with Infinite Exploration. 34–36

GLR Generalized Likelihood Ratio. 147, 148, 150–153

IL Imitation Learning. 6, 7, 146, 147, 160, 161, 163, 164

IRL Inverse Reinforcement Learning. 160

IS Importance Sampling. 42–44

KL Kullback Leibler. 43, 123, 124, 131, 163, 218, 220–222, 224, 226–228

LP Linear Programming. 27, 29

List of Acronyms

LSPI Least Squares Policy Iteration. 36

LSTD Least Squares Temporal Difference. 36, 38

MC Markov Chain, Monte Carlo. 17, 33, 34

MCE Maximum Causal Entropy. 160

MDP Markov Decision Process. i, iii, 5, 6, 8, 11–29, 32–37, 40, 41, 49–51, 53, 54, 56, 61–65, 67–74, 76–78, 106, 107, 110, 129, 149, 162, 170–174, 176, 178–181, 184, 187, 193, 194, 196, 201–203, 205, 207, 208, 235–238

MDPIP Markov Decision Process with Imprecise Probabilities. 61–63

MDPST Markov Decision Process with Set-valued Transition. 62

ML Machine Learning, Maximum Likelihood. 1, 2, 163, 164

MRP Markov Reward Process. 16, 17

NSMDP Non-Stationary Markov Decision Process. 63

PFQI Persistent Fitted Q-Iterations. 6, 7, 171, 184–186, 192, 193, 195–197, 199, 201–203, 209

PG Policy Gradient. 40–42

PI Policy Iteration. 27–29, 37, 38

POIS Policy Optimization via Importance Sampling. 42, 44, 88

POMDP Partially Observable Markov Decision Process. 63

PPO Proximal Policy Optimization. 88

PS Policy Search. 39, 43

REMPS Relative Entropy Model Policy Search. 5–7, 119, 120, 123, 128, 129, 132–142, 207, 208, 218, 221

REMS Relative Entropy Model Search. 135, 136

REPS Relative Entropy Policy Search. 42, 43, 88, 124–126, 134–136, 141

RL Reinforcement Learning. i–iv, 2–4, 6, 8, 9, 11–14, 20–23, 26, 31, 32, 34, 36, 39, 40, 43, 51, 65, 66, 88, 96, 97, 133, 137, 139, 145, 146, 160, 169–171, 195, 196, 208, 209

SMI Safe Model Iteration. 98, 99, 109, 112

SPI Safe Policy Iteration. 39, 88, 98, 99, 102, 109, 112, 115

SPMI Safe Policy Model Iteration. 5, 7, 87, 88, 97, 98, 100–102, 105–107, 109, 110, 112–116, 119, 207, 208

TD Temporal Difference. 32–36

TMDP Threatened Markov Decision Process. 65

TRPO Trust Region Policy Optimization. 88

VI Value Iteration. 27–29, 37, 186, 187, 192