University of New Hampshire
University of New Hampshire Scholars' Repository

Doctoral Dissertations | Student Scholarship

Fall 2021

LEARNING TO ACT WITH ROBUSTNESS

Reazul Hasan Russel
University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/dissertation

Recommended Citation

Russel, Reazul Hasan, "LEARNING TO ACT WITH ROBUSTNESS" (2021). *Doctoral Dissertations*. 2634.
https://scholars.unh.edu/dissertation/2634

This Dissertation is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

LEARNING TO ACT WITH ROBUSTNESS

BY

Reazul Hasan Russel

MS in Computer Science, University of New Hampshire
NH, USA 2018
BSc in Computer Science and Engineering, Southeast University
Dhaka, Bangladesh 2012

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Computer Science

September, 2021

ALL RIGHTS RESERVED

©2021

Reazul Hasan Russel

This dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science by:

Momotaz Begum
Assistant Professor of Computer Science,
University of New Hampshire

Mouhacine Benosman
Senior Principal Research Scientist,
Mitsubishi Electric Research Laboratories

Ernst Linder
Professor of Mathematics and Statistics,
University of New Hampshire

Marek Petrik, Dissertation Director
Assistant Professor of Computer Science,
University of New Hampshire

Wheeler Ruml
Professor of Computer Science,
University of New Hampshire

On July, 2021

Approval signatures are on file with the University of New Hampshire Graduate School.

To my parents Samsur and Roksana

ACKNOWLEDGEMENTS

I am forever grateful to my advisor, Marek Petrik, for guiding me and supporting me in this research work. His unfathomable knowledge and deep insight have steered my way throughout this journey. He was always available whenever I needed direction. Marek is a true legend of the field and I feel lucky to get the chance of working with him over the last five years.

I want to thank the whole UNH community for being so warm and welcoming. This has made my time at UNH enjoyable and productive. I am grateful to the UNH AI Lab and all of my lab-mates. It always was refreshing and exciting to discuss things with all the brightest minds around, no matter what the topic of the discussion was. This work would never be possible without those precious moments, thank you Bahram, Tianyi, Bence, Soheil, Will, Shayan, Monkie, Jason, Sammie, and Andreas. I also want to thank my dissertation committee members: Momotaz Begum, Mouhacine Benosman, Ernst Linder, and Wheeler Ruml.

I was delighted to have many wonderful people in my life in the past few years who are not directly affiliated to UNH. It is not easy to write down a finite list of the names and I will not dare to try that. Few names without which this section cannot be complete are: Tamjid vai and Parna apu, Zahid vai and Lily apu, Dip vai and Rawnak apu, Zareef, Zafinah, and little Zuhain.

My dad always had a fascination about science and philosophy. He predicted me to be a scientist soon after my birth. My mom, who cannot read or write by herself, never had any doubt about that projection. Being in a lower-middle class family in an underdeveloped

community, it was more realistic for them to send their son to work instead of school. They rather chose to send me to school and arranged everything I need for that. I have no idea how they managed it, I will never know. My dad would be the happiest person to glance at this thesis, I hope it is still the case up in the heaven. I am grateful to my mom for her relentless support and encouragement. I owe every bit of myself to their unconditional love and sacrifices.

Last but not the least, I want to thank my beloved wife, Tasnim, for her endless support and patience throughout the whole journey of my PhD. Her love and courage always have guided me through the challenging times. I am grateful to all my family members for their support, thank you Silvia apa, Shama, Saki, Rafat, Arafat, my in-laws, Parul aunt, Monir uncle and Shan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

LEARNING TO ACT WITH ROBUSTNESS

by

Reazul Hasan Russel

University of New Hampshire, September, 2021

Reinforcement Learning (RL) is learning to act in different situations to maximize a numerical reward signal. The most common approach of formalizing RL is to use the framework of optimal control in an inadequately known Markov Decision Process (MDP). Traditional approaches toward solving RL problems build on two common assumptions: i) exploration is allowed for the purpose of learning the MDP model and ii) optimizing for the expected objective is sufficient. These assumptions comfortably hold for many simulated domains like games (e.g. Atari, Go), but are not sufficient for many real-world problems. Consider for example the domain of precision medicine for personalized treatment. Adopting a medical treatment for the sole purpose of learning its impact is prohibitive. It is also not permissible to embrace a specific treatment procedure by considering only the expected outcome, ignoring the potential of worst-case undesirable effects. Therefore, applying RL to solve real-world problems brings some additional challenges to address.

In this thesis, we assume that exploration is impossible because of the sensitivity of actions in the domain. We therefore adopt a *Batch* RL framework, which operates with a logged set of fixed dataset without interacting with the environment. We also accept the need of finding

solutions that work well in both average and worst case situations, we label such solutions as *robust*. We consider the robust MDP (RMDP) framework for handling these challenges. RMDPs provide the foundations of quantifying the uncertainties about the model by using so called *ambiguity sets*. Ambiguity sets represent the set of plausible transition probabilities - which is usually constructed as a multi-dimensional confidence region. Ambiguity sets determine the trade-off between robustness and average-case performance of an RMDP.

This thesis presents a novel approach to optimizing the shape of ambiguity sets constructed with weighted L_1-norm. We derive new high-confidence sampling bounds for weighted L_1 ambiguity sets and describe how to compute near-optimal weights from coarse estimates of value functions. Experimental results on a diverse set of benchmarks show that optimized ambiguity sets provide significantly tighter robustness guarantees.

In addition to reshaping the ambiguity sets, it is also desirable to optimize the size and position of the sets for further improvement in performance. In this regard, this thesis presents a method for constructing ambiguity sets that can achieve less conservative solutions with the same worst-case guarantees by 1) leveraging a Bayesian prior, and 2) relaxing the requirement that the set is a confidence interval. Our theoretical analysis establishes the safety of the proposed method, and the empirical results demonstrate its practical promise.

In addition to optimizing ambiguity sets for RMDPs, this thesis also proposes a new paradigm for incorporating robustness into the constrained-MDP framework. We apply robustness to both the rewards and constrained-costs, because robustness is equally (if not more) important for the constrained costs as well. We derive required gradient update rules and propose a policy gradient class of algorithm. The performance of the proposed algorithm is evaluated on several problem domains.

Parallel to Robust-MDPs, a slightly different perspective on handling model uncertainties is to compute soft-robust solutions using a risk measure (e.g. Value-at-Risk or Conditional Value-at-Risk). In high-stakes domains, it is important to quantify and manage risk that arises from inherently stochastic transitions between different states of the model. Most

prior work on robust RL and risk-averse RL address the inherent transition uncertainty and model uncertainty independently. This thesis proposes a unified Risk-Averse Soft-Robust (RASR) framework that quantifies both model and transition uncertainties together. We show that the RASR objective can be solved efficiently when formulated using the Entropic risk measure. We also report theoretical analysis and empirical evidences on several problem domains.

The methods presented in this thesis can potentially be applied in many practical applications of artificial intelligence, such as agriculture, healthcare, robotics and so on. They help us to broaden our understanding toward computing robust solutions to safety critical domains. Having robust and more realistic solutions to sensitive practical problems can inspire widespread adoption of AI to solve challenging real world problems, potentially leading toward the pinnacle of the age of automation.

CHAPTER 1
INTRODUCTION

Artificial Intelligence (AI) is defined as the study of rational actions based on situations. Learning a sequence of actions to achieve a goal is known as planning- which is an important sub-field of AI. Typical planning involves an agent that can perceive a situation through its sensors and can act depending on that [1]. AI devises a well trained agent capable of taking sensible actions based on the perceptions. Planning usually involves a finite set of distinct situations, a finite set of actions, a dynamics of the model specifying the outcome of each action and an objective function to optimize. Typical planning takes the predictive model dynamics as granted without worrying about where they come from [2]. This thesis deviates from such conventional approach and seeks to develop a goal-seeking agent that operates in an uncertain environment. The target is to optimize the interplay between planning and real-time action selection while also learning about a model dynamics of the environment. Methods optimizing such objectives are traditionally known as *Reinforcement Learning (RL)*. Learning from trial-and-error and dealing with delayed reward feedback are two main distinguished features of RL. In this thesis, we will focus on planning problems involving sequential decision making under model uncertainty.

Uncertainty is an inherent part of real-world optimization problems, meaning the problem is not known exactly when it is being solved. Measurement/estimation errors in the data collection process, implementation errors due to the impossibility of implementing a solution exactly, limited data-sets, modeling errors are some common reasons for data uncertainty. It is common in real-world problems that a small uncertainty in data can make the nominal

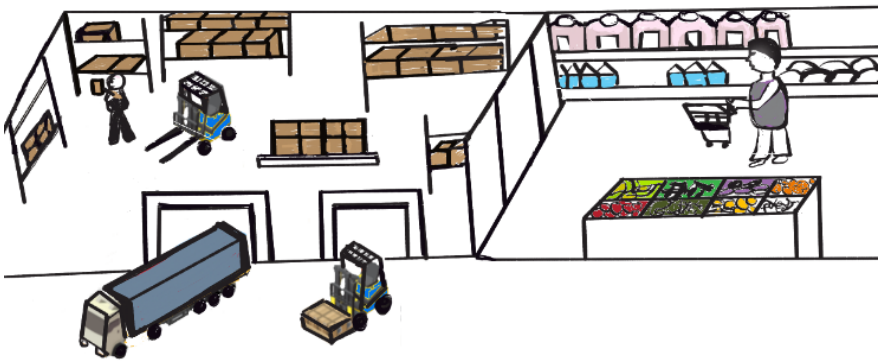

Figure 1.1: Inventory control problem to meet customer demands.

solution practically meaningless. This issue can get magnified for problems involving sequential decision making, yielding a different type of "curse" that requires attention and cure. This is known as *curse of uncertainty*. This refers to the fact that the solution to a sequential decision making problem can be very sensitive with respect to the model uncertainty, when the model dynamics is estimated from samples. As a result, any RL method needs to handle this uncertainty in a tractable way.

We now describe two specific examples: i) inventory management and ii) invasive species management. Both examples involve sequential decision making under model uncertainty. They will help us better understand the context of the problems and their associated challenges.

Inventory Management Inventory control problems are one among the earliest areas of application for sequential decision making models. The scope of the model is to determine the optimal reorder level for a single product in a single store at each decision epoch (e.g. week or month). The amount of a product available in stock at the time of review represents the system state. The action corresponding to a state represents the amount of product to order from the warehouse. Transition to a next state is determined by the order amount and the random customer demand for the product throughout the decision epoch. The demand distribution is usually unknown, associates significantly high uncertainty and is estimated from historical data. A decision rule specifies the restocking amount as a function of the

Figure 1.2: Managing invasive species.

state representing the current inventory level at hand. The goal is to find a reordering policy that is able to meet the customer demands while minimizing the long-run average ordering and inventory carrying costs. A detail description of the model can be found in Chapters 1.2 and 3.2 of [3] and also in [4].

Invasive Species Management Ecological models are often complex, stochastic in nature, data collection is expensive and also involves a lot of uncertainty. Developing an optimal management strategy is therefore very difficult. Yet it is important that the decisions are robust due to their long term impacts. In this invasive species management problem, the population dynamics of the species is modeled in an ecosystem where the abundance level of the species represents a state. The state space evolves according to an exponential population dynamics. The land manager has a choice to apply or not to apply a treatment action based on the current population. Applying a control measure incurs an immediate cost, but can bring future retribution with species being under control. The affects of a treatment action depends on the current population level and is highly variable due to many environmental factors. Data available to model the population dynamics are usually not sufficient to infer a precise model. The goal is then to develop a good strategy that remains effective even when the reality significantly deviates from what the data-set asserts. A detailed description

of the model can be found in Section 4.2 of [5] and also in Chapter 5 of [6]. This species management problem is a an instance of sequential decision making because the decision about treatment measure has a temporal aspect. The land manager needs to periodically decide about the action to take in different instances of times.

These two problems, however, are some mere examples. Almost all control system engineering problems involve sequential decision making and therefore can be formulated as an RL problem. However, a precise model for planning may not be readily available for such problems. This thesis proposes methods to compute robust and risk-sensitive solutions for such problems when the model dynamics are not known precisely.

1.1 Framework

Reinforcement Learning (RL) is a branch of machine learning that aims to develop intelligent agents capable of learning to act in an unknown environment. The goal is to optimize some long-term objectives represented by a scalar value known as reward signal. We assume in this thesis that the environment dynamics is stochastic and the states of the environment are fully observable. Markov Decision Processes (MDPs) provide a versatile framework for modeling RL problems with these characteristics. MDPs incorporate three essential aspects required for learning, namely: sensation of the situations, notion of actions applicable in a situation and the concept of a goal or objective. RL involves simultaneous learning of *good* actions in different situations (exploitation) along with learning about the dynamics of the *unknown* MDP model (exploration). This poses one unique challenge for RL which is commonly known as

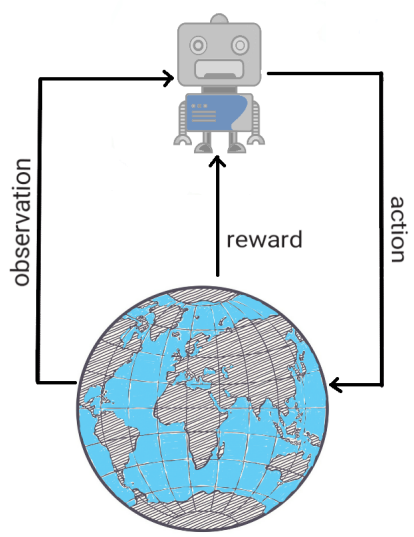

Figure 1.3: Sequential Decision Making.

exploration-exploitation trade-off.

MDP is a simple model capable of representing an RL problem with a finite set of states, a finite set of actions, transitions between states and a reward signal. The objective is to maximize the discounted infinite-horizon sum of rewards, where rewards from distant future have discounted values but are not irrelevant. MDPs provide enough flexibility to model a wide variety of different problems. In general, MDPs are learned from historical data. Given a precise MDP model for a problem, a reasonable solution can be computed in a tractable way. It is also easy to incorporate different assumptions and constraints into an MDP framework to better model a specific problem. Such flexibility gives rise to many variations of MDPs, including but not limited to: Constrained MDPs (CMDPs), Robust MDPs (RMDPs), Partially Observable MDPs (POMDPs), Continuous-time MDPs and so on. One particular flavor of MDPs we will be using throughout this thesis is RMDP, where the transition probability is uncertain and the objective is to maximize the worst-case value within a set of plausible models. These models are discussed in more detail in the next chapter.

1.2 Challenges in Real-World Decision Making

Recently RL has been used to solve several challenging simulated domains and games like Go, Atari, StarCraft etc. [7, 8]. Training RL methods to solve such simulated games have several advantages like: data is unlimited and can be obtained at will from simulation, the system dynamics are often deterministic and stationary, poor choice of actions does not have costly consequences and exploration is welcome to the highest extent.

Having impressive successes in simulated games and synthetic domains, RL has a high potential to make an impact on real-world problems involving sequential decisions. Such problems are common in agriculture, resource management, inventory management, personalized recommendation systems, healthcare, autonomous driving, and robotics. Yet our understanding of applying RL to solve these problems is limited. Some of the main challenges

of applying RL to solve a real-world problem are:

- It is often very costly to make a mistake. For example, trying a medication just to learn its impact is risky and can cause severe harm to a patient. So, exploration to collect more data in the real world is not always practical.

- The amount of data available to train an RL agent is often limited. Gathering more data can be restrictive as well. For example, autonomous vehicles cannot simply keep driving on roads just to collect more data without worrying about the safety of others.

- It is a very common requirement for sensitive problems to learn solutions that can provide a guarantee about its worst-case performance. For example, in an electric power system decision and control problem, any learned control law needs to guarantee that power system outage is not going to happen.

While such challenges are not there in simulated domains like games, they are indeed a part of most real-world problems. One common fact is that, it is not feasible to develop a good simulator for most real world problems as well. Because they are complex, inherently stochastic, evolve in a non-stationary way, have strong safety constraints, and simulating them can be difficult and costly in terms of both time and money. So, building a perfect simulator and then keep training with unlimited data is not a way to go. Moreover, only optimizing the expected return can be insufficient for most real world problems. So, developing solutions for real world problems poses a different and harder set of challenges.

1.3 Contributions

The goal of this thesis is to develop robust and risk-averse algorithms for problems requiring sequential decision making. We use robust MDPs (RMDPs) to compute policies with provable worst-case guarantees in reinforcement learning. The quality and robustness of an RMDP solution are determined by the ambiguity set−the set of plausible transition

probabilities−which is usually constructed as a multi-dimensional confidence region. We depart from the traditional methods of constructing ambiguity sets as confidence regions using concentration inequalities, which usually leads to overly conservative solutions. The main contributions in this direction are:

- Computing weights from the value function estimates to customize the shape of the ambiguity sets for a specific problem. Show that the structure of a near-optimal ambiguity set is problem specific and need not be uniform and symmetric in shape.
- Incorporating prior knowledge using Bayesian inference and optimize the size and position of the ambiguity sets. Show that the novel ambiguity sets are tractable, significantly less conservative than existing ones and are guaranteed to provide a robust estimate.

Constrained-MDPs (CMDPs) are a super class of MDPs that incorporate multiple reward functions. One reward function is used to set the optimization objective and the others are used to set some constraints restricting the space of admissible policies. Many practical problems come with such constraints and the CMDP framework provides a useful model to deal with them. While robustness is important in general MDPs, it is also important to incorporate robustness on the constraint costs. In this regard, this thesis contributes in:

- Incorporating robustness to both objective and constraints of CMDPs, leading to a new paradigm of Robust-CMDPs (RCMDPs). We derive the associated optimization objective and propose a policy optimization technique.

A class of methods that build on robust optimization but employ different risk measures and reduce conservativeness are known as epistemic risk aversion [9] or soft-robustness [10, 11]. These methods also estimate the range of possible models consistent with the observed data and then optimize a policy with respect to a risk metric across different models. This thesis uses entropic risk measure, which is an exponential utility based convex risk measure and contributes in:

- Developing a unified Risk-Averse Soft-Robust (RASR) framework that quantifies and manages both model and transition uncertainties. We propose a tabular method for RASR framework and also present its scaled up version for larger problems.

For all the proposed ideas, we report relevant theoretical analysis along with empirical evaluation on several problem domains.

1.4 Outline

The thesis is organized as follows: Chapter 2 presents the foundations of RL and describes many of the relevant concepts required to formulate the research ideas presented in later chapters of the thesis. Chapter 3 presents the detailed derivations and theories of weighted norm-bounded ambiguity sets. Construction of near-optimal ambiguity sets under Bayesian framework is presented in Chapter 4. Chapter 5 describes the unified Robust-CMDP framework and proposes the constrained robust policy optimization techniques. Chapter 6 presents the Risk-Averse Soft-Robust (RASR) framework along with relevant theoretical and empirical analysis.

CHAPTER 2

REINFORCEMENT LEARNING AND ROBUSTNESS

Reinforcement Learning is learning to map situations to actions that maximize a long term objective [2, 12]. The actions are not labeled for training, rather the agent needs to learn about most rewarding actions by trying them. An action affects both the immediate reward and the next state yielding short and long term consequences.

An important (but not compulsory) component of the reinforcement learning framework is a model of the environment. This thesis focuses on dynamical decision making in stochastic environment represented by a model with finite set of states and a finite set of actions. The dynamics of the stochastic system is represented by a transition probability distribution. This distribution is supposed to be known in an ideal world, but unfortunately that is not the case in reality. We assume throughout this thesis that this transition distribution is uncertain and resides within an ambiguity set. The nature plays against the decision maker at each decision stage by picking an adversarial transition within that uncertainty set. The goal in robust RL is to maximize the worst-case expected value over the set of plausible adversarial actions. In this section, we formalize the framework of robust RL that we use in this thesis.

Before going into the details in later sections, we now specify some important definitions and notations: we use vectors $x \in \mathbb{R}^n$ throughout this thesis to represent various quantities. All vectors are column vectors in finite dimensional spaces unless otherwise specified. Vectors $\mathbf{1}$ and $\mathbf{0}$ denote all ones and zeros respectively of an appropriate size suitable for the context. An identity matrix of appropriate size is represented by $\boldsymbol{I}$.

Definition 2.0.1. (Vector Norm) Let us assume that $x \in \mathbb{R}^n$ is a vector. A norm $\|x\| : x \to \mathbb{R}$ is a function from vector x to a real number representing some sense of length or magnitude of the vector x. Following are the definitions of some specific norms:

- L_p norm: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$
- L_1 norm: $\|x\|_1 = \sum_{i=1}^n |x_i|$
- L_∞ norm: $\|x\|_\infty = \max_{i=1}^n |x_i|$
- Weighted L_1 norm: $\|x\|_{1,w} = \sum_{i=1}^n w_i|x_i|$
- Span seminorm: $\|x\|_s = \max_{i=1}^n x_i - \min_{i=1}^n x_i$, where span seminorm satisfies all the properties of a norm except that $\|x\|_s = 0$ does not imply that $x = \mathbf{0}$.
- Dual norm: $\|z\|_\star = \sup\{z^\intercal x \ : \ \|x\| \leq 1\}$, it is well known that dual norms to L_1, L_2, and L_∞ are norms L_∞, L_2, and L_1 respectively.

2.1 Markov Decision Processes (MDPs)

Markov Decision Process (MDP) is the standard mathematical framework to model the environment for reinforcement learning [2, 3, 13]. An MDP is a tuple, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, p_0)$, where $\mathcal{S} = \{1, \ldots, S\}$ is a finite set of states, $\mathcal{A} = \{1, \ldots, A\}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability defining the next state s' given the current state s and action a, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function. The rewards are known but the true transition probabilities $P^\star : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are unknown. An initial state distribution $p_0 : \mathcal{S} \to [0, 1]$ is such that $\sum_{s \in \mathcal{S}} p_0(s) = 1$. At each time step $t = 1, \ldots, T$, the decision maker observes a state $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$, receives a reward $r_t \in \mathbb{R}$ and transitions to a new state $s_{t+1} \sim P(s_t, a_t)$. This thesis focuses on maximizing the utility for the infinite horizon discounted MDP, where utility is the $\gamma-$discounted cumulative sum of

rewards. With $\gamma < 1$ and $|r| \leq R_{max}$, the utility for a sequence of states is defined as:

$$\begin{aligned} v([s_0, s_1, \ldots]) &= r(s_1) + \gamma r(s_2) + \gamma^2 r(s_3) + \ldots \\ &= \sum_{t=0}^{\infty} \gamma^t r(s_{t+1}) \\ &\leq \sum_{t=0}^{\infty} \gamma^t R_{\max} \\ &= R_{\max}/(1-\gamma) \end{aligned}$$

The goal is to learn a policy mapping each state to an action that maximizes the utility.

Definition 2.1.1. (Policy)([3]) A policy represented by π is defined as a mapping from a state $s \in \mathcal{S}$ to possible actions $a \in \mathcal{A}$. A deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ maps one action to each state of the MDP and a randomized policy $\pi : \mathcal{S} \to \Delta^{\mathcal{A}}$ assigns a distribution over the available actions in each state of the MDP. The set of all deterministic stationary policies is denoted by Π.

For infinite-horizon discounted MDPs used in this thesis, there always exists an optimal *deterministic* and *stationary* policy [3]. Our focus in this thesis therefore remains on stationary policies where the optimal actions in the same state stay constant over time. We consider both deterministic and stochastic policies in different parts of this thesis depending on the problem settings.

Throughout the thesis, the matrix P_π will denote the transition probability matrix where rows represent the *from* states and columns represent the *to* states. For ay states $s, s' \in \mathcal{S}$:

$$P_\pi(s, s') = \sum_{a \in \mathcal{A}} \pi(s, a) P(s, a, s')$$

The rewards r_π for a state s and policy π is defined as:

$$r_\pi(s) = \sum_{a \in \mathcal{A}} \pi(s, a) \cdot P_\pi(s, a)^T r(s, a)$$

The optimization objective with a policy π is the expected utility of executing π starting at state $s_0 \in \mathcal{S}$ and is expressed as:

$$v_\pi(s_0) = \mathbb{E}_P \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \right]$$

Where the expectation is with respect to the transition probability distributions determined by s and π. This expected utility is known as the value function.

Definition 2.1.2. (Value Function)([3]) A value function $v : s \to \mathbb{R}$ is an estimate of the expected utility of being in a state s when following a policy π.

The optimal policies are often represented with a value function, as shown later in definition 2.1.4. Now, among all the available stationary stochastic policies Π, there exist one (or more) policy which has a higher value function compared to all others. This is called an optimal policy and is denoted as $\pi_s^\star$:

$$\pi_s^\star \in \arg\max_{\pi \in \Pi} v_\pi(s)$$

Note that the optimal policy $\pi^\star$ is independent of the initial state in the infinite horizon MDPs. Because this thesis analyzes the impact of using different transition probabilities as the true transition model is unknown, we use a subscript to indicate which ones are used. The optimal value function for some transition probabilities P is, therefore, denoted as $v_P^\star : \mathcal{S} \to \mathbb{R}$, and the value function for a *deterministic policy* $\pi : \mathcal{S} \to \mathcal{A}$ is denoted as v_P^π. The total return $\rho(\pi, P)$ of a policy π under transition probabilities P is:

$$\rho(\pi, P) = p_0^\intercal v_P^\pi,$$

where p_0 is the initial state distribution.

As the value of being in a state is determined as the expected sum of discounted rewards from that state onward, it is therefore obvious that there is a relationship between the value

function of a state and it's neighbors. The value function of a state s can be decomposed as the immediate reward of state s plus the expected discounted values of next states s' following an optimal action $a \in \mathcal{A}$:

$$v(s) = \max_{a \in \mathcal{A}} \big(r(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \big) \tag{2.1}$$

This famous recursive formulation is known as *Bellman optimality equation* [14]. The value function of the states are solutions of the set of Bellman equations. There are n Bellman equations when total number of states is n, one associated to each state. The n equations contain n unknowns representing the utilities of the states. But the equations are not linear because of the max operator and therefore cannot be solved using linear algebra techniques. This set of equations can instead be solved in an iterative way, which is the basis for many techniques in the RL literature including a fundamental algorithm known as *value iteration*.

The value function for state s and a policy π is $v^\pi(s)$ and satisfies:

$$v^\pi(s) = r(s,a) + \gamma \sum_{s'} P(s'|s,\pi(s)) v^\pi(s') \tag{2.2}$$

This is known as the Bellman equation for policy evaluation. While the Bellman optimality equation defined in (2.1) involves a max operator and is non-linear, the policy evaluation in equation (2.2) is linear. Therefore, the policy evaluation question involves a system of linear equations and can be solved quickly using linear programming techniques.

The Bellman optimality equation (2.1) is a contraction and is guaranteed to converge to a fixed point. Finding a fixed point of the nonlinear Bellman operator is equivalent to finding the optimal value function, which then leads to an optimal policy.

Definition 2.1.3. (Bellman Operator)([3]) The Bellman operator $T : \mathbb{R}^S \to \mathbb{R}^S$ and the

policy evaluation update $T_\pi : \mathbb{R}^S \to \mathbb{R}^S$ for a policy π are defined as:

$$T_\pi v = r_\pi + \gamma P_\pi v$$

$$Tv = \max_{\pi \in \Pi} T_\pi v$$

Here T is a non-linear operator representing (2.1) and T_π is an affine operator representing (2.2). The operator T is usually defined with a state-wise decomposition. The optimal value function $v^\star \in \mathbb{R}^{\mathcal{S}}$ is achieved if and only if the Bellman operator reaches a stationary point: $v^\star = Tv^\star$ [14]. The simplest way to compute an optimal policy from the optimal value function is to take the greedy policy with respect to the value function.

Definition 2.1.4. (Greedy Policy)([3]) A greedy policy takes in each state the action that maximizes the expected value of transitioning to the following state.

$$\pi(s) = \arg \max_{a \in \mathcal{A}(s)} + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') v(s')$$

Therefore, an optimal policy can be obtained easily from an optimal value function. But unfortunately, there is no known strongly-polynomial algorithm that can solve an MDP in a number of arithmetic operations polynomial in S and A [15, 16]. The computational complexity for solving an infinite horizon $\gamma-$discounted MDP is *P-complete* [15].

Given a policy π, the value function induced by π can be determined in $O(S^3)$ arithmetic operations by solving a system of linear equations, as of (2.2) for each $s \in \mathcal{S}$. And policy improvement (2.1) can be performed in $O(S^2 A)$ operations [3,17]. The total running time for MDP solution methods (e.g. value iteration, policy iteration) is therefore polynomial if and only if the total number of iterations required to find an optimal policy is polynomial [16]. One exception is the linear programming based solution method for MDPs [3], which requires number of arithmetic operations polynomial in S, A and B. Here B represents the maximum number of bits required to represent any transition P or reward r [16].

2.2 Batch Reinforcement Learning

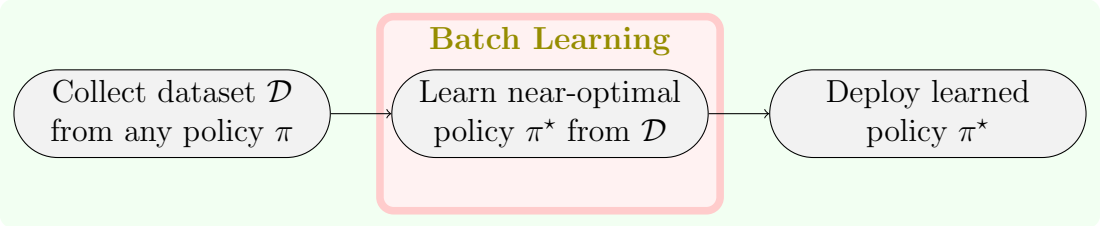

Figure 2.1: Batch reinforcement learning with three sequential phases.

Many of the algorithms presented in this thesis operate in a *batch* reinforcement learning [18] setup, where a policy needs to be computed from a logged dataset without interacting with the environment. This setting is common when experimentation is either too expensive or time-consuming, such as in medical care, agriculture, or even robotics.

Definition 2.2.1. (Batch RL) ([18]) The batch RL task is to find a policy that maximizes the expected sum of rewards within the general agent-environment loop of classical RL, but the learning experience (set of transition samples) is a priori given and fixed.

In batch RL setting, the agent is not allowed to interact with the environment during learning. Figure 2.1 shows the batch RL setup with three different sequential phases: i) data collection, ii) learning near-optimal policy from batch of data, and iii) execution of the learned policy. Policies remain fixed after the learning phase is done. Exploration and online policy improvement are not permitted. The exploration-exploitation trade-off is therefore not a concern in Batch RL. Instead of learning online by taking an action a_t in state s_t at time step t and then updating policy according to the observed next state s_{t+1} and reward r_{t+1}, as done in a general reinforcement learning setup, the learning agent only receives a fixed and finite dataset $\mathcal{D}$ of n transition samples: $\mathcal{D} \subseteq \{(s_t, a_t, r_{t+1}, s_{t+1}) \ : \ s_t, s_{t+1} \in \mathcal{S}, a_t \in \mathcal{A}, t = 1 \dots n\}$. The only assumption about $\mathcal{D}$ is that the state s_{t+1} in $(s_t, a_t, r_{t+1}, s_{t+1}) \in \mathcal{S}$ is distributed according to the *true* transition probabilities: $s_{t+1} \sim P^\star(s_t, a_t, \cdot)$. We make no assumptions on the policy used to generate the dataset.

Batch reinforcement learning introduces two important challenges [19–22]. First, the amount of data may be insufficient to compute a good policy. Second, evaluating the quality of a policy without simulation can be difficult. We tackle these challenges by inferring a model of the environment given the dataset $\mathcal{D}$ and learning a *robust* policy that can provide high-confidence lower bound on the *true* return [19, 23, 24]. The later chapters 3 and 4 of this thesis will unfold the relevant details of these treatments.

2.3 Safe Return Estimate

We operate in this thesis in a *batch RL* setup where a fixed dataset $\mathcal{D}$ based on the historic interactions with the environment is provided: $\mathcal{D} \subseteq \{(s, a, s') \ : \ s, s' \in \mathcal{S}, a \in \mathcal{A}\}$. More data cannot be collected at will in this situation, but a solution with certain performance guarantee is still important to obtain. For example, it can reduce the chance of an unpleasant surprise when the policy is deployed. Or it can also be used to justify the need to collect more data because a better performing policy cannot be learned from the present batch of data [19, 23, 24]. If the lower bound on the return is smaller than the return of the currently deployed policy, then the current policy need not be replaced.

Our *objective* is to compute a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the return $\rho(\pi, P^\star)$. Because the objective depends on the unknown $P^\star$, we instead compute a policy with the greatest lower guarantee on the return. The term *safe return estimate* refers to the lower bound estimate.

Definition 2.3.1 (Safe Return Estimate)**.** The estimate $\tilde{\rho} : \Pi \to \mathbb{R}$ of return is called *safe* for a policy π with probability $1 - \delta$ if it satisfies:

$$\mathbb{P}_{P^\star}\Big[\tilde{\rho}(\pi) \leq \rho(\pi, P^\star) \,\Big|\, \mathcal{D}\Big] \geq 1 - \delta \ .$$

Remark. Under Bayesian assumptions, $P^\star$ is a random variable and the guarantees are conditional on the dataset $\mathcal{D}$. This is different from the frequentist approach, in which the

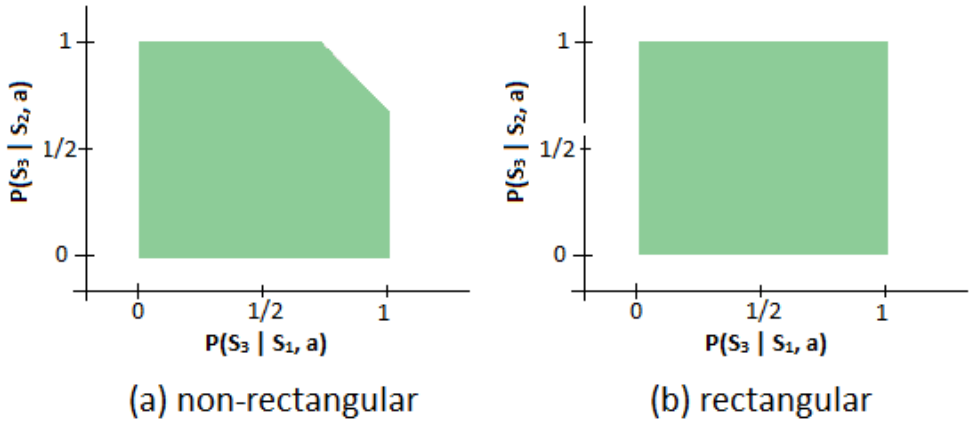

Figure 2.2: Rectangularity: dependence of transition probabilities between different states.

random variable is $\mathcal{D}$ and the guarantees are conditional on $P^\star$. See, for example, Sections 5.2.2 and 6.1.1 in [25] for a discussion of the merits of the two approaches. Unless it is apparent from the context, we indicate whether the probability is conditional on $\mathcal{D}$ or $P^\star$ whenever it appears on later chapters.

Having a safe return estimate is very important in practice. A low safe estimate informs the stakeholders that the policy may not perform well when deployed. They may, instead, choose to gather more data, keep the existing (baseline) policy, or use a more informative domain [19, 26].

2.4 Robust Markov Decision Processes (RMDPs)

Robust Markov Decision Processes (RMDPs) are a convenient model that can be used to compute and tractably optimize the *safe* return estimate ($\max_\pi \tilde{\rho}(\pi)$). Our RMDP model has the same states $\mathcal{S}$, actions $\mathcal{A}$, rewards $r_{s,a}$ as the MDP. The transition probabilities for each state s and action a, denoted as $p_{s,a} \in \Delta^S$, are assumed chosen adversarialy from an *ambiguity set* $\mathcal{P}_{s,a}$. We use $\mathcal{P}$ to refer cumulatively to $\mathcal{P} = \bigotimes_{s_t \in \mathcal{S}, a_t \in \mathcal{A}} \mathcal{P}_{s,a}$, for all states s and actions a

Definition 2.4.1. (Ambiguity Set) ([27]) An ambiguity set $\mathcal{P}_{s,a}$ for state s and action a is a set of confidence for the transition probability distribution over the next states: $\mathcal{P}_{s,a} =$

$\{[p(1), p(2), \ldots, p(S)] \in R_+^S : \sum_{i=1}^{S} p(i) = 1\}$. A convenient way of defining ambiguity sets is to use a norm-distance from a given *nominal transition probability* $\bar{p}_{s,a}$:

$$\mathcal{P}_{s,a} = \left\{ p \in \Delta^S \ : \ \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a} \right\} \tag{2.3}$$

for a given $\psi_{s,a} \geq 0$ and a nominal point $\bar{p}_{s,a}$.

We focus on ambiguity sets defined by the L_1 norm because they give rise to RMDPs that can be solved efficiently [28]. We restrict our attention to s, a-rectangular ambiguity sets. Rectangular ambiguity sets allow the nature to choose the worst transition probability independently for each state and action [29,30]. Limitations of rectangular ambiguity sets are well known [31–33] but they represent a simple, tractable, and practical model. Figure 2.2 presents the notion of rectangularity, where the horizontal X-axis represents the probability of transitioning to a state S_3 from state S_1 after taking an action a and the vertical Y-axis represents the probability of transitioning to a state S_3 from state S_2 with action a. Figure 2.2(a) on the left shows that, depending on the position in X-axis, the range of values in Y-axis is not constant and so the transitions from states s_1 and s_2 are dependent. Figure 2.2(b) shows that the probabilities in Y-axis is uniform and does not depend on the X-axis. So the transition probabilities from states S_1 and S_2 are independent, which is known as rectangular.

The quality of the optimal RMDP policy depends on the ambiguity set used to compute the solution. It must be the smallest set that is large enough to guarantee that the solution is a lower bound. RL algorithms usually construct data-driven ambiguity sets as *confidence regions* derived from concentration inequalities [19, 20, 34, 35]. Using, for example, a 95% confidence region over possible transition probabilities translates to a 95% confidence that the RMDP return lower bounds the true return. Unfortunately, concentration inequalities lead to solutions that are too conservative to be practical. Another approach is to construct ambiguity sets from likelihood levels of probability distributions, but this method requires

complex modeling and does not provide finite-sample guarantees [27, 36–38].

RMDPs have properties that are similar to regular MDPs (see, for example, [29, 30, 37, 39, 40]). The robust Bellman operator $\widehat{T}_{\mathcal{P}}$ for an ambiguity set $\mathcal{P}$ for a state s computes the best action with respect to the worst-case realization of the transition probabilities:

$$\begin{aligned}(\widehat{T}_{\mathcal{P}} v)(s) &= \max_{a \in \mathcal{A}} \min_{p \in \mathcal{P}_{s,a}} (r_{s,a} + \gamma \cdot p^{\mathsf{T}} v) \\ &= \max_{a \in \mathcal{A}} \min_{p \in \Delta^S} \left\{ (r_{s,a} + \gamma \cdot p^{\mathsf{T}} v) \middle| \, \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a} \right\}\end{aligned} \tag{2.4}$$

The symbol $\widehat{T}^{\pi}_{\mathcal{P}}$ denotes a robust Bellman update for a given *stationary* policy π:

$$(\widehat{T}^{\pi}_{\mathcal{P}} v)(s) = \min_{p \in \mathcal{P}_{s,\pi(s)}} (r_{s,\pi(s)} + \gamma \cdot p^{\mathsf{T}} v^{\pi})$$

The optimal robust value function $\hat{v}^{\star}$, and the robust value function $\hat{v}^{\pi}$ for a policy π must, similarly to MDPs, satisfy:

$$\hat{v}^{\star} = \widehat{T}_{\mathcal{P}} \hat{v}^{\star}, \qquad \hat{v}^{\pi} = \widehat{T}^{\pi}_{\mathcal{P}} \hat{v}^{\pi} \ .$$

In general, we use a hat to denote quantities in the RMDP and omit it for the MDP. When the ambiguity set $\mathcal{P}$ is not obvious from the context, we use it as a subscript $\hat{v}^{\star}_{\mathcal{P}}$. The robust return $\hat{\rho}$ is defined as [36]:

$$\hat{\rho}(\pi, \mathcal{P}) = \min_{P \in \mathcal{P}} \rho(\pi, P) = p_0^{\mathsf{T}} \hat{v}^{\pi}_{\mathcal{P}} \ ,$$

where $p_0 \in \Delta^S$ is the initial distribution. In the next two chapters, we describe methods that construct $\mathcal{P}$ from $\mathcal{D}$ in order to guarantee that $\hat{\rho}$ is a tight lower bound on ρ.

The optimal policies for RMDPs are stochastic, history dependent and NP-hard to compute for non-rectangular ambiguity sets [30, 36]. But the problem becomes tractable for *s,a*-rectangular ambiguity sets. Ho et al. [41] show that the robustness in *s,a*-rectangular setting can be handled with $O(S \log S)$ additional time for each state and action, keeping the overall complexity tractable.

2.4.1 Ambiguity Sets as Confidence Regions

We now describe the standard approach to constructing ambiguity sets as multidimensional confidence regions. This is a natural approach but, as we discuss later in chapters 3 and 4, may be unnecessarily conservative.

Before describing how the ambiguity sets are constructed, we need the following auxiliary lemma. The lemma shows that when the robust Bellman update lower-bounds the true Bellman update then the value function estimate is safe.

Lemma 2.4.1. *Consider a policy π, its robust value function $\hat{v}^\pi$, and true value function v^π such that $\hat{v}^\pi = \widehat{T}^\pi \hat{v}^\pi$ and $v^\pi = T^\pi v^\pi$. Then, $\hat{v}^\pi \leq v^\pi$ element-wise whenever $\widehat{T}^\pi \hat{v}^\pi \leq T^\pi \hat{v}^\pi$.*

Proof. Using the assumption $\widehat{T}^\pi \hat{v}^\pi \leq T^\pi \hat{v}^\pi$, and from $\hat{v}^\pi = \widehat{T}^\pi \hat{v}^\pi$ and $v^\pi = T^\pi v^\pi$, we get by algebraic manipulation:

$$\hat{v}^\pi - v^\pi = \widehat{T}^\pi \hat{v}^\pi - T_P^\pi v^\pi \leq T^\pi \hat{v}^\pi - T^\pi v^\pi = \gamma P_\pi (\hat{v}^\pi - v^\pi) \ .$$

Here, P_π is the transition probability matrix for the policy π. Subtracting $\gamma P_\pi(\hat{v}^\pi - v^\pi)$ from the above inequality gives:

$$(\mathbf{I} - \gamma P_\pi)(\hat{v}^\pi - v^\pi) \leq \mathbf{0} \ ,$$

where $\mathbf{I}$ is the identity matrix. Because the matrix $(\mathbf{I} - \gamma P_{\pi^\star})^{-1}$ is monotone, as can be seen from its Neumann series, we get:

$$\hat{v}^\pi - v^\pi \leq (\mathbf{I} - \gamma P_\pi)^{-1} \mathbf{0} = \mathbf{0} \ ,$$

which proves the result. □

Note that the inequality holds with respect to the robust value function $\hat{v}^\pi$. The require-

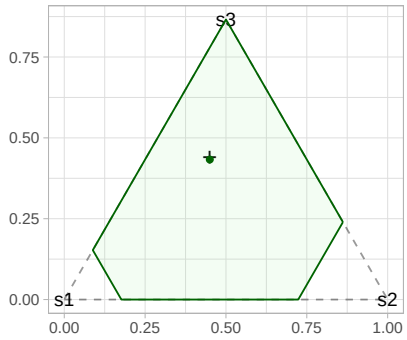

Figure 2.3: An ambiguity set constructed with distribution-free Hoeffding bound for 90% confidence, projected onto a 3 state simplex.

ment $\widehat{T}^{\pi}\hat{v}^{\pi} \leq T^{\pi}\hat{v}$ in 2.4.1 can be restated as:

$$\min_{p \in \mathcal{P}_{s,a}} p^{\mathsf{T}}\hat{v}^{\pi} \leq p_{s,a}^{\mathsf{T}}\hat{v}^{\pi} \ , \tag{2.5}$$

for each state s and action $a = \pi(s)$. It can be readily seen that the inequality above is satisfied when $p_{s,a} \in \mathcal{P}_{s,a}$.

2.4.2 Distribution-free Confidence Region

Distribution-free confidence regions are used widely in reinforcement learning to achieve robustness [19] and to guide exploration [42, 43]. The confidence region is constructed around the mean transition probability by combining the Hoeffding inequality with the union bound [19, 34]. We refer to this set as a *Hoeffding confidence region* and define it as follows for each s and a:

$$\mathcal{P}_{s,a}^{H} = \left\{ p \in \Delta^{S} : \|p - \bar{p}_{s,a}\|_1 \leq \sqrt{\frac{2}{n_{s,a}} \log \frac{SA2^{S}}{\delta}} \right\},$$

where $\bar{p}_{s,a}$ is the mean transition probability computed from $\mathcal{D}$ and $n_{s,a}$ is the number of transitions in $\mathcal{D}$ originating from state s and an action a. Figure 2.3 shows an ambiguity set $\mathcal{P}_{s,a}^{H}$ projected onto a 3-state simplex. For sake of clarity, the requirement that the

probabilities sum to 1 is omitted. The + represents the ground truth and the $\cdot$ nearby is the nominal point. The green shaded region represents the ambiguity set.

Theorem 2.4.2. *The robust value function $\hat{v}_{\mathcal{P}^H}$ for the ambiguity set $\mathcal{P}^H$ satisfies:*

$$\mathbb{P}_{\mathcal{D}}\left[\hat{v}^{\pi}_{\mathcal{P}^H} \leq v^{\pi}_{P^\star},\ \forall \pi \in \Pi \mid P^\star\right] \geq 1 - \delta\ . \tag{2.6}$$

In addition, suppose that $\hat{\pi}^\star_{\mathcal{P}^H}$ is the optimal solution to the robust MDP. Then, $p_0^\mathsf{T}\hat{v}^\star_{\mathcal{P}^H}$ is a safe *return estimate of $\hat{\pi}^\star_{\mathcal{P}^H}$.*

Proof. The proof is a simple extension of prior results [19]. The first part of the statement follows directly from Lemma A.2.1 and Lemma A.2.3. The second part of the statement follows from the fact that the lower bound property holds uniformly across all policies. □

2.5 Conclusion

In this chapter, we introduced some basic definitions and useful concepts related to the research ideas presented in this dissertation. The latter chapters will build on these ideas and relevant concepts will directly be referred to whenever necessary.

CHAPTER 3

WEIGHTED L_1-NORM BOUNDED AMBIGUITY SETS

3.1 Introduction

Some recent results show that RMDPs with weighted L_1 norms can be solved very efficiently [28]. Motivated by that, this chapter proposes a new approach to optimizing the shape of L_1-norm bounded ambiguity sets for robust-MDPs. We choose problem-specific weights for weighted-L_1 norm to construct ambiguity sets. We also derive new concentration inequalities that extend previous results from the uniform L_1 norm ambiguity sets [34] to weighted L_1 sets. We show that this can be used to provide better high-confidence guarantees on the optimized return. Our proposed methods operate in a batch reinforcement learning setting in which transition probabilities must be estimated from a fixed and limited set of logged data. Our goal is broadly similar to [44] and [45], but we show that our methods apply to both frequentist and Bayesian setting.

Several methods have been proposed in the literature to construct ambiguity sets and to mitigate their sensitivity. One important factor in this regard is the underlying rectangularity assumption [30]. A rectangular ambiguity set leads to a tractable but overly pessimistic solution [36, 46]. Most common methods for constructing rectangular ambiguity sets operate in a classical frequentist setting where the ambiguity sets are defined as a plausible region of deviation from the expectation [47, 48]. This deviation is constrained by an L_p-norm, KL-divergence, ϕ-divergence, or Wasserstein metric [23, 49–51]. In contrast, we consider in this chapter a weighted-L_p-norm where the weights adapt contextually based on the problem.

The remainder of the chapter is organized as follows. Section 3.2 describes the robust

objective to optimize with an ambiguity set. Weight based optimization of ambiguity sets is presented in Section 3.3. We derive the finite-sample guarantees in Section 3.4 and a comprehensive empirical evaluation is presented in Section 3.5. We finally draw the concluding remarks in Section 3.6.

3.2 Research Objective

We want to construct ambiguity sets to maximize the guaranteed return (see Definition 2.3.1) for a given confidence level $1-\delta$. Optimizing for such an ambiguity set for every s and a can be stated as the following conceptual optimization problem:

$$\begin{aligned} \max_{\mathcal{P}_{s,a}} \quad & \min_{p \in \mathcal{P}_{s,a}} \left(r_{s,a} + \gamma\, p^{\mathsf{T}} \hat{v}^{\star} \right) \\ \text{s.t.} \quad & \mathbb{P}\left[p^{\star}_{s,a} \in \mathcal{P}_{s,a}, \forall s \in \mathcal{S}, a \in \mathcal{A} \right] \geq 1-\delta \ . \end{aligned} \tag{3.1}$$

Because the Bellman operator is monotone, maximizing the value of each state individually maximizes the return [45]. The distributionally-constrained optimization problem in (3.1) is intractable [27] and depends on the optimal robust value function $\hat{v}^{\star}$ which is unknown and depends on $\mathcal{P}$. To mitigate these issues, we restrict our attention to optimizing the weights of L_1-norm based ambiguity sets and assume to readily have a rough estimate of $\hat{v}^{\star}$. One particular example of such an estimate of $\hat{v}^{\star}$ is the value function computed from the MDP constructed with nominal transition probabilities.

3.3 Optimizing Ambiguity Set Weights

In this section, we outline the general approach to tackling the desired optimization in (3.1). We relax the problem and use strong duality theory to get bounds that can be optimized tractably.

As noted above, maximizing the guaranteed return can be achieved by maximizing the Bellman update for every state. To this effect, assume some fixed $s \in \mathcal{S}$ and $a \in \mathcal{A}$ and let z

denote an estimate of the optimal robust value function: $z = r_{s,a} + \gamma\,\hat{v}$. The robust Bellman update in (2.4) for s and a then simplifies to:

$$q(z) = \min_{p \in \Delta^{\mathcal{S}}} \left\{ p^{\mathsf{T}} z : \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a} \right\} \ . \tag{3.2}$$

In the remainder of the section, we drop the s, a subscripts when they are obvious from the context.

The impact of the choice of the norm in (3.2) on the value of $q(z)$ is not trivial, and we are not aware of a technique that could be used to optimize it directly. We instead maximize a lower bound on this value that the following theorem establishes.

Theorem 3.3.1. *The estimate of expected next value can be bounded from below as:*

$$q(z) \geq \bar{p}^{\mathsf{T}} z - \min_{\lambda \in \mathbb{R}} \psi \|z + \lambda \mathbf{1}\|_\infty \ , \tag{3.3}$$

where $\|\cdot\|_\infty$ *used in* (3.3) *is the dual norm to the norm* $\|\cdot\|_1$ *in* (3.2)*.*

Recall from Definition (2.0.1) that the *dual norm* is defined as:

$$\|z\|_\star = \sup \left\{ z^\intercal x \ : \ \|x\| \leq 1 \right\} \ .$$

Proof. By relaxing the non-negativity constraints on p, we get the following optimization problem:

$$q(z) \geq \min_{p \in \mathbb{R}^S} \left\{ p^{\mathsf{T}} z : \|p - \bar{p}\| \leq \psi,\ \mathbf{1}^{\mathsf{T}} p = 1 \right\} \ .$$

Here, $\mathbf{1}$ is a vector of all ones of the appropriate size. Dualizing this optimization problem and following algebraic manipulation, detailed in Appendix A.1.2, we get the desired lower bound. □

The lower bound in (3.3) is still hard to optimize. But, as we show next, it has a simpler form for weighted L_1 norm. Choosing any fixed λ also provides a lower bound which, we

also show later, can be readily maximized.

We focus on ambiguity sets defined in terms of weighted L_1 norm, which are defined for *positive* weights $w \in \mathbb{R}^S_{>0}$ as:

$$\|z\|_{1,w} = \sum_{i=1}^{S} w_i |z_i|$$

The dual norms for a weighted L_1 norm is a weighted L_∞ norm as Lemma A.1.1 shows. Using this fact, Theorem 3.3.1 can be specialized to L_1 weighted ambiguity sets as follows.

Corollary 3.3.2 (Weighted L_1 Ambiguity Set)**.** *Suppose that $q(z)$ is defined in terms of a weighted L_∞ norm for some $w > \mathbf{0}$. Then $q(z)$ can be lower-bounded as follows:*

$$\begin{aligned} q(z) &= \min_{p \in \Delta^S} \left\{ p^\mathsf{T} z : \|p - \bar{p}\|_{1,w} \leq \psi \right\} \\ &\geq \bar{p}^\mathsf{T} z - \psi \|z - \lambda \mathbf{1}\|_{\infty, \frac{1}{w}} \end{aligned}$$

for any $\lambda \in \mathbb{R}$. Moreover, when $w = \mathbf{1}$, the bound is tightest when $\lambda = (\max_i z_i + \min_i z_i)/2$ and the bound turns to $q(z) \geq \bar{p}^\mathsf{T} z - \frac{\psi}{2}\|z\|_s$ with $\|\cdot\|_s$ representing the span semi-norm*.*

The optimal λ being a median follows because maximization over λ values is identical to the formulation of the optimization problem for the *quantile regression*.

3.3.1 Optimizing Norm Weights

In this section, we introduce tractable methods that optimize weights w in the ambiguity set in order to maximize $q(z)$.

The objective is to choose weights w that will maximize the lower bound on $q(z)$ established in Corollary 3.3.2 as follows:

$$\max_{w \in \mathbb{R}^S_{++}} \left\{ \bar{p}^\mathsf{T} z - \psi \|z - \bar{\lambda}\mathbf{1}\|_{\infty, \frac{1}{w}} \;:\; \sum_{i=1}^{S} w_i^2 = 1 \right\} \tag{3.4}$$

The value $\bar{\lambda}$ in (3.4) is fixed ahead of time and does not change with w. The constraint

$\sum_{i=1}^{S} w_i^2 = 1$ serves to normalize w in order to preserve the desired robustness guarantees with *the same* ψ. This is because scaling both w and ψ simultaneously by an identical factor leaves the ambiguity set unchanged. This regularization constraint is motivated by the finite-sample guarantees in Section 3.4 and our empirical results.

Next, omitting terms that are constant with respect to w simplifies the optimization to:

$$w^\star \in \operatorname*{argmin}_{w \in \mathbb{R}^S_{++}} \left\{ \|z - \bar{\lambda}\mathbf{1}\|_{\infty, \frac{1}{w}} \ : \ \sum_{i=1}^{S} w_i^2 = 1 \right\} . \tag{3.5}$$

The nonlinear optimization problem in (3.5) is convex and can be, surprisingly, solved *analytically*. Let $b_i = |z_i - \bar{\lambda}|$ for $i = 1, \ldots, S$. Introducing an auxiliary variable t further simplifies the optimization problem:

$$\min_{t, w \in \mathbb{R}^S_{++}} \left\{ t \ : \ t \geq b_i / w_i, \ \sum_{i=1}^{S} w_i^2 = 1 \right\} . \tag{3.6}$$

The constraints $w > \mathbf{0}$ cannot be active (because of $1/w_i$) and may be safely ignored. Then, the convex optimization problem in (3.6) has a linear objective, $S + 1$ variables (w's and t), and $S + 1$ constraints. All constraints are active, therefore, in the optimal solution $w^\star$ [52] which must satisfy:

$$w_i^\star = {}^{b_i}\!/\!\sqrt{\textstyle\sum_{j=1}^{S} b_j^2} \ . \tag{3.7}$$

Since $\sum_i w_i^2 = 1$ implies $\sum_i b_i^2 / t^2 = 1$, we conclude that $t = \sqrt{\sum_i b_i^2}$.

Next, we establish new finite-sample bounds for these new types of ambiguity sets.

3.4 Complexity Analysis and Finite-Sample Guarantees

In this section, we first analyse the time complexity of our method and then we describe new sampling bounds that can be used to construct ambiguity sets that provide desired sampling guarantees. We describe both frequentist and Bayesian methods. The following example demonstrates how different norm weights impact the shape of the ambiguity set.

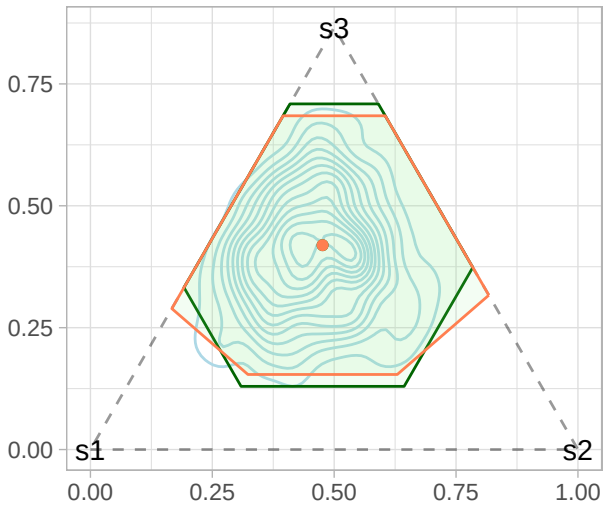

Figure 3.1: A visualization of ambiguity sets for an MDP in 3.4.1.

Example 3.4.1. Consider an MDP with 3 states s_1, s_2, s_3 and a single action a_1. True & unknown transition probability is $P^\star(s_1, a_1, \cdot) = [0.3, 0.2, 0.5]$, and the value function is $v = [0, 0, 1]$. The contours of posterior probability distribution and the ambiguity sets for state s_1 are shown projected onto a simplex in Figure 3.1. The *green set* is constructed with unweighted L_1 norm and the *orange set* is constructed with optimized weights for the L_1 norm. Although both sets have the same probability measure, the weighted set yields a better return estimate for $v^\star$.

Complexity Analysis We have a closed form approach for computing weights to optimize the shape of the ambiguity sets. It therefore does not add much extra computational cost. Robustness can still be handled in $O(S \log S)$ in the weighted set, as presented in Ho et al. [41]. The overall complexity remains tractable and belongs to P-complete.

3.4.1 Bayesian Credible Intervals (BCI)

In Bayesian statistics, credible intervals are comparable to classical confidence intervals [25]. An important advantage of using Bayesian techniques for robust optimization is that they can effectively leverage prior domain knowledge [53].

Russel and Petrik [45] suggest an approach to construct ambiguity regions from credible intervals. The method starts with sampling from the posterior probability distribution of $P^\star$ given data $\mathcal{D}$ to estimate the mean transition probability $\bar{p}_{s,a} = \mathbb{E}_{P^\star}[p^\star_{s,a}|\mathcal{D}]$. Then the smallest possible ambiguity set around the mean is obtained by solving the following optimization problem for each state s and action a:

$$\psi^B_{s,a} = \min_{\psi \in \mathbb{R}_+} \left\{ \psi \ : \ \mathbb{P}\left[\|p^\star_{s,a} - \bar{p}_{s,a}\| > \psi \mid \mathcal{D} \right] < \frac{\delta}{SA} \right\} \ .$$

Finally, the Bayesian ambiguity set can be obtained by:

$$\mathcal{P}^B_{s,a} = \left\{ p \in \Delta^{\mathcal{S}} : \|p - \bar{p}_{s,a}\| \leq \psi^B_{s,a} \right\} \ .$$

This construction applies easily to any form of norm used in the construction of ambiguity sets. That is, it is easy to generalize this method for weighted L_1 ambiguity sets that we study in this work. Algorithm 1 summarizes the steps to construct Bayesian ambiguity sets in quasi-linear time.

Algorithm 1: Weighted Bayesian Credible Intervals (WBCI)

Input: Distribution θ over $p^\star_{s,a}$, confidence level δ, sample count n, weights w
Output: Nominal point $\bar{p}_{s,a}$ and $\psi_{s,a}$

Sample $X_1, \ldots, X_n \in \Delta^S$ from θ: $X_i \sim \theta$;
Nominal point: $\bar{p}_{s,a} \leftarrow (1/n) \sum_{i=1}^n X_i$;
Compute distances $d_i \leftarrow \|\bar{p}_{s,a} - X_i\|_{p,w}$ and sort in *increasing* order ;
$\psi_{s,a} \leftarrow d_{\lceil (1-\delta)n \rceil}$;
return $\bar{p}_{s,a}$ *and* $\psi_{s,a}$;

3.4.2 Weighted Frequentist Confidence Intervals (WFCI)

We present a new finite-sample bound that can be used to construct frequentist ambiguity sets with weighted L_1 norm. This bound is necessary to guarantee high-confidence return guarantees. These results significantly extend the existing bounds which have been limited to the L_1 deviation [34, 35, 45, 54].

Theorem 3.4.1 (Weighted L_1 Error Bound)**.** *Suppose that $\bar{p}_{s,a}$ is the empirical estimate of the transition probability obtained from $n_{s,a}$ samples for some $s \in \mathcal{S}$ and $a \in \mathcal{A}$. If the weights $w \in \mathbb{R}^S_{++}$ are sorted in a non-increasing order $w_i \geq w_{i+1}$, then:*

$$\mathbb{P}\left[E \geq \psi_{s,a}\right] \leq 2 \sum_{i=1}^{S-1} 2^{S-i} \exp\left(-\frac{\psi_{s,a}^2 n_{s,a}}{2w_i^2}\right) ,$$

where $E = \|\bar{p}_{s,a} - p^\star_{s,a}\|_{1,w}$.

Importantly, replacing the sum in the theorem above by a uniform upper bound on w_i would be insufficient to improve ambiguity sets. Theorem A.1.2 further tightens the bound of Theorem 3.4.1 by using Bernstein's inequality in place of Hoeffding's inequality.

The next theorem establishes a new finite-sample bound for weighted L_∞ sets.

Theorem 3.4.2 (Weighted L_∞ Error Bound)**.** *Suppose that $\bar{p}_{s,a}$ is the empirical estimate of the transition probability obtained from $n_{s,a}$ samples for some $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then:*

$$\mathbb{P}\left[E \geq \psi_{s,a}\right] \leq 2 \sum_{i=1}^{S} \exp\left(-2\frac{\psi_{s,a}^2 n_{s,a}}{w_i^2}\right) ,$$

where $E = \|\bar{p}_{s,a} - p^\star_{s,a}\|_{\infty,w}$.

The proofs of both theorems are deferred to Appendix A.1.3.

Theorem 3.4.1 establish the error bounds that can be used to construct ambiguity sets of appropriate size. Unlike with the standard error bound, $\psi_{s,a}$ cannot be determined readily from the bounds analytically. However, since the confidence level function is monotonically increasing, $\psi_{s,a}$ can be easily determined numerically using a bisection method.

Recall that the weights in (3.4) are optimized under a constraint that $\sum_{i=1}^S w_i^2 = 1$ to preserve the confidence guarantee regardless of the weight scales. The constraint is derived from an approximation of the guarantee in Theorem 3.4.2 (similar for Theorem 3.4.1) by

linearizing it from Jensen's inequality:

$$\sum_{i=1}^{S} \exp\left(-2\frac{\psi_{s,a}^2 n_{s,a}}{w_i^2}\right) \approx S \exp\left(-2\frac{1}{S}\sum_{i=1}^{S}\frac{\psi_{s,a}^2 n_{s,a}}{w_i^2}\right) . \tag{3.8}$$

Where '$\approx$' used in (3.8) explicitly denotes that the quantity on the right hand side is an approximation. Then, bounding the right hand side with δ and taking the log and applying Jensen's inequality again gives us:

$$\frac{-1}{2\psi_{s,a}^2 n_{s,a}} \log\left(\frac{\delta}{S}\right) \leq \frac{1}{\frac{1}{S}\sum_{i=1}^{S} w_i^2} .$$

Therefore, a constant value of $\sum_{i=1}^{S} w_i^2$ provides an upper bound on the confidence in the equation above. We emphasize that this is not a bound but rather an approximation due to the linearization step.

3.5 Empirical Evaluation

In this section, we empirically evaluate the advantage of using weighted ambiguity sets in Bayesian and frequentist settings. We assess L_1-bounded ambiguity sets, both with weights and without weights. We include the results derived for L_∞ norm for the completeness of the evaluation. We compare Bayesian credible regions with frequentist's Hoeffding and Bernstein style sets. We start by assuming a true underlying model that produces the simulated datasets containing 100 samples for each state and action. The frequentist methods use these datasets to construct an ambiguity set. Bayesian methods combine the data with a prior to compute a posterior distribution and then draw $10,000$ samples from the posterior distribution to construct a Bayesian ambiguity set. We use an uninformative uniform prior over the reachable next states for all the experiments unless otherwise specified. This prior is somewhat informative in the sense that it contains the knowledge of non-zero transitions implied by the datasets. The performance of the methods is evaluated by the

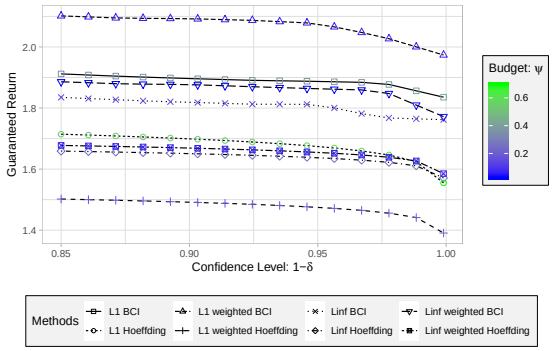

Figure 3.2: Single Bellman Update: the guaranteed return for a monotonic value function $v = [1, 2, 3, 4, 5]$.

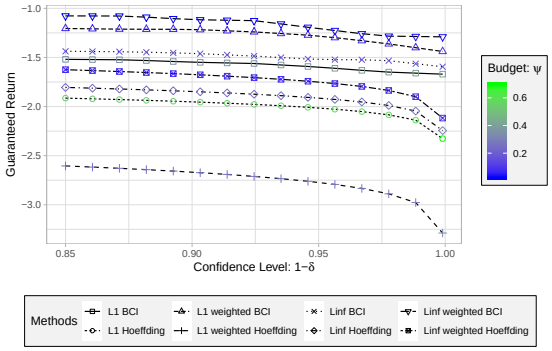

Figure 3.3: Single Bellman Update: the guaranteed return for a sparse value function $v = [0, 0, 0, 0, -5]$.

guaranteed robust returns computed for a range of different confidence levels. We strengthen the weighted L_1 error bound by a factor of two to match with the unweighted one.

Single Bellman Update. In this experiment, we set up a very trivial problem to meticulously examine our proposed method. We consider a transition from a single state s_0 and an action a_0 leading to 5 terminal states $s_1, \ldots, s_5$. The value functions are assumed to be fixed and known. The prior is uniform Dirichlet over the next states. Figure 3.2 and Figure 3.3 show a comparison of average guaranteed returns for 100 independent trials for different value functions. The weighted methods outperform unweighted methods in all instances. Also, the weighted BCI methods are significantly better than other frequentist methods.

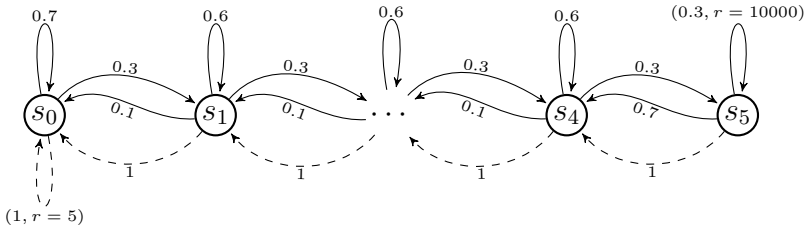

Figure 3.4: RiverSwim problem with six states and two actions (left-dashed arrow, right-solid arrow). The agent starts in either s_1 or s_2.

		Confidence = 0.5		Confidence = 0.95	
	Methods	Uniform	Weighted	Uniform	Weighted
Bayesian	L_1 BCI	5290	**23155**	1152	**15814**
	L_∞ BCI	5290	20673	1152	13142
Frequentist	L_1 Hoeffding	490	634	490	490
	L_1 Bernstein	490	490	490	490
	L_∞Hoeffding	490	**7976**	490	**4183**

Table 3.1: Guaranteed robust return for the RiverSwim experiment.

RiverSwim. We consider the standard RiverSwim [55] domain shown in Figure 3.4 for evaluating our methods. The process follows by sampling synthetic datasets from the true model and then computing the guaranteed robust returns for different methods. We use a uniform Dirichlet distribution over the next states as prior. Table 3.1 summarizes the results. All the weighted methods dominate unweighted methods, and the weighted L_1 BCI method provides the highest guaranteed return. The return of the optimal policy for the true model is $56,687$. At the 50% confidence level, the gap between the optimal return and guaranteed return is reduced by 34% and 13% for weighted L_1 BCI and weighted L_∞ Hoeffding sets respectively over the standard uniform weight sets.

Population Growth Model. We also apply our method in an exponential population growth model [6]. Our model constitutes a simple state-space with exponential dynamics. At each time step, the land manager has to decide whether to apply a control measure to

		Confidence=0.5		Confidence=0.95	
	Methods	Uniform	Weighted	Uniform	Weighted
Bayesian	L_1 BCI	-98659	**-9356**	-108009	**-11307**
	L_∞ BCI	-132781	-35934	-137053	-51834
Frequentist	L_1 Hoeffding	-116167	-106078	-118684	-109301
	L_1 Bernstein	-133712	-129420	-134680	-130826
	L_∞Hoeffding	-132737	**-31761**	-133938	**-46332**

Table 3.2: Guaranteed robust return for the Population experiment.

reduce the growth rate of the species. We refer to [5] for more details of the model. The results are summarized in Table 3.2. Returns for all the methods are negative, which implies a high management cost. Policies computed with frequentist and unweighted methods yield a very high cost. Bayesian and weighted methods significantly outperform other methods. The return of the optimal policy for the true model is $18,448$. At the 50% confidence level, the gap between the optimal return and guaranteed return is reduced by over 75% for both weighted L_1 BCI and weighted L_∞ Hoeffding over the standard uniform weight.

		Confidence=0.5		Confidence=0.95	
	Methods	Uniform	Weighted	Uniform	Weighted
Bayesian	L_1 BCI	310	**428**	291	**414**
	L_∞ BCI	177	278	153	258
Frequentist	L_1 Hoeffding	192	245	180	238
	L_1 Bernstein	121	200	106	188
	L_∞Hoeffding	132	**255**	117	**242**

Table 3.3: Guaranteed robust return for the Inventory experiment.

Inventory Management Problem. Next, we take the classic inventory management problem [4]. The inventory level is discrete and limited by the number of states S. The purchase cost, sale price, and holding cost are $2.49, 3.99$, and 0.03 respectively. The demand is sampled from a normal distribution with a mean $S/4$ and a standard deviation of $S/6$. The initial state is 0 (empty stock). Table 3.3 summarizes the computed guaranteed returns of different methods at 0.5 and 0.95 confidence levels. The guaranteed returns computed with Bayesian and weighted methods are significantly higher than other methods in this problem domain. The return of the optimal policy for the true model is 550. At the 50% confidence level, the gap between the optimal return and guaranteed return is reduced by 50% and 30% for weighted L_1 BCI and weighted L_∞ Hoeffding sets respectively over the standard uniform weight.

	Methods	Confidence=0.5		Confidence=0.95	
		Uniform	Weighted	Uniform	Weighted
Bayesian	L_1 BCI	41.11	47.33	40.48	47.29
	L_∞ BCI	39.95	**47.48**	38.94	**47.44**
Frequentist	L_1 Hoeffding	9.89	45.11	9.14	45.09
	L_1 Bernstein	1.01	44.26	1.00	44.38
	L_∞Hoeffding	37.52	**47.35**	36.94	**47.31**

Table 3.4: Guaranteed robust return for the Cart-Pole experiment.

Cart-Pole. We evaluate our method on Cart-Pole, a standard RL benchmark problem [2, 56]. We collect samples of 100 episodes from the true dynamics. We fit a linear model with that dataset to generate synthetic samples and aggregate nearby states on a resolution of 200 using K-nearest neighbor strategy. The results are summarized in 3.4. Again, in this case, all the Bayesian and weighted methods outperform other methods. The return of the optimal policy for the true model is 51. At the 50% confidence level, the gap between the optimal return and guaranteed return is reduced by 64% and 71% for weighted L_∞ BCI and weighted L_∞ Hoeffding sets respectively over the standard uniform weight.

3.6 Contributions

In this chapter, I proposed a novel approach for optimizing the shape of the L_1-norm bounded ambiguity sets with weights, which goes beyond the conventional L_1-constrained ambiguity sets studied in the literature. This was a joint project with Bahram Behzadian, who proposed a similar method for L_∞-norm bounded ambiguity sets. We together show that the optimal shape of an ambiguity set is problem dependent and is driven by the characteristics of the value function. I derived new finite sample guarantees for the weighted L_1-norm and empirically validated the performance against other baseline methods. The whole work has been done under close supervision of my advisor, and later with Chin Pang Ho. An earlier version of this work was presented at NeurIPS 2019 Workshop on Safety and Robustness in Decision Making. The full paper was published at The 24th International Conference on

Artificial Intelligence and Statistics (AISTATS 2021).

CHAPTER 4

DATA-DRIVEN BAYESIAN AMBIGUITY SETS FOR RMDPS

In this chapter, we argue that constructing ambiguity sets as confidence regions leads to solutions that are unnecessarily conservative. Confidence regions inherently provide robust guarantees for *all* policies and *all* value functions *simultaneously*. It is sufficient, instead, to provide the guarantees for the optimal RMDP policy and value function. Our algorithm (RSVF) provides a tighter lower bound on the return of the optimal policy by interleaving RMDP computations with optimizing the *size* and the *position* of ambiguity sets. Using (hierarchical) Bayesian models helps to further tighten the lower bounds by leveraging prior domain knowledge. We also derive new L_1 concentration inequalities of possible independent interest.

Gupta [57] also constructs ambiguity sets that are not confidence regions. However, their setting and objectives are markedly different from ours and do not readily apply to RMDPs. In general, Bayesian methods for constructing ambiguity sets for RMDPs are not yet understood well and have received only limited attention [58].

Confidence regions derived from concentration inequalities have been used previously to compute bounds on the true return in off-policy policy evaluation [20, 59]. These methods, unfortunately, do not readily generalize to the policy optimization setting, which we target. Other work has focused reducing variance rather than on high-probability bounds [21,22,60]. Methods for exploration in reinforcement learning, such as MBIE or UCRL2, also construct ambiguity sets using concentration inequalities [42, 54, 61, 61, 62] and compute optimistic (upper) bounds to guide exploration.

The following example will be used throughout this chapter to demonstrate the proposed methods and visualize ambiguity sets.

Example 4.0.1. Consider an RMDP with 3 states: s_1, s_2, s_3 and a single action a_1. Assume that the true transition probability is $P^\star(s_1, a_1, \cdot) = [0.3, 0.2, 0.5]$. In $\mathcal{D}$, there are 3 occurrences of transitions (s_1, a_1, s_1), 2 of transitions (s_1, a_1, s_2), and 5 of transitions (s_1, a_1, s_3). The prior distribution over $p^\star_{s_1,a_1}$ is Dirichlet with concentration parameters $\alpha = (1, 1, 1)$. 4.1 depicts ambiguity sets for state s_1 and action a_1. The plus sign marks $p^\star_{s_1,a_1}$, while the dot marks the nominal point of the ambiguity set; the contours indicate the density of the posterior Dirichlet distribution.

The remainder of the chapter is organized as follows. Section 4.1 outlines the approach of constructing ambiguity sets as Bayesian credible region. Section 4.2 describes the main contribution of this chapter, RSVF, a new method for constructing tight ambiguity sets from Bayesian models that are adapted to the optimal policy. RSVF provides tighter robustness guarantees without using confidence regions, which is justified in Section 4.3. Finally, Section 4.4 presents empirical results on several problem domains.

4.1 Bayesian Credible Region (BCI)

We now describe how to construct ambiguity sets from Bayesian credible (or confidence) regions. To the best of our knowledge, this approach has not been studied in depth previously. The construction starts with a (hierarchical) Bayesian model that can be used to sample from the posterior probability of $P^\star$ given data $\mathcal{D}$. The implementation of the Bayesian model is irrelevant as long as it generates posterior samples efficiently. For example, one may use a Dirichlet posterior, or use MCMC sampling libraries like JAGS, Stan, or others [63].

The posterior distribution is used to optimize for the *smallest* ambiguity set around the mean transition probability. Smaller sets, for a fixed nominal point, are likely to result in

less conservative robust estimates. The BCI ambiguity set is defined as follows:

$$\mathcal{P}^B_{s,a} = \left\{ p \in \Delta^S \ : \ \|p - \bar{p}_{s,a}\|_1 \leq \psi^B_{s,a} \right\} \ ,$$

where nominal point is $\bar{p}_{s,a} = \mathbb{E}_{P^\star}[p^\star_{s,a} \mid \mathcal{D}]$.

There is no closed-form expression for the Bayesian ambiguity set size. It must be computed by solving the following optimization problem for each state s and action a:

$$\psi^B_{s,a} = \min_{\psi \in \mathbb{R}_+} \left\{ \psi \ : \ \mathbb{P}\left[\|p^\star_{s,a} - \bar{p}_{s,a}\|_1 > \psi \mid \mathcal{D} \right] < \frac{\delta}{SA} \right\} \ .$$

The nominal point $\bar{p}_{s,a}$ is fixed (not optimized) to preserve tractability. This optimization problem can be solved by the Sample Average Approximation (SAA) algorithm [64]. The main idea is to sample from the posterior distribution and then choose the minimal size $\psi_{s,a}$ that satisfies the constraint. Algorithm 2, summarizes the sort-based method.

We assume that it is possible to draw enough samples from $P^\star$ that the sampling error becomes negligible. Because the finite-sample analysis of SAA is simple but tedious, we omit it in the interest of clarity.

The Bayesian ambiguity sets guarantee safe estimates.

Theorem 4.1.1. *The robust value function $\hat{v}_{\mathcal{P}^B}$ for the ambiguity set $\mathcal{P}^B$ satisfies:*

$$\mathbb{P}_{P^\star}\left[\hat{v}^\pi_{\mathcal{P}^B} \leq v^\pi_{P^\star}, \ \forall \pi \in \Pi \mid \mathcal{D} \right] \geq 1 - \delta \ .$$

Algorithm 2: Bayesian Credible Interval (BCI)

Input: Distribution θ over $p^\star_{s,a}$, confidence level δ, sample count m
Output: Nominal point $\bar{p}_{s,a}$ and L_1 norm size $\psi_{s,a}$

Sample $X_1, \ldots, X_m \in \Delta^S$ from θ: $X_i \sim \theta$;
Nominal point: $\bar{p}_{s,a} \leftarrow (1/m) \sum_{i=1}^m X_i$;
Compute distances $d_i \leftarrow \|\bar{p}_{s,a} - X_i\|_1$ and sort *increasingly*;
Norm size: $\psi_{s,a} \leftarrow d_{(1-\delta)\,m}$;
return $\bar{p}_{s,a}$ *and* $\psi_{s,a}$;

In addition, suppose that $\hat{\pi}^\star_{\mathcal{P}^B}$ is the optimal solution to the robust MDP. Then, $p_0^\mathsf{T}\hat{v}^\star_{\mathcal{P}^B}$ is a safe *return estimate of $\hat{\pi}^\star_{\mathcal{P}^B}$.*

Proof. The first part of the statement follows directly from Lemma A.2.2 and the definition of $\psi^B_{s,a}$. The second part of the statement follows from the fact that the lower bound property holds uniformly across all policies. □

This theorem only proves that the constructed lower bound on the return is safe. It does not address the tightness of the bound.

BCI ambiguity sets $\mathcal{P}^B$ can be much less conservative than Hoeffding set $\mathcal{P}^H$, given informative priors, but also involve greater computation complexity. Next, we further improve on BCI.

4.2 Optimized Bayesian Ambiguity Sets

In this section, we describe the new algorithm for constructing Bayesian ambiguity sets that can compute less-conservative lower bounds on the return. RSVF (robustification with sensible value functions) is a Bayesian method that uses samples from the posterior distribution over $P^\star$ to construct tight ambiguity sets.

Before describing the algorithm, we use the setting of Example 4.0.1 to motivate our approach. To minimize distractions by technicalities, assume that the goal is to compute the return for a *single* time step starting from state s_1. Assume also that the value function $v = (1, 0, 0)$ is known, all rewards from s_1 are 0, and $\gamma = 1$. Recall that our goal is to construct a safe return estimate $\tilde{\rho}(\pi)$ of $\mathrm{V@R}^{0.1}_{P^\star}[\rho(\pi, P^\star)]$ at the 90% level. When the value function is known, it is possible to construct the *optimal* ambiguity set $\mathcal{P}^\star$ such that $\hat{\rho}(\pi) = \min_{p\in\mathcal{P}^\star} p^\mathsf{T} v = \mathrm{V@R}^{0.1}_{P^\star}[\rho(\pi, P^\star)]$ as:

$$\mathcal{P}^\star = \left\{ p \in \Delta^3 \ : \ p^\mathsf{T} v \geq \mathrm{V@R}^{0.1}_{P^\star}[\rho(\pi, P^\star)] \right\} .$$

It can be shown readily that this ambiguity set is optimal in the sense that any set for which

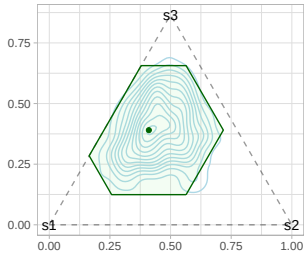

Figure 4.1: Contours of the posterior distribution and the 90%-confidence region.

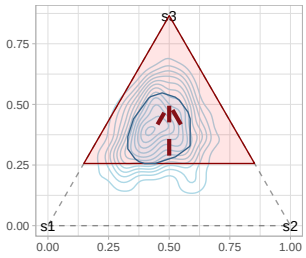

Figure 4.2: Optimal Bayesian ambiguity set (red) for a value function $v = (0, 0, 1)$.

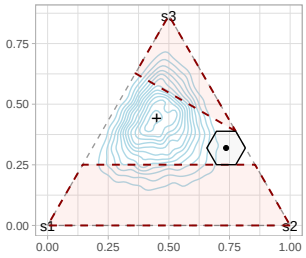

Figure 4.3: Sets $\mathcal{K}_{s_1,a_1}(v_i)$ (dashed red) for $i = 1, 2$ and $\mathcal{L}_{s_1,a_1}(\{v_1, v_2\})$ (black).

$\tilde{\rho}(\pi)$ is exact must be a subset of $\mathcal{P}^\star$ [57]. Figure 4.2 depicts the optimal ambiguity set along with the arrow that indicates the direction along which v increases.

The optimal ambiguity set described above cannot be used directly, unfortunately, because the value function is unknown. It would be tempting to construct the ambiguity set as the *intersection* of optimal sets for all possible value functions; a polyhedral approximation of this set is shown in Figure 4.2 using a blue color. Unfortunately, this approach is not (usually) correct and will not lead to a safe return estimate. This can be shown from the fact that support functions to convex sets are convex and V@R is not a convex (concave) function [65, 66]; see [57] for a more detailed discussion.

Since it is not possible, in general, to simply consider the intersection of optimal ambiguity sets for all possible value functions, we approximate the optimal ambiguity set for a few reasonable value functions. For this purpose, we use a set $\mathcal{K}_{s,a}(v)$, which is almost a complement to the optimal ambiguity set, and is defined as follows:

$$\begin{aligned} \mathcal{K}_{s,a}(v) &= \left\{p \in \Delta^S \mid p^\mathsf{T} v \leq g_{s,a}(v)\right\} \\ g_{s,a}(v) &= \max\left\{g \mid \mathbb{P}_{P^\star}[g \leq (p^\star_{s,a})^\mathsf{T} v \mid \mathcal{D}] \geq \zeta\right\} \ , \end{aligned} \tag{4.1}$$

where $\zeta = 1 - \delta/(SA)$. The lower dashed set in Figure 4.3 depicts this set $\mathcal{K}$ for $v = (0, 0, 1)$ in Example 4.0.1.

The next lemma formalizes the safety-sufficiency of $\mathcal{K}$. Note that the rewards $r_{s,a}$ are not

a factor in this lemma because they are certain and cancel out.

Lemma 4.2.1. *Consider any ambiguity set $\mathcal{P}_{s,a}$ and a value function v. Then $\min_{p\in\mathcal{P}_{s,a}} p^\mathsf{T} v \leq (p^\star_{s,a})^\mathsf{T} v$ with probability $1-\delta/(SA)$ if and only if $\mathcal{P}_{s,a}\cap\mathcal{K}_{s,a}(v)\neq\emptyset$.*

Proof. To show the "if" direction, let $\hat{p}\in\mathcal{P}_{s,a}\cap\mathcal{K}_{s,a}(v)$. Such $\hat{p}$ exists because the intersection is nonempty. Then, $\min_{p\in\mathcal{P}_{s,a}} p^\mathsf{T} v \leq \hat{p}^\mathsf{T} v \leq g_{s,a}(v)$. By definition, $g_{s,a}(v)\leq (p^\star_{s,a})^\mathsf{T} v$ with probability $1-\delta/(SA)$.

To show the "only if" direction, suppose that $\hat{p}$ is a minimizer in $\min_{p\in\mathcal{P}_{s,a}} p^\mathsf{T} v$. The premise translates to $\mathbb{P}_{P^\star}[\hat{p}^\mathsf{T} v \leq (p^\star_{s,a})^\mathsf{T} v \mid \mathcal{D}] \geq 1-\delta/(SA)$. Therefore, $g_{s,a}(v)\geq \hat{p}^\mathsf{T} v$ and $\hat{p}\in\mathcal{P}_{s,a}\cap\mathcal{K}_{s,a}$ and the intersection is non-empty. □

If any ambiguity set $\mathcal{P}_{s,a}$ intersects $\mathcal{K}_{s,a}(\hat{v}^\pi_\mathcal{P})$ for each state s,a then the value function $\hat{v}^\pi_\mathcal{P}$ is safe. This is sufficient, when the value function is known, but we need to generalize the approach to a setting in which the value function is one of many possible ones. The set $\mathcal{L}_{s,a}(\mathcal{V})$ provides such a guarantee for a set of possible value functions (POV) $\mathcal{V}$. Its center is chosen to minimize its size while intersecting $\mathcal{K}_{s,a}(v)$ for each v in $\mathcal{V}$ and is constructed as follows.

$$\begin{aligned}
&\mathcal{L}_{s,a}(\mathcal{V}) = \left\{p\in\Delta^S \mid \|p-\theta_{s,a}(\mathcal{V})\|_1 \leq \psi_{s,a}(\mathcal{V})\right\}\\
&\psi_{s,a}(\mathcal{V}) = \min_{p\in\Delta^S} f(p), \quad \theta_{s,a}(\mathcal{V})\in\arg\min_{p\in\Delta^S} f(p), \quad f(p)=\max_{v\in\mathcal{V}}\min_{q\in\mathcal{K}_{s,a}(v)}\|q-p\|_1
\end{aligned} \tag{4.2}$$

The optimization in (4.2) can be represented and solved as a linear program and accelerated using coordinate minimization techniques. Figure 4.3 shows the set $\mathcal{L}$ in black solid color. It is the smallest set that intersects the two $\mathcal{K}$ sets for value functions $v_1=(0,0,1)$ and $v_2=(2,1,0)$ in 4.0.1. The following lemma formalizes the properties of $\mathcal{L}_{s,a}$.

Lemma 4.2.2. *For any finite set $\mathcal{V}$ of value functions, the following inequality holds for all $v\in\mathcal{V}$ simultaneously:*

$$\mathbb{P}_{P^\star}\left[\min_{p\in\mathcal{L}_{s,a}(\mathcal{V})} p^\mathsf{T} v \leq (p^\star_{s,a})^\mathsf{T} v \;\middle|\; \mathcal{D}\right] \geq 1-\frac{\delta}{SA}\ .$$

Proof. Assume an arbitrary $v \in \mathcal{V}$ and let $q_v^\star \in \arg\min_{q \in \mathcal{K}_{s,a}(v)} \|q - \theta_{s,a}(\mathcal{V})\|_1$ using the notation of (4.2). From the definition of $\theta_{s,a}(\mathcal{V})$ in (4.2), the value q_v is in the ambiguity set $\mathcal{L}_{s,a}(\mathcal{V})$. Given that also $q_v \in \mathcal{K}_{s,a}(v)$, Lemma 4.2.1 shows that:

$$\mathbb{P}_{P^\star}\left[\min_{p \in \mathcal{L}_{s,a}(\mathcal{V})} p^\mathsf{T} v \leq (p_{s,a}^\star)^\mathsf{T} v \;\middle|\; \mathcal{D}\right] \geq 1 - \frac{\delta}{SA} \;,$$

because $q_v \in \mathcal{L}_{s,a}(v) \cup \mathcal{K}_{s,a}(v) \neq \emptyset$. This completes the proof since v is any from $\mathcal{V}$. □

We are now ready to describe RSVF, which is outlined in Algorithm 3. RSVF takes an optimistic approach to approximating the optimal ambiguity set. It starts with a small set of potential optimal value functions (POV) and constructs an ambiguity set that is safe for these value functions. It keeps increasing the POV set until $\hat{v}^\star$ is in the set and the policy is safe.

Algorithm 3: RSVF: Adapted Ambiguity Sets

Input: Confidence $1 - \delta$ and posterior $\mathbb{P}_{P^\star}[\cdot \mid \mathcal{D}]$
Output: Policy π and lower bound $\tilde{\rho}(\pi)$

```
k ← 0;
Pick some initial value function v̂_0;
Initialize POV: V_0 ← ∅ ;
repeat
    Augment POV: V_{k+1} ← V_k ∪ {v_k} ;
    For all s, a update P_{s,a}^{k+1} ← L_{s,a}(V_{k+1}) ;
    Solve v̂_{k+1} ← v̂*_{P_{k+1}} and π̂_{k+1} ← π̂*_{P_{k+1}};
    k ← k + 1 ;
until safe for all s, a: K_{s,a}(v̂_k) ∩ P_{s,a}^k ≠ ∅;
return (π̂_k, p_0^T v̂_k) ;
```

The following theorem states that 3 produces a safe estimate of the true return.

Theorem 4.2.3. *Suppose that 3 terminates with a policy $\hat{\pi}_k$ and a value function $\hat{v}_k$ in the iteration k. Then, the return estimate $p_0^\mathsf{T} \hat{v}_k$ is safe:*

$$\mathbb{P}_{P^\star}\left[p_0^\mathsf{T} \hat{v}_k \leq p_0^\mathsf{T} v_{P^\star}^{\hat{\pi}_k} \;\middle|\; \mathcal{D}\right] \geq 1 - \delta \;.$$

Proof. Recall that Algorithm 3 terminates only if $\mathcal{K}_{s,a}(\hat{v}_k) \cap \mathcal{P}^k_{s,a} \neq \emptyset$ for each state s and action a. Then, according to Lemma 4.2.1, we get with probability $1 - \delta/(SA)$:

$$\min_{p \in \mathcal{P}^k_{s,a}} p^\mathsf{T} \hat{v}_k \leq (p^\star_{s,a})^\mathsf{T} \hat{v}_k$$

for any fixed state s and action a. By the union bound, the inequality holds simultaneously for all states and actions with probability $1 - \delta$. That means that with probability $1 - \delta$ we can derive the following using basic algebra:

$$\begin{aligned}
\min_{p \in \mathcal{P}^k_{s,a}} p^\mathsf{T} \hat{v}_k &\leq (p^\star_{s,a})^\mathsf{T} \hat{v}_k && \forall s \in \mathcal{S}, a \in \mathcal{A} \\
r_{s,a} + \min_{p \in \mathcal{P}^k_{s,a}} p^\mathsf{T} \hat{v}_k &\leq r_{s,a} + (p^\star_{s,a})^\mathsf{T} \hat{v}_k && \forall s \in \mathcal{S}, a \in \mathcal{A} \\
\widehat{T}^{\hat{\pi}_k}_{\mathcal{P}^k} \hat{v}_k &\leq T^{\hat{\pi}_k}_{P^\star} \hat{v}_k
\end{aligned}$$

Note that $\hat{v}_k$ is the robust value function for the policy $\hat{\pi}_k$ since $\hat{v}_k = \hat{v}^\star_{\mathcal{P}_k}$ and $\hat{\pi}_k = \hat{\pi}^\star_{\mathcal{P}_k}$. Proposition 2.4.1 finally implies that $\hat{v}_k \leq v^{\hat{\pi}_k}_{P^\star}$ with probability $1 - \delta$. □

The proof above is technical but conceptually simple. It is based on two main properties. The first one is the construction of optimal ambiguity sets for the known value function as outlined above. The second is the fact that the ambiguity set needs to be robust with respect to the robust value function $\hat{v}$ and *not* the optimal value function $v^\star$. This is subtle, but *crucial* since $\hat{v}$ is a constant while $v^\star$ is a random variable in the Bayesian setting. The RSVF approach, therefore, does not work when frequentist guarantees are required. Confidence regions, described in Section 2.4.1, are designed for situations when robustness is required with respect to a random variable, and are therefore overly conservative in our setting. See Section 4.3 for more in-depth discussion.

It is however important to mention its limitations. This result shows only that the return estimate $\hat{\rho}$ is safe; it does not show that it is good. There are, of course, naive safe estimates such as $\tilde{\rho}(\pi) = (1-\gamma)^{-1} \min_{s,a} r_{s,a}$. Since RSVF tightly approximates the optimal ambiguity

sets, we expect it to perform significantly better and we present empirical evidence of it in Section 4.4. RSVF, as described in 3, is not guaranteed to terminate. To terminate after a specific number of iterations, the algorithm can simply fall back to the BCI sets for states and actions for which the termination condition is not satisfied. Line 6 of Algorithm 3 is formulated and solved as a linear program and therefore is polynomial time. Line 7 computes robust value function and robust policy and is known to be polynomial time operation [30,41]. Algorithm 3 therefore belongs to *P-complete* class.

4.3 Why Not Confidence Regions

Constructing ambiguity sets from confidence regions seems intuitive and natural. It may be surprising that RSVF abandons this intuitive approach. In this section, we describe two reasons why confidence regions are unnecessarily conservative compared to RSVF sets.

The first reason why confidence regions are too conservative is because they assume that the value function depends on the true model $P^\star$. To see this, consider the setting of Example 4.0.1 with $r_{s_1,a_1} = 0$. When an ambiguity set $\mathcal{P}_{s_1,a_1}$ is built as a confidence region such that $\mathbb{P}[p^\star_{s_1,a_1} \in \mathcal{P}_{s_1,a_1}] \geq 1-\delta$, it satisfies:

$$\mathbb{P}_{P^\star}\left[\min_{p\in\mathcal{P}_{s,a}} p^\mathsf{T} v \leq (p^\star_{s,a})^\mathsf{T} v,\ \forall v \in \mathbb{R}^S \;\middle|\; \mathcal{D}\right] \geq 1-\delta.$$

Notice the value function inside of the probability operator. Proposition 2.4.1 shows that this guarantee is needlessly strong. It is, instead, sufficient that the inequality (2.5) holds just for $\hat{v}^\pi$ which is independent of $P^\star$ in the Bayesian setting. The following weaker condition is sufficient to guarantee safety:

$$\mathbb{P}_{P^\star}\left[\min_{p\in\mathcal{P}_{s,a}} p^\mathsf{T} v \leq (p^\star_{s,a})^\mathsf{T} v \;\middle|\; \mathcal{D}\right] \geq 1-\delta,\ \forall v \in \mathbb{R}^S \tag{4.3}$$

Notice that v is outside of the probability operator. This set is smaller and provides the same guarantees, but may be more difficult to construct [57].

The second reason why confidence regions are too conservative is because they construct a uniform lower bound for all policies π as is apparent in Theorem 4.1.1. This is unnecessary, again, as 2.4.1 shows. The robust Bellman update only needs to lower bound the Bellman update for the computed value function $\hat{v}^{\pi}$, not for all value functions. As a result, (4.3), can be further relaxed to:

$$\mathbb{P}_{P^\star}\left[\min_{p\in\mathcal{P}_{s,a}} p^{\mathsf{T}}\hat{v}^{\pi_R} \leq (p^\star_{s,a})^{\mathsf{T}}\hat{v}^{\pi_R} \;\middle|\; \mathcal{D}\right] \geq 1-\delta, \tag{4.4}$$

where π_R is the optimal solution to the robust MDP. RSVF is less conservative because it constructs ambiguity sets that satisfy the weaker requirement of (4.4) rather than confidence regions.

4.4 Empirical Evaluation

In this section, we empirically evaluate the safe estimates computed using Hoeffding, BCI, and RSVF ambiguity sets. We start by assuming a true model and generate simulated datasets from it. Each dataset is then used to construct an ambiguity set and a safe estimate of policy return. The performance of the methods is measured using the average of the absolute errors of the estimates compared with the true returns of the *optimal* policies. All of our experiments use a 95% confidence for the safety of the estimates.

We compare ambiguity sets constructed using BCI, RSVF, with the Hoeffding sets. To reduce the conservativeness of Hoeffding sets when transition probabilities are sparse, we use a modification inspired by the Good-Turing bounds [42]. The modification is to assume that any transitions from s, a to s' are impossible if they are missing in the dataset $\mathcal{D}$. We also compare with the "Hoeffding Monotone" formulation $\mathcal{P}^T$ even when there is no guarantee that the value function is really monotone. This helps us to quantify the limitations of using concentration inequalities. Finally, we compare the results with the "Mean Transition" which solves the expected model $\bar{p}_{s,a}$ and provides no safety guarantees.

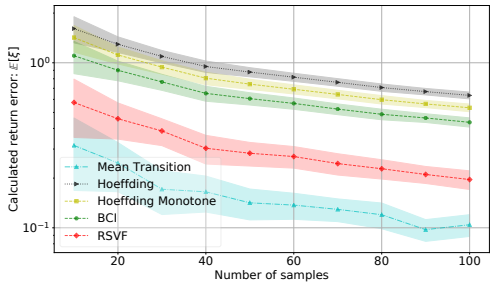

Figure 4.4: Expected regret of safe estimates with 95% confidence regions for the Bellman update with an uninformative Dirichlet prior.

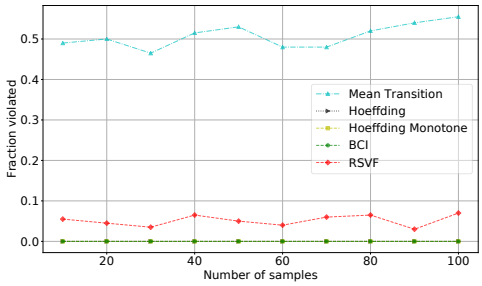

Figure 4.5: Rate of violations of the safety requirement with 95% confidence regions for the Bellman update with an uninformative Dirichlet prior.

We do not evaluate the computational complexity of the methods since they target problems constrained by data and not computation. The Bayesian methods are generally more computationally demanding but the scale depends significantly on the type of the prior model used. All Bayesian methods draw $1,000$ samples from the posterior for each state and action.

4.4.1 Bellman Update

In this section, we consider a transition from a single state s_0 and action a_0 to 5 states $s_1, \ldots, s_5$. The value function for the states $s_1, \ldots, s_5$ is fixed to be $[1, 2, 3, 4, 5]$. RSVF is run for a single iteration with the given value function. The single iteration of RSVF in this simplistic setting helps to quantify the possible benefit of using RSVF-style methods over BCI. The ground truth is generated from the corresponding prior for each one of the problems.

Uninformative Dirichlet Priors This setting considers a uniform Dirichlet distribution with $\alpha = [1, 1, 1, 1, 1]$ as the prior. This prior provides little information. Figure 4.4 compares the computed robust return errors. The value ξ represents the regret of predicted returns, which is the absolute difference between the *true* optimal value and the robust estimate: $\xi = |\rho(\pi^\star_{P^\star}, P^\star) - \tilde{\rho}(\hat{\pi}^\star)|$. Here, $\tilde{\rho}$ is the robust estimate and $\hat{\pi}^\star$ is the optimal robust

solution. The smaller the value, the tighter and less conservative the safe estimate is. Figure 4.5 shows the rate of safety violations: $\mathbb{P}_{\mathcal{D}}[\tilde{\rho}(\hat{\pi}^\star) > \rho(\hat{\pi}^\star, P^\star) \mid P^\star]$. The number of samples is the size of dataset $\mathcal{D}$. All results are computed by averaging over 200 simulated datasets of the given size generated from the ground-truth $P^\star$.

The results show that BCI improves on both types Hoeffding bounds and RSVF further improves on BCI. The mean estimate provides the tightest bounds, but Figure 4.5 demonstrates that it does not provide any meaningful safety guarantees. It also provides insights into how RSVF improves on the other methods. Because the goal is to guarantee estimates are computed with 95% confidence, one would expect the safety guarantees to be violated about 5% of the time. BCI and Hoeffding solutions violate the safety requirements 0% of the time. RSVF is optimal in this setting and meets the allowed 5% violation.

Informative Gaussian Priors To evaluate the effect of using an informative prior, we use a problem inspired by inventory optimization. The states $s_1, \ldots, s_5$ represent inventory levels. The inventory level corresponds to the state index (1 in the state s_1) except that the inventory in the current state s_0 is 5. The demand is assumed to be Normally distributed with an unknown mean μ and a *known* standard deviation $\sigma = 1$. The prior over μ is Normal with the mean $\mu_0 = 3$ and, therefore, the posterior over μ is also Normal. The current action assumes that no product is ordered and, therefore, only the demand is subtracted from s_0.

Figure 4.6 compares the regret of safe estimates which were generated identically to the uninformative example. It shows that with an informative prior, BCI performs significantly better than Hoeffding bounds. RSVF provides still tighter bounds than BCI. The violations plot (not shown) is almost identical to 4.5.

4.4.2 Full MDP

In this section, we evaluate the methods using MDPs with relatively small state-spaces. They can be used with certain types of value function approximation, like aggregation [67], but

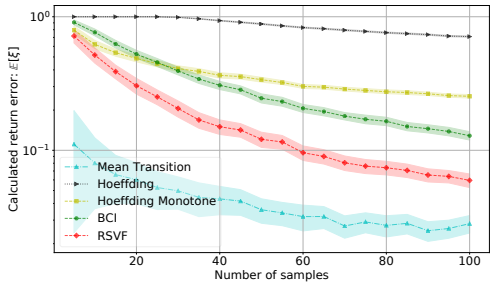

Figure 4.6: Expected regret of safe estimates with 95% confidence regions for the Bellman update with an informative prior.

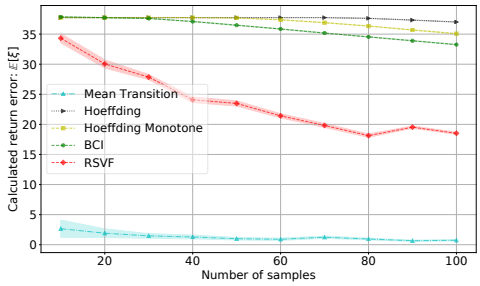

Figure 4.7: Expected regret of safe estimates with 95% confidence regions for the RiverSwim: an MDP with an uninformative prior.

we evaluate them only on tabular problems to prevent approximation errors from skewing the results. To prevent the sampling policy from influencing the results, each dataset $\mathcal{D}$ has the same number of samples from each state.

Uninformative Prior We first use the standard RiverSwim domain for the evaluation [43]. The methods are evaluated identically to the Bellman update above. That is, we generate synthetic datasets from the ground truth and then compare expected regret of the robust estimate with respect to the true return of the *optimal* policy for the ground truth. As the prior, we use the uniform Dirichlet distribution over all states. Figure 4.7 shows the expected robust regret over 100 repetitions. The x-axis represents the number of samples in $\mathcal{D}$ for each state. It is apparent that BCI improves only slightly on the Hoeffding sets since the prior is not informative. RSVF, on the other hand, shows a significant improvement over BCI. All robust methods have safety violations of 0% indicating that even RSVF is unnecessarily conservative here.

Informative Prior Next, we evaluate RSVF on the MDP model of a simple exponential population model [32]. Robustness plays an important role in ecological models because they are often complex, stochastic, and data collection is expensive. Yet, it is important that the decisions are robust due to their long term impacts.

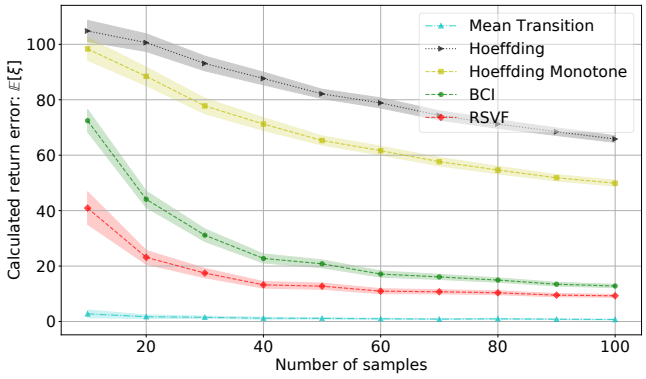

Figure 4.8: Expected regret of safe estimates with 90% confidence regions for the ExpPopulation: an MDP with an informative prior.

We only outline the population model here and refer the interested reader to [**?**] for more details. The population N_t of a species at time t evolves according to the exponential dynamics $N_{t+1} = \min(\lambda_t N_t, K)$. Here, λ is the growth rate and K is the carrying capacity of the environment. A manager must decide, at each time t, whether to apply a treatment that reduces the growth rate λ. The growth rate λ_t is defined as: $\lambda_t = \bar{\lambda} - z_t N_t \beta_1 - z_t \max(0, N_t - \bar{N})^2 \beta_2 + \mathcal{N}(0, \sigma_y^2)$, where β_1 and β_2 are the coefficients of treatment effectiveness and z_t is the indicator of treatment. A noisy estimate y_t of the population N_t is observed: $y_t \sim N_t + \mathcal{N}(0, \sigma_y^2)$. The state in the MDP is the population y_t discretized to 20 values. There are two actions whether to apply the treatment. The rewards capture the costs of high population and the treatment application. The exponential growth model is used as the prior and all priors and posteriors are Normally distributed.

Figure 4.8 shows the average regret of the safe predictions. BCI can leverage the prior information to compute tighter bounds, but RSVF further improves on BCI. The rate of safety violations is again 0% for all robust methods.

4.5 Contributions

In this chapter, I proposed a new Bayesian algorithm for constructing ambiguity sets in RMDPs, improving over standard distribution-free methods. This algorithm is able to in-

corporate prior knowledge and can significantly improve over other existing methods. Most of the theoretical analysis presented in this chapter was done by my advisor. I empirically validated the performance of the proposed RSVF algorithm and compared it to other baseline methods. An earlier version of this chapter was presented at NeurIPS 2018 Workshop on Probabilistic Reinforcement Learning and Structured Control. The full paper was published at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). I also investigate the utility of this RSVF method in online setting for safe exploration purpose and that study was presented at The Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM 2019).

CHAPTER 5
ROBUST CONSTRAINED POLICY OPTIMIZATION

Constrained Markov decision processes (CMDPs) are a super class of general MDPs that incorporate an expected cumulative cost constraints [68] in addition to the regular reward-based objective. The safety constraints imposed in CMDPs are important in real-life applications, where one cannot afford to risk violating some given constraints, e.g., in autonomous cars, there are hard safety constraints on the car velocities and steering angles [69]. The general formulation of CMDPs is specific to the case of known models, and we refer to these CMDPs as *non-robust*. Several solution methods are available for solving non-robust CMDPs: linear programming-based solutions [68], surrogate-based methods [70, 71], Lagrangian methods [68, 72].

In addition to the constrained MDP setup, training for real-world applications often occurs in simulated environments. The result is then transferred to the real world, typically followed by fine-tuning, a process referred to as Sim2Real [73]. The simulator is, by definition, inaccurate with respect to the real-world, due to approximations and lack of system identification [67]. Furthermore, for safety critical applications, a trained policy in simulation should offer certain guarantees about safety when transferred to the real world. Robust MDPs (RMDPs), as described in previous chapters, provide a framework to learn policies that can deal with model inaccuracies and also can provide robustness guarantees. But one noticeable characteristic of RMDPs is the fact that they do not consider any safety constraints as imposed in the CMDP setting.

In light of these practical motivations, we propose in this chapter to unite the two concepts

of CMDPs and RMDPs, to ensure both safety and robustness. In this RCMDP concept, we propose to simultaneously consider the worst-case scenario for both the performance cost, as well as the safety constraints. Such RCMDPs then can certify that the safety constraints are satisfied in the worst-case situation while the performance is also optimized and guaranteed. That is, if deployed, the worst-case objective is optimized while making sure that the worst-case constraint cost will not exceed a pre-determined safety budget with high probability.

The rest of this chapter is organized as follows: Section 5.1 describes the formulation of our Robust-CMDP problem and the objective we seek to optimize. We derive a Bellman-style equation for RCMDPs and propose a gradient based optimization scheme in Section 5.2. We then propose and evaluate a policy-gradient and an actor-critic algorithm in Section 5.3 and draw concluding remarks in Section 5.4.

5.1 Problem Formulation

As described in Chapter 2.4, we consider Robust Markov Decision Processes (RMDPs) with a finite number of states $\mathcal{S} = \{1, \ldots, S\}$ and finite number of actions $\mathcal{A} = \{1, \ldots, A\}$. Every action $a \in \mathcal{A}$ is available for the decision maker to take in every state $s \in \mathcal{S}$. After taking an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the decision maker transitions to a next state $s' \in \mathcal{S}$ according to the *true*, but *unknown*, transition probability $p^\star_{s,a} \in \Delta^{\mathcal{S}}$ and receives a reward $r_{s,a,s'} \in \mathbb{R}$. We use $p_{s,a}$ to denote transition probabilities from $s \in \mathcal{S}$ and $a \in \mathcal{A}$, and condense it to refer to transition function as $p = \big(p_{s,a}\big)_{s\in\mathcal{S},a\in\mathcal{A}} \in \big(\Delta^{\mathcal{S}}\big)^{\mathcal{S}\times\mathcal{A}}$. We condense the rewards to vectors $r_{s,a} = \big(r_{s,a,s'}\big)_{s'\in\mathcal{S}} \in \mathbb{R}^{\mathcal{S}}$ and $r = \big(r_{s,a}\big)_{s\in\mathcal{S},a\in\mathcal{A}}$.

Our RMDP setting assumes that the transition $p_{s,a}$ is chosen adversarially from an ambiguity set $\mathcal{P}_{s,a} \in \big(\Delta^{\mathcal{S}}\big)^{\mathcal{S}\times\mathcal{A}}$ for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$. An ambiguity set $\mathcal{P}_{s,a}$, defined for each state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, is a set of feasible transitions quantifying the uncertainty in transition probabilities. We restrict our attention to $s, a-$rectangular ambiguity sets which simply assumes independence between transition probabilities of different state-action pairs [29, 30]. We define the L_1-norm bounded ambiguity sets around the nominal

transition probability $\bar{p}_{s,a} = \mathbb{E}[p^{\star}_{s,a}|\mathcal{D}]$, for some dataset $\mathcal{D}$ as:

$$\mathcal{P}_{s,a} = \big\{p \in \Delta^S \mid \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a}\big\},$$

where $\psi_{s,a} \geq 0$ is the budget of allowed deviations. As discussed in Section 2.4.2, this budget $\psi_{s,a}$ can be computed for each $s \in \mathcal{S}$, $a \in \mathcal{A}$ using Hoeffding bound [45]: $\psi_{s,a} = \sqrt{\frac{2}{n_{s,a}} \log \frac{SA2^S}{\delta}}$, where $n_{s,a}$ is the number of transitions in dataset $\mathcal{D}$ originating from state s and an action a, and δ is the confidence level. This $\psi_{s,a}$, if used to compute a policy in RMDPs, then guarantees that the computed return is a lower bound with probability δ. Note that this is just one specific choice for the ambiguity set, our method can be extended to any other type of ambiguity sets (e.g. $L_\infty-$norm, Bayesian, weighted, sampling based etc.). We use $\mathcal{P}$ to generally refer to $\mathcal{P}_\tau = \bigotimes_{s_t \in \mathcal{S}, a_t \in \mathcal{A}} \mathcal{P}_{s,a}$, where τ denotes the total number of time steps starting from $T - \tau$, T is the length of the horizon, and $t \in \{T - \tau, T - \tau + 1, \ldots, T\}$. For example, with $\tau = T$ we have $\mathcal{P}_T = \bigotimes_{s_t \in \mathcal{S}, a_t \in \mathcal{A}} \mathcal{P}_{s,a}$ starting from time step 0. This collectively represents the ambiguity set along with the notion of independence between state-action pairs in a tabular setting with discrete states and actions. Sampling based sets under approximate methods (e.g. neural network) for large and continuous problems also extend on this similar notion of ambiguity sets [74, 75].

A stationary randomized policy $\pi(\cdot|s)$ for state $s \in \mathcal{S}$ defines a probability distribution over actions $a \in \mathcal{A}$. Note that, we use a slightly different notation $\pi(\cdot|s)$ to represent randomized policies instead of $\pi(s)$ used in previous chapters to represent deterministic policies. The set of all randomized stationary policies is denoted by $\Pi \in \big(\Delta^{\mathcal{A}}\big)^{\mathcal{S}}$. We parameterize the randomized policy for state $s \in \mathcal{S}$ as $\pi_\theta(\cdot|s)$ where $\theta \subseteq \mathbb{R}^k$ is a $k-$dimensional parameter vector. Let $\xi = \{s_0, a_0, c_0, d_0, \ldots, s_{T-1}, a_{T-1}, c_{T-1}, d_{T-1}, s_T\}$ be a sampled trajectory generated by executing a policy π_θ from a starting state $s_0 \sim p_0$ under transition probabilities $p \in \mathcal{P}$, where p_0 is the distribution of initial states. Then the probability of sampling a trajectory ξ is: $p^{\pi_\theta}(\xi) = p_0(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t)$ and the total reward along the

trajectory ξ is: $g(\xi, r) = \sum_{t=0}^{T-1} \gamma^t r_{s_t, a_t, s_{t+1}}$ [2,3]. The value function $v_p^{\pi_\theta} : \mathcal{S} \to \mathbb{R}$ for a policy π_θ and transition probability p is: $v_p^{\pi_\theta} = \mathbb{E}_{\xi \sim p}\left[g(\xi, r)\right]$ and the total return is:

$$\rho(\pi_\theta, p, r) = p_0^T v_p^{\pi_\theta}$$

Because the RMDP setting considers different possible transition probabilities within the ambiguity set $\mathcal{P}$, we use a subscript p (e.g. $v_p^{\pi_\theta}$) to indicate which one is used, in case it is not clear from the context.

We define a robust value function $\hat{v}_{\mathcal{P}}^{\pi_\theta}$ for an ambiguity set $\mathcal{P}$ as: $\hat{v}_{\mathcal{P}}^{\pi_\theta} = \min_{p \in \mathcal{P}} v_p^{\pi_\theta}$. Similar to ordinary MDPs, the robust value function can be computed using robust Bellman operator as [36, 37]:

$$(\widehat{T}_{\mathcal{P}} v)(s) = \max_{a \in \mathcal{A}} \min_{p \in \mathcal{P}_{s,a}} (r_{s,a} + \gamma \cdot p^{\mathsf{T}} v)$$

The optimal robust value function $\hat{v}^\star$, and the robust value function $\hat{v}_{\mathcal{P}}^{\pi_\theta}$ for a policy π_θ are unique and satisfy $\hat{v}^\star = \widehat{T}_{\mathcal{P}} \hat{v}^\star$ and $\hat{v}_{\mathcal{P}}^{\pi_\theta} = \widehat{T}_{\mathcal{P}}^{\pi_\theta} \hat{v}^{\pi_\theta}$ [36]. The robust return $\hat{\rho}(\pi_\theta, \mathcal{P}, r)$ for a policy π_θ and ambiguity set $\mathcal{P}$ is defined as [37, 76]:

$$\hat{\rho}(\pi_\theta, \mathcal{P}, r) = \min_{p \in \mathcal{P}} \rho(\pi_\theta, p, r) = p_0^T \hat{v}_{\mathcal{P}}^{\pi_\theta}$$

where p_0 is the initial state distribution.

Constrained RMDP (RCMDP) In addition to rewards $r_{s,a}$ for RMDPs described above, we incorporate a constraint cost $d'_{s,a,s'} \in \mathbb{R}$, where $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, representing some kind of constraint on behavior's safety. Consider for example an autonomous car that makes money (reward r) for each complete trip but incurs a big fine (constraint cost d) for traffic violations or a collision. We define the constraint cost $d'_{s,a,s'}$ to be a negative reward $d_{s,a,s'} = -d'_{s,a,s'}$, which brings consistency in representing the *worst-case* with a minimum over the ambiguity set $\mathcal{P}$ for both the objective and the constraint. An associated constraint budget $\beta \in \mathbb{R}_+$ describes the total budget for constraint violations. This

arrangement resembles the constrained-MDP setting as described in [68], but with additional robustness.

Similar to reward based estimates described above, the total constraint cost along a trajectory ξ is: $g(\xi, d) = \sum_{t=0}^{\infty} \gamma^t d_{s_t, a_t, s_{t+1}}$, robust value function for policy π_θ and ambiguity set $\mathcal{P}$ is: $\hat{u}^{\pi_\theta} = \min_{p \in \mathcal{P}} \mathbb{E}_{\xi \sim p} \left[g(\xi, d) \right]$ and the robust return:

$$\hat{\rho}(\pi_\theta, \mathcal{P}, d) = \min_{p \in \mathcal{P}} \rho(\pi_\theta, p, d) = p_0^T \hat{u}^{\pi_\theta}$$

Similar to $\hat{v}^\star$, the optimal constraint value function $\hat{u}^\star$ is also unique and independently satisfies the Bellman optimality equation [68]. We now formally define the objective of Robust Constrained MDP (RCMDP) as below:

$$\underset{\pi_\theta \in \Pi}{\text{maximize}} \quad \hat{\rho}(\pi_\theta, \mathcal{P}, r) \tag{5.1a}$$

$$\text{subject to} \quad \hat{\rho}(\pi_\theta, \mathcal{P}, d) \geq \beta \tag{5.1b}$$

This objective resembles the objective of a CMDP [68], but with additional robustness integrated by the quantification of the uncertainty about the model. The interpretation of the objective is to find a policy π_θ that maximizes the worst-case return estimates, while satisfying the constraints in all possible situations.

5.2 Robust Constrained Optimization

A standard approach for solving the optimization problem (5.1) is to apply the Lagrange relaxation procedure (Chapter 3 of [52]), which turns it into an unconstrained optimization problem:

$$\mathfrak{L}(\pi_\theta, \lambda) = \hat{\rho}(\pi_\theta, \mathcal{P}, r) - \lambda \Big(\beta - \hat{\rho}(\pi_\theta, \mathcal{P}, d) \Big) \tag{5.2}$$

where λ is known as the *Lagrange multiplier*. Note that, the objective in (5.2) is non-convex and therefore is not tractable. The dual function of $\mathfrak{L}(\pi_\theta, \lambda)$ involves a point-wise maximum

with respect to π_θ and is written as [77]:

$$d(\lambda) = \max_{\pi_\theta \in \Pi} \mathfrak{L}(\pi_\theta, \lambda)$$

The dual function $d(\lambda)$ provides an upper bound on (5.2) and therefore needs to be minimized to contract the gap from optimality.

$$\mathfrak{D}^\star = \min_{\lambda \in \mathbb{R}_+} d(\lambda) \tag{5.3}$$

The dual problem in (5.3) is convex and tractable, but the question remains about how large the duality gap is. Or in other words, how sub-optimal the solution $\mathfrak{D}^\star$ of the dual problem (5.3) is with respect to the solution of the original problem stated in (5.1). To answer that question, *Paternain et. al.* [77] show that strong duality holds in this case under some mild conditions and the duality gap is arbitrarily small even with the parameterization (π_θ) of policies. We therefore tempt to optimize the dual version of this problem using gradients.

We rewrite the objective (5.2) and perform some algebraic manipulation as below:

$$\begin{aligned}
\mathfrak{L}(\pi_\theta, \lambda) &= \hat{\rho}(\pi_\theta, \mathcal{P}, r) - \lambda\Big(\beta - \hat{\rho}(\pi_\theta, \mathcal{P}, d)\Big) \\
&\overset{(a)}{=} \min_{p \in \mathcal{P}} \mathbb{E}_{\xi_1 \sim p}\left[g(\xi_1, r)\right] - \lambda\Big(\beta - \min_{q \in \mathcal{P}} \mathbb{E}_{\xi_2 \sim q}\left[g(\xi_2, d)\right]\Big) \\
&\overset{(b)}{=} \mathbb{E}_{\xi_1 \sim \tilde{p}}\left[g(\xi_1, r)\right] + \lambda\, \mathbb{E}_{\xi_2 \sim \tilde{q}}\left[g(\xi_2, d)\right] - \lambda\beta \\
&= \sum_{\xi_1 \in \Xi_{\tilde{p}}} p^{\pi_\theta}(\xi_1) g(\xi_1, r) + \lambda \sum_{\xi_2 \in \Xi_{\tilde{q}}} p^{\pi_\theta}(\xi_2) g(\xi_2, d) - \lambda\beta
\end{aligned}$$

Where $\Xi_{\tilde{p}}$ is the set of all possible trajectories induced by policy π_θ under transition function $\tilde{p}$. Similarly, $\Xi_{\tilde{q}}$ is the set of all possible trajectories induced by policy π_θ under transition function $\tilde{q}$. Step (a) above follows by assuming that the initial state distribution p_0 concentrates all of its mass to one single state s_0. And (b) follows with $\tilde{p} = \arg\min_{p \in \mathcal{P}} \mathbb{E}_{\xi_1 \sim p}\left[g(\xi_1, r)\right]$ and $\tilde{q} = \arg\min_{q \in \mathcal{P}} \mathbb{E}_{\xi_2 \sim q}\left[g(\xi_2, d)\right]$. Note that, $\tilde{p}$ and $\tilde{q}$ are distinct, independent and depend on rewards r and constraint costs d respectively. However,

the rewards and constraint costs are coupled together in reality, meaning that the set of two trajectories $\Xi_{\tilde{p}}$ and $\Xi_{\tilde{q}}$ would not be different. So we select one set of trajectories Ξ being either $\Xi_{\tilde{p}}$ or $\Xi_{\tilde{q}}$. This selection of Ξ may happen based on our priorities toward robustness of reward r (with corresponding trajectory $\Xi_{\tilde{p}}$) or constraint cost d (with corresponding trajectory $\Xi_{\tilde{q}}$). Or, it can also be the best (e.g. yielding higher objective value) set among $\Xi_{\tilde{p}}$ and $\Xi_{\tilde{q}}$ satisfying the constraint. We then have a simplified formulation for $\mathfrak{L}$ as below:

$$\mathfrak{L}(\pi_\theta, \lambda) = \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) - \lambda\beta \tag{5.4}$$

The goal is then to find a saddle point $(\pi_\theta{}^*, \lambda^*)$ of $\mathfrak{L}$ that satisfies $\mathfrak{L}(\pi_\theta, \lambda^*) \leq \mathfrak{L}(\pi_\theta{}^*, \lambda^*) \leq \mathfrak{L}(\pi_\theta{}^*, \lambda)$, $\forall \theta \in \mathbb{R}^k$ and $\forall \lambda \in \mathbb{R}_+$. This is achieved by ascending in θ and descending in λ using the gradients of objective $\mathfrak{L}$ with respect to θ and λ respectively [78].

Theorem 5.2.1. *The gradient of $\mathfrak{L}$ with respect to θ and λ can be computed as:*

$$\nabla_\theta \mathfrak{L}(\pi_\theta, \lambda) = \sum_{\xi} \hat{p}^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \sum_{t=0}^{T-1} \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}$$
$$\nabla_\lambda \mathfrak{L}(\pi_\theta, \lambda) = \sum_{\xi} \hat{p}^{\pi_\theta}(\xi) g(\xi, d) - \beta$$

Proof. See A.3.1 for the detailed derivation. □

With a fixed Lagrange multiplier λ, the constraint budget β in (5.4) offsets the sum by a constant amount. We can therefore omit this constant and define the Bellman operator for RCMDPs. We then show that this operator is a contraction.

Proposition 5.2.2. *The Bellman equation for RCMDPs can be defined as:*

$$\hat{w}^{\pi_\theta}(s) = \min_{p \in \mathcal{P}_{s,\pi(s)}} \mathbb{E}_{s' \sim p} \Big[r'_{s,\pi(s),s'} + \gamma \hat{w}^{\pi_\theta}(s') \Big] \tag{5.5}$$

Where $r'_{s,\pi(s),s'} = r_{s,\pi(s),s'} + \lambda d_{s,\pi(s),s'}$.

Proof.

$$
\begin{aligned}
\hat{w}^{\pi_\theta}(s) &= \min_{p\in\mathcal{P}_T} \mathbb{E}_{\xi\sim p}\left[g(\xi,r)+\lambda g(\xi,d)\right] \\
&\overset{(a)}{=} \min_{p\in\mathcal{P}_T} \mathbb{E}_{\xi\sim p}\Big[r_{s,\pi_\theta(s),s'}+\gamma r_{s',\pi_\theta(s'),s''}+\gamma^2 r_{s'',\pi_\theta(s''),s'''}\dots \\
&\qquad\qquad +\lambda\big(d_{s,\pi_\theta(s),s'}+\gamma d_{s',\pi_\theta(s'),s''}+\gamma^2 d_{s'',\pi_\theta(s''),s'''}+\dots\big)|\xi\Big] \\
&= \min_{p\in\mathcal{P}_T} \mathbb{E}_{\xi\sim p}\Big[\big(r_{s,\pi_\theta(s),s'}+\lambda d_{s,\pi_\theta(s),s'}\big)+\gamma\big(r_{s',\pi_\theta(s'),s''}+\lambda d_{s',\pi_\theta(s'),s''}\big) \\
&\qquad\qquad +\gamma^2\big(r_{s'',\pi_\theta(s''),s'''}+\lambda d_{s'',\pi_\theta(s''),s'''}\big)+\dots|\xi\Big] \\
&= \min_{p\in\mathcal{P}_T} \mathbb{E}_{\xi\sim p}\Big[r'_{s,\pi_\theta(s),s'}+\gamma r'_{s',\pi_\theta(s'),s''}+\gamma^2 r'_{s'',\pi_\theta(s''),s'''}+\dots|\xi\Big] \\
&\overset{(b)}{=} \min_{p\in\mathcal{P}_{s,\pi_\theta(s)}} \mathbb{E}_{s'\sim p}\Big[r'_{s,\pi_\theta(s),s'}+\gamma \min_{p\in\mathcal{P}_{T-1}} \mathbb{E}_{\xi'\sim p}\left[r'_{s',\pi_\theta(s'),s''}+\gamma r'_{s'',\pi_\theta(s''),s'''}+\dots|\xi'\right]\Big] \\
&= \min_{p\in\mathcal{P}_{s,\pi_\theta(s)}} \mathbb{E}_{s'\sim p}\Big[r'_{s,\pi_\theta(s),s'}+\gamma\hat{w}^{\pi_\theta}(s')\Big]
\end{aligned}
$$

□

Here (a) follows by expanding total return given a trajectory ξ and (b) follows by evaluating the one-step immediate transition apart. We define the Bellman optimality equation for RCMDPs as:

$$(\widehat{T}^{rc}_{\mathcal{P}} w)(s) := \max_{a\in\mathcal{A}} \min_{p\in\mathcal{P}_{s,a}} (r'_{s,a}+\gamma\cdot p^{\mathsf{T}} w) \tag{5.6}$$

Proposition 5.2.3. *The Bellman operator $\widehat{T}^{rc}_{\mathcal{P}}$ defined in (5.5) for RCMDPs is a contraction.*

Proof. The proof follows directly from Theorem 3.2 of [36]. □

The RCMDP Bellman operator $\widehat{T}^{rc}_{\mathcal{P}}$ therefore satisfies the Bellman optimality equation and converges to a fixed point.

5.2.1 Policy Gradient Algorithm

Algorithm 4 presents a robust constrained policy gradient algorithm based on the gradient update rules derived above in Theorem 5.2.1. The algorithm proceeds in an episodic way

based on trajectories and updates parameters based on the Monte-Carlo estimates. The algorithm requires an ambiguity set $\mathcal{P}$ as its input, which can be constructed with empirical estimates for smaller problems, as shown in chapter 3 and chapter 4. Or it can also be a parameterized estimate for larger problems [79].

Algorithm 4: Robust-Constrained Policy Gradient (RC-PG) Algorithm

Input: A differentiable policy parameterization π^θ, ambiguity set $\mathcal{P}$, confidence level α, step size schedules ζ_2 and ζ_1.

Output: Policy parameters θ

Initialize policy parameter: $\theta \leftarrow \theta_0$

for $k \leftarrow 0, 1, 2, \ldots$ **do**

  Sample initial state: $s_0 \sim p_0$

  Trajectory: $\xi \leftarrow \emptyset$

  `/* Simulate trajectory */`

  for $t \leftarrow 0, 1, 2, \ldots, T$ **do**

    Sample action: $a_t \sim \pi_\theta(\cdot|s_t)$

    Worst-case transitions with confidence α: $\hat{p}^{\pi_\theta} \leftarrow \arg\min_{p \in \mathcal{P}_{s,a}} p^T \hat{v}^{\pi_\theta}$

    Sample next state: $s_{t+1} \sim \hat{p}^{\pi_\theta}$;

    Observe reward $r_{s_t,a_t,s_{t+1}}$ and constraint cost $d_{s_t,a_t,s_{t+1}}$

    Append to trajectory: $\xi \leftarrow \left\{ s_t, a_t, s_{t+1}, r_{s_t,a_t,s_{t+1}}, d_{s_t,a_t,s_{t+1}}, \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \right\}$

  `/* Loop backward and update parameters with` ξ `*/`

  θ update: $\theta \leftarrow \theta + \zeta_2(k) \nabla_\theta \mathfrak{L}(\pi_\theta, \lambda)$

  λ update: $\lambda \leftarrow \lambda - \zeta_1(k) \nabla_\lambda \mathfrak{L}(\pi_\theta, \lambda)$

return θ;

The step size schedules used in Algorithm 4 satisfy the standard conditions for stochastic approximation algorithms [80]. That is, θ update is on the fastest time-scale $\zeta_2(k)$ and the λ update is on a slower time-scale $\zeta_1(k)$. This results in a two time-scale stochastic approximation algorithm, we derive its convergence to a saddle point as below.

Theorem 5.2.4. *Under assumptions **(A1)** - **(A7)** as stated in Appendix A.3.2, the sequence of parameter updates of Algorithm 4 converges almost surely to a locally optimal policy $\pi_\theta^\star$ as the number of trajectories $k \to \infty$.*

Proof. We report the proof in Appendix A.3.4. □

5.2.2 Actor Critic Algorithm

Algorithm 5: Robust Constrained Actor Critic (RC-AC) Algorithm

Input: A differentiable policy parameterization π_θ, a differentiable state-value function $w^{\pi_\theta}(s, f)$, confidence level α, step size schedule ζ_1 and ζ_2.

Output: Policy parameters θ

Initialize policy parameter $\theta \in \mathbb{R}^k$ and state-value weights $f \in \mathbb{R}^{k'}$;
for $j \leftarrow 0, 1, 2, \ldots$ **do**
  Sample initial state: $s_0 \sim p_0$;
  $t \leftarrow 0$;
  `/* Loop for each step along a trajectory */`
  while s_t *not terminal* **do**
    Sample action: $a_t \sim \pi_\theta(\cdot|s_t)$
    Worst-case transitions with confidence α: $\hat{p}^{\pi_\theta} \leftarrow \arg\min_{p \in \mathcal{P}_{s,a}} p^T w^{\pi_\theta}$
    Sample next state $s_{t+1} \sim \hat{p}^{\pi_\theta}$ and observe $r_{s_t,a_t,s_{t+1}}$ and $d_{s_t,a_t,s_{t+1}}$;
    TD error: $\delta_t \leftarrow r'_{s_t,a_t,s_{t+1}} + \gamma w^{\pi_\theta}(s_{t+1}, f) - w^{\pi_\theta}(s_t, f)$;
    `/* Update parameters with gradient estimates */`
    θ **update:** $\theta \leftarrow \theta + \zeta_2(k)\delta_t \nabla_\theta \mathfrak{L}(\pi_\theta, \lambda)$;
    f **update:** $f \leftarrow f + \zeta_1(k)\delta_t \nabla_f w^{\pi_\theta}(s_t, f)$;
    $t \leftarrow t + 1$;
return θ ;

The general issue of having high variance in the Monte Carlo based policy gradient algorithm can be handled by introducing state values to use as baselines [2]. As the optimal value function for RCMDPs can be computed using Bellman style recursive updates as shown in (5.5), an extension of the above PG algorithm to the actor-critic framework is straightforward. Algorithm (5) presents an actor critic (AC) algorithm for RCMDPs. The state-value parameterization with f brings a new dimension in algorithm (5) and results in a three time-scale stochastic algorithm. The convergence properties for this AC algorithm can be derived in a way similar to Theorem 5.2.4. We therefore omit the detailed derivations. Robustness introduced in PG and AC algorithms can be handled in polynomial time. Like general AC and PG algorithms, the time complexity for each iteration of algorithm 4 and 5 therefore remains to be $O(|\theta|)$, where $|\theta|$ is the number of policy parameters.

5.3 Empirical Study

In this section, we empirically evaluate the performance of our robust-constrained policy gradient algorithm on an inventory management [3,4,81] problem. We also report results for a robust-constrained version of actor-critic (AC) algorithm in cart-pole [2,82] domain. Note that, the prefix R will denote Robust and the prefix C will denote constrained versions of the algorithms.

5.3.1 Inventory Management Problem

The state space of the inventory management problem is discrete and is represented by the level of inventory. The purchase cost of each product is 2.49, sale price is 3.99 and holding cost is 0.03. The demand for a product is random and comes from a normal distribution with *unknown* parameters. The reward is represented by the *profit = revenue - costs*. The goal is to order products from a supplier in order to meet customer demands. This standard inventory setting further incorporates a constraint associated to stock-out event, which triggers when the demand exceeds the current stock of an item. A stock-out event usually results in lost revenues and customer dissatisfaction, therefore incorporating an additional cost for a company.

This experiment on inventory management problem is run with a confidence level $\delta = 0.9$, which translates to a lower bound on the return estimates with 90% confidence level as discussed in previous chapters. We use a discount factor $\gamma = 0.9$, and $n_{s,a} = 100$ number of samples drawn for each state-action from the underlying true transition distribution $p^{\star}_{s,a}$. We compare our robust-constrained method *RC-PG* as described in algorithm 4 with general policy gradient algorithm [2]. We also evaluate a variant of PG method that is robust, but does not involve any constraint.

We analyze the robustness of policies in a perturbed version of the inventory problem, where the perturbation is introduced by varying the standard deviation of the demand

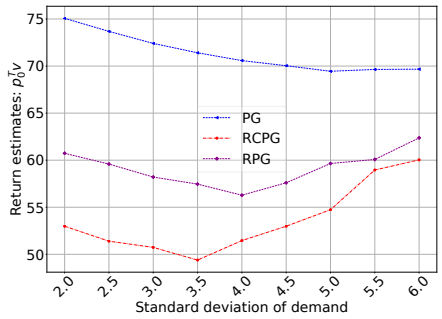

Figure 5.1: Estimated returns as the demand distribution varies.

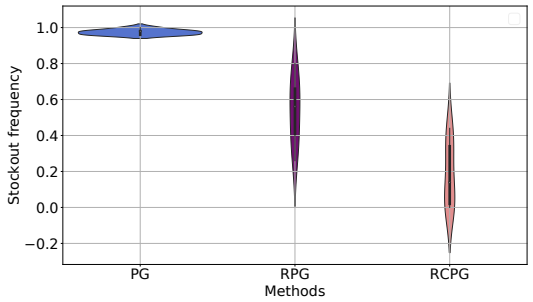

Figure 5.2: Stock-out frequency for different methods.

distribution. Figure 5.1 shows the estimated returns for different policies in the *Y-axis* and the standard deviation of demand on the *X-axis*. Policy computed by the PG method offers the highest return estimates throughout the whole range of perturbed environments. But this policy neither provides any worst-case guarantee for performance, nor does it care about constraint satisfaction. The violin plot in figure 5.2 confirms that behavior. On the other hand, the policy computed with RC-PG method has the lowest return estimates. But this policy provides a worst-case performance guarantee along with best constraint satisfaction as displayed in figure 5.2. The robust PG method does not explicitly consider the constraint. It therefore trade-offs some constraint satisfaction performance with higher return estimates as shown in figure 5.1.

5.3.2 Cart-pole

We next evaluate our algorithm on cart-pole, a standard RL benchmark problem [2,82]. The task here is to balance a pole atop a cart by pushing the cart left or right. We implement an actor-critic (AC) algorithm [2, 83] using a simple neural network of 1 fully connected hidden layer with 128 weights and *ReLu* activations. We explicitly introduce a noise in the environment by altering the mass of the pole from a finite set of preset values. We train the agent on this perturbed version of the environment and then evaluate the policy on an environment perturbed with a different set of values for the pole mass. A comparison is

Methods	Return Mean	Return V@R (90%)	Constraint Violation
AC	175.45 ± 2.99	153.3	2.3%
C-AC	171.88 ± 6.96	147.6	0.0%
R-AC	118.22 ± 6.07	84.5	1.1%
RC-AC	123.26 ± 8.64	89.5	0.05%

Table 5.1: Evaluating learned policy in test environments for cart-pole problem.

provided in Table 6.3. The non-robust versions of the actor-critic method (AC and C-AC) have higher expected return along with a higher performing tail performance computed as 90% value-at-risk. But they do not provide robustness guarantees and can perform poorly in the worst-case situations. The robust methods (R-AC and RC-AC) provide a lower estimate for the expected return and tail performance, but is expected to provide consistent performance throughout a range of different parameter values of the environment. Also, it can be seen in the table that the constrained methods are able to reduce the constraint violation rates to almost zero.

5.4 Contributions

This is a joint work with Mitsubishi Electric Research Lab (MERL), thanks to my collaborators Mouhacine Benosman and Jeroen van Baar. Mouhacine envisioned the utility of uniting the ideas of constrained MDPs and robust MDPs, leading to Robust Constrained MDPs (RCMDPs). I derived the RCMDP framework and developed theoretical foundations. I proposed policy gradient class of algorithms for optimizing the RCMDP objective and empirically validated its usefulness. All of these have been done under close supervision of Mouhacine and Jeroen. An earlier version of this work was presented at NeurIPS 2020 workshop on The Challenges of Real World Reinforcement Learning.

CHAPTER 6

RISK-AVERSE SOFT-ROBUST REINFORCEMENT LEARNING

The most common goal when solving Markov Decision Processes (MDP) is to maximize the expected sum of discounted rewards [2]. As discussed in previous chapters, good policies will achieve good rewards only in expectations, and may fail catastrophically due to stochastic transitions and uncertain models. When the stakes are high, it is, therefore, better to compute risk-averse policies that give up some of the expected rewards in return of minimizing the probability of a catastrophic failure [67, 76].

In this chapter, we propose a new method to compute policies that mitigates the risk of failure that could arise from either stochastic transition probabilities or uncertain models. In effect, this chapter combines risk-averse and robust reinforcement learning, two streams of work that address similar concerns but have been treated mostly independently thus far. We argue that the combined Risk-Averse Soft-Robust (RASR) objective is more appropriate in domains that involve high stakes and uncertain models. Surprisingly, solving the combined RASR objective can be easier than solving objectives that target robust RL and risk-averse RL objectives individually.

Many framework for measuring risk have been studied. Risk measures have gained popularity in machine learning in recent years. This is perhaps because they combine attractive computational properties with good interpretability and realistic assumptions. Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) [74, 78, 84, 85] are popular risk measures used in RL because of their simplicity and interpretability. Their use in sequential optimization is complicated because they are not dynamically-consistent, which means that the optimal

policy may need to be history-dependent and the Bellman optimality equations cannot be readily used to compute the optimal value function [29, 86]. One can define an iterated version of VaR and CVaR, but these are difficult to interpret, can be overly conservative, and violate an important property of law invariance.

Risk-averse reinforcement learning optimizes a risk-sensitive objective that penalizes the variability in returns caused by stochastic transitions. This uncertainty is referred to also as *aleatory uncertainty*. For example, policy gradient and actor-critic algorithms to optimize risk-averse objective for MDPs have been developed recently [78, 87, 88] for several common risk-measures. These methods do not consider model uncertainty which leads to methods that differ from our work in several crucial aspects.

Robust reinforcement learning targets problems in which the model of the domain is not known precisely [30, 36]. The agent is instead uncertain about several models that might best represent the reality. This uncertainty, which is subjective to the agent, is usually known as *epistemic* uncertainty. Limited data, inaccurate measurement of model parameters, overlooked factors etc. are some common reasons for epistemic uncertainties. Robust optimization is a popular approach to handle this uncertainty. Instead of estimating a single model of the environment, such as transition probabilities, robust optimization techniques estimate a range of plausible models. They compute the best policy for the worst-case plausible model from the estimated range. This approach is simple and can be computationally effective [28]. Unfortunately, robust policies are reliable but too conservative [76].

Soft-robust optimization methods connect risk aversion with robust optimization to achieve robustness while computing policies that are less conservative [9, 10, 75, 89, 90]. The methods also estimate the range of possible models, or transition probabilities, that are consistent with the observed data. But then optimize a policy with respect to a risk metric of its performance across different models. In one early example of this approach, the *percentile criterion* optimizes the value-at-risk (VaR) of the policy's performance with respect to uncertain model [91]. This allows to trade off the performance between the average and

worst-case models more effectively.

Table 6.1: Comparison of previous risk-sensitive methods.

References	Uncertainty Types		Risk Measures		
	Aleatory	Epistemic	Variance	CVaR	Entropic
RASR (this work)	✓	✓	✗	✗	✓
Lobo et al. [89]	✗	✓	✗	✓	✗
Nass et al. [92]	✓	✗	✗	✗	✓
Fei et al. [93]	✓	✗	✗	✗	✓
Eriksson and Dimitrakakis [94]	✗	✓	✗	✓	✓
Hiraoka et al. [90]	✗	✓	✗	✓	✗
Prashanth and Ghavamzadeh [87]	✓	✗	✓	✗	✗
Chow and Ghavamzadeh [78]	✓	✗	✗	✓	✗
Tamar et al. [88]	✓	✗	✗	✓	✗
Tamar et al. [95]	✓	✗	✓	✗	✗

Table 6 provides a comparative overview of prior works related to applying risk measures in RL. Some methods proposed previously only handle aleatory uncertainty. All these methods assume that the model is precisely known and risk-measures are only required to deal with the inherent stochasticity. On the other hand, another set of methods only handle epistemic uncertainty. These methods only care about model uncertainty and overlook the fact that simultaneous treatment of inherent transition uncertainty is important. RASR framework proposed in this paper is the only method that simultaneously handles both of these uncertainties and provides rigorous theoretical analysis with empirical evidences.

As the main contribution of this chapter, we study the basic computational properties of Markov decision processes that are both robust and risk averse. We show that when the same entropic risk measure is used, then the finite horizon problem can be solved optimally, while the infinite horizon discounted objective can be approximated closely. This is in stark contrast with prior work on risk-averse and robust reinforcement learning, which typically involves solving NP hard problems. Our contributions are five-fold: i) propose a unified risk-averse soft-robust (RASR) framework to deal with both epistemic and aleatory uncertainties, ii) derive Bellman equation for RASR framework and propose a value iteration algorithm,

iii) formulate gradient update rule to optimize RASR objective and propose an actor-critic algorithm for larger problems, iv) derive a finite sample convergence analysis for entropic risk measure, and v) empirically validate the utility of our RASR framework on a set of problem domains.

The remainder of the chapter is organized as follows: Section 6.1 formally describes the problem setting and establishes several useful properties for entropic risk measure. The RASR framework is presented in Section 6.2 along with corresponding theoretical analysis and algorithms. Section 6.3 presents the empirical evaluation on several problem domains. Section 6.4 finally draws the concluding remarks.

6.1 Problem Formulation

We use the standard Markov Decision Process (MDP) model with a finite number of states $\mathcal{S} = \{1, \ldots, S\}$ and finite number of actions $\mathcal{A} = \{1, \ldots, A\}$. Every action $a \in \mathcal{A}$ is available for the decision maker to take in every state $s \in \mathcal{S}$. After taking an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the decision maker transitions to a next state s' and receives a reward $|r_{s,a,s'}| \leq r_{\max} \in$. A transition probability function $P : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ describes this state transition given the current $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. An initial state distribution is $p_0 \in \Delta^{\mathcal{S}}$ and $\gamma \in [0, 1]$ is a discount factor.

A solution to an MDP is a policy $\pi : \mathcal{S} \to \Delta^{\mathcal{A}}$, which defines an action $a \in \mathcal{A}$ given a state $s \in \mathcal{S}$. The set of all randomized policies is defined as $\Pi = (\Delta^{\mathcal{A}})^{\mathcal{S}}$ and $\Pi_D = \mathcal{A}^{\mathcal{S}}$ denotes the set of all deterministic policies. Our objective is to maximize the infinite horizon $\gamma-$discounted return [2, 3]:

$$v_P^{\pi} = \mathbb{E}_P \left[\sum_{t=0}^{\infty} \gamma^t r_{s_t, \pi(s_t), s_{t+1}} \right]$$

where $s_0 \sim p_0$ and $s_{t+1} \sim P(s_t, a_t)$.

We operate in a *batch* RL setting [18] where a logged set of data $\mathcal{D} = \left\{ s_i, a_i, s_i' \right\}_{i=1}^{M}$ is provided. We do not have any assumption about the policy that is used to generate the

dataset $\mathcal{D}$, it can be any arbitrary baseline policy π_B. The only assumption is that, the next state $s'_i \in \mathcal{S}$ given a current state $s_i \in \mathcal{S}$ and action $a_i \in \mathcal{A}$ is distributed according to the *true* transition probability $p^\star_{s_i,a_i}$. Note that, the true transition model $P^\star$ is a random variable in the Bayesian setting. Given the data $\mathcal{D}$, we derive a posterior distribution $\hat{P}$ over the true transition model $P^\star$ conditional on $\mathcal{D}$: $\hat{P} = P^\star|\mathcal{D}$. As discussed in Chapter 4, the posterior can be derived analytically by using conjugate priors like Dirichlet distribution, or can also be obtained with MCMC sampling libraries like JAGS, or Stan [63]. We denote by $\hat{P}^\omega$ a sample from the posterior distribution $\hat{P}$ with weight f^ω, where f is the probability measure function of $\hat{P} : \Omega \to \left(\Delta^{\mathcal{S}}\right)^{\mathcal{S}\times\mathcal{A}}$. We consider a *dynamic* model of uncertainty where the uncertain parameters can vary at every time step. This setting is common [51, 75, 96], but can be more pessimistic because exploitation of current state feature information may not be possible [89].

A trajectory τ of state transitions with policy π and starting state $s_0 \sim p_0$ can be defined in the dynamic setting as:

$$\begin{aligned}\tau =&\Big(s_t\Big)_{t=0}, \Big(P_t^\omega \sim \hat{P}, s_t \sim P_t^\omega(\cdot|s_{t-1}, \pi(s_{t-1}))\Big)_{t=1}, \dots, \\ &\dots, \Big(P_t^\omega \sim \hat{P}, s_t \sim P_t^\omega(\cdot|s_{t-1}, \pi(s_{t-1}))\Big)_{t=T-1}, \Big(P_t^\omega \sim \hat{P}, s_t \sim P_t^\omega(\cdot|s_{t-1}, \pi(s_{t-1}))\Big)_{t=T}\end{aligned}$$

The probability of sampling such a trajectory τ is: $p_\theta(\tau) = p_0(s_0) \prod_{t=1}^{T} \pi_\theta(a_t|s_t) \hat{P}_t^\omega(s_{t+1}|s_t, a_t) f_t^\omega$. The total $\gamma-$ discounted return for a trajectory τ is $R(\tau) = \sum_{t=0}^{T} \gamma^t r_{s_t,\pi(s_t),s_{t+1}}$. The expected $\gamma-$discounted return is:

$$v^{\mathbb{E}}_{\pi,\hat{P}}(s_0) = \mathbb{E}_{\tau\sim p_\theta(\tau)}\Big[R(\tau)\Big]$$

where $s_0 \sim p_0$. This quantity represents the value function in dynamic setting, which satisfies Bellman optimality equation for each $s \in \mathcal{S}$ and converges to a fixed point [51].

6.1.1 Entropic Risk Measure

Entropic risk measure $\rho^{\alpha} : X \to \mathbb{R}$ for a random variable X is a popular risk measure based on exponential utility function and for a risk-aversion parameter $\alpha > 0$, it takes the form [97, 98]:

$$\rho_X^{\alpha}(X) = -\frac{1}{\alpha} \log \big(\mathbb{E}_X[\exp(-\alpha X)] \big) \tag{6.1}$$

The entropic risk measure satisfies the properties of monotonicity, translation invariance and convexity [99]:

Definition 6.1.1. For all $X, Y : \Omega \to \mathbb{R}$ and a scalar $m \in \mathbb{R}$

- Monotonicity: If $X \leq Y$, then $\rho_X^{\alpha}(X) \geq \rho_Y^{\alpha}(Y)$.
- Translation invariance: $\rho_X^{\alpha}(X + m) = \rho_X^{\alpha}(X) - m$.
- Convexity: $\rho_{X,Y}^{\alpha}\big(\lambda X + (1-\lambda)Y\big) \leq \lambda \rho_X^{\alpha}(X) + (1-\lambda)\rho_Y^{\alpha}(Y)$, for $0 \leq \lambda \leq 1$.

But ρ^{α} is not a coherent risk measure because it does not satisfy the positive homogeneity property [97]. The dual representation of the entropic risk measure takes the following form [99]:

$$\rho_X^{\alpha}(X) = \sup_{Q \in \Delta} \left\{ \mathbb{E}_Q[-X] - \frac{1}{\alpha} D_{\mathrm{KL}}\left(Q || P \right) \right\}$$

Where Δ denotes the class of all probability measures on X and $D_{\mathrm{KL}}\left(Q || P \right)$ is the relative entropy of $Q \ll P$. This dual representation is convex, monotone and translation invariant and simply follows from the conjugate and bi-conjugate representation of (6.1) [97].

With an abuse of notation, we denote the joint entropic risk measure of two *independent* random variables X and Y as:

$$\rho_{X,Y}^{\alpha}(X + Y) = -\frac{1}{\alpha} \log \Big(\mathbb{E}_{X,Y} \left[\exp\left(-\alpha(X+Y)\right) \right] \Big)$$

We now derive some useful properties of the entropic risk measure that we will need later in this chapter.

Lemma 6.1.1. *(Comonotonic Additive) For two independent random variables X and Y, $\rho_X^\alpha(X) + \rho_Y^\alpha(Y) = \rho_{X,Y}^\alpha(X+Y)$.*

Proof.

$$\begin{aligned}
\rho_X^\alpha(X) + \rho_Y^\alpha(Y) &= -\frac{1}{\alpha}\log\Big(\mathbb{E}_X\left[\exp\left(-\alpha X\right)\right]\Big) - \frac{1}{\alpha}\log\Big(\mathbb{E}_Y\left[\exp\left(-\alpha Y\right)\right]\Big) \\
&= -\frac{1}{\alpha}\log\Big(\mathbb{E}_X\left[\exp\left(-\alpha X\right)\right]\cdot\mathbb{E}_Y\left[\exp\left(-\alpha Y\right)\right]\Big) \\
&= -\frac{1}{\alpha}\log\Big(\mathbb{E}_{X,Y}\left[\exp\left(-\alpha X\right)\cdot\exp\left(-\alpha Y\right)\right]\Big) \\
&= -\frac{1}{\alpha}\log\Big(\mathbb{E}_{X,Y}\left[\exp\left(-\alpha(X+Y)\right)\right]\Big) \\
&= \rho_{X,Y}^\alpha(X+Y)
\end{aligned}$$

□

Lemma 6.1.2. *(Recursive) For two independent random variables X and Y, $\rho_X^\alpha\Big(X + \rho_Y^\alpha(Y)\Big) = \rho_{X,Y}^\alpha(X+Y)$.*

Proof.

$$\begin{aligned}
\rho_X^\alpha\Big(X+\rho_Y^\alpha(Y)\Big) &= -\frac{1}{\alpha}\log\left(\mathbb{E}_X\left[\exp\Big(-\alpha\Big(X+\rho_Y^\alpha(Y)\Big)\Big)\right]\right) \\
&\overset{(a)}{=} -\frac{1}{\alpha}\log\left(\mathbb{E}_X\left[\exp\Big(-\alpha\rho_Y^\alpha(X+Y)\Big)\right]\right) \\
&= -\frac{1}{\alpha}\log\left(\mathbb{E}_X\left[\exp\Big(-\alpha\frac{-1}{\alpha}\log\mathbb{E}_Y\left[\exp\big(-\alpha(X+Y)\big)\right]\Big)\right]\right) \\
&= -\frac{1}{\alpha}\log\left(\mathbb{E}_{X,Y}\left[\exp\big(-\alpha(X+Y)\big)\right]\right) \\
&= \rho_{X,Y}^\alpha(X+Y)
\end{aligned}$$

Here (a) follows because entropic risk measure ρ^α is *cash-invariant* [100]. □

Lemma 6.1.3. *(Translation Invariant) For two independent random variables X, Y, $\rho_X^\alpha\Big(X+\rho_Y^\alpha(Y)\Big) = \rho_X^\alpha(X) + \rho_Y^\alpha(Y)$.*

Proof. From Lemma 6.1.1 and Lemma 6.1.2, we have:

$$\rho_X^\alpha\Big(X + \rho_Y^\alpha(Y)\Big) = \rho_{X,Y}^\alpha(X+Y) = \rho_X^\alpha(X) + \rho_Y^\alpha(Y)$$

□

Next, we analyze the finite-sample convergence properties of entropic risk measure for a random variable $X \subseteq [0, U]$. One important concept in this regard is *Optimized Certainty Equivalent (OCE)*, which is defined as follows:

Definition 6.1.2. (OCE) Let ϕ: $\mathbb{R} \to \mathbb{R} \cup +\infty$ be a closed, concave function with dom $\phi \subseteq \mathbb{R}_+$ and have a minimum value of 0 attained at 1. Then the OCE of a random variable $X \in \mathcal{X}$ can be defined following Definition 2.2 of [101] as:

$$S_\phi(X) = \sup_{\eta \in \mathbb{R}} \Big\{ \eta + \mathbb{E}\left[\phi(X - \eta)\right] \Big\} \tag{6.2}$$

We can compute an estimate $\hat{S}_\phi(X)$ of (6.2) from i.i.d samples $X_1, \ldots, X_N$ as:

$$\hat{S}_\phi(X_1, \ldots, X_N) = \sup_{\eta \in \mathbb{R}} \Big\{ \eta + \frac{1}{N} \sum_{i=1}^{N} \phi(X_i - \eta) \Big\}$$

Theorem 6.1.4. *An OCE estimate $\hat{S}_\phi$ for random variable X can be upper bounded by an amount ε as:*

$$P(|\hat{S}_\phi(X_1, \ldots, X_N) - S_\phi(X)| \geq \varepsilon) \leq 2 \exp\left(-2(\varepsilon/\phi(U))^2 \cdot N\right)$$

Proof. The proof follows directly from Theorem 3.2 of [102]. □

Theorem 6.1.4 leads us to derive the finite-sample deviation bound for entropic risk measure.

Corollary 6.1.5. *When $S_\phi(X) = \rho^\alpha_{ent}(X)$ for some $\alpha \in (0, 1]$, then we have:*

$$P(|\hat{\rho}^\alpha(X_1, \ldots, X_N) - \rho^\alpha(X)| \geq \varepsilon) \leq 2 \exp\left(-2\alpha^2\varepsilon^2 N\right)$$

Proof. The proof follows by substituting the utility function $\phi(t) = \frac{1}{\alpha}\big(1 - \exp(-\alpha t)\big)$ of the entropic risk measure into the deviation bound of OCE derived in Theorem 6.1.4. See Appendix A.4.1 for the full derivation. □

6.2 Risk-Averse Soft-Robust (RASR) Framework

We formally define the RASR framework in this section. At each time step t, both the model parameters P^ω_t and the transition to a next state s_{t+1} are uncertain under dynamic model of uncertainty. The RASR framework therefore simultaneously takes both of these uncertainties into consideration. We define the RASR value function $\hat{v}^\pi_{\hat{P}}$ for a policy π and posterior distribution $\hat{P}$ as the entropic risk measure over $\gamma-$discounted return under the dynamic model of uncertainty.

$$\hat{v}^\pi_{\hat{P}}(s) = \rho^\alpha_{\hat{P},S,A}\Big[\sum_{t=0}^{T} \gamma^t r_{s_t,\pi(s_t),s_{t+1}} | S_0 \sim p_0, S_{t+1} \sim \hat{P}_t(s_t, a_t), A_t \sim \pi(S_t), \hat{P}_t \sim f\Big] \tag{6.3}$$

Where $s \sim s_0$. For $\gamma = 1$, The RASR value function $\hat{v}^\pi_{\hat{P}}$ satisfies a Bellman style equation as stated below.

Theorem 6.2.1. *(Bellman Equation) For a fixed policy π and horizon length $T < \infty$, the RASR value function $\hat{v}^\pi_{\hat{P}}$ satisfies a Bellman equation for each $s \in \mathcal{S}$:*

$$\hat{v}^\pi_{\hat{P}}(s) = \rho^\alpha_{P^\omega \sim \hat{P}, s' \sim P^\omega(\cdot|s,\pi(s))}\Big[r_{s,a,s'} + \hat{v}^\pi_{\hat{P}}(s')\Big]$$

Proof.

$$
\begin{aligned}
\hat{v}^{\pi}_{\hat{P}}(s) &= \sum_{k=0}^{T} \rho^{\alpha}_{P^{\omega}_{t+k}\sim\hat{P},s_{t+k+1}\sim P^{\omega}_{t+k}(\cdot|s_{t+k},\pi(s_{t+k}))}\left[r_{s_{t+k},\pi(s_{t+k}),s_{t+k+1}}\right]\Big|s_t = s \\
&\overset{(a)}{=} \rho^{\alpha}_{P^{\omega}_{t}\sim\hat{P},s_{t+1}\sim P^{\omega}_{t}(\cdot|s_t,\pi(s_t))}\Big[r_{s_t,\pi(s_t),s_{t+1}} \\
&\qquad + \sum_{k=1}^{T} \rho^{\alpha}_{P^{\omega}_{t+k}\sim\hat{P},s_{t+k+1}\sim P^{\omega}_{t+k}(\cdot|s_{t+k},\pi(s_{t+k}))}\left[r_{s_{t+k},\pi(s_{t+k}),s_{t+k+1}}\right]\Big]\Big|s_t = s \\
&\overset{(b)}{=} \rho^{\alpha}_{P^{\omega}_{t}\sim\hat{P},s'\sim P^{\omega}_{t}(\cdot|s_t,\pi(s_t))}\Big[r_{s_t,\pi(s_t),s'} + \hat{v}^{\pi}_{\hat{P}}(s')\Big]\Big|s_{t+1} = s'
\end{aligned}
$$

Here (a) follows from Lemma 6.1.3 and (b) follows by replacing the definition of the value function for a next state s_{t+1}. □

We now approximate the infinite horizon discounted objective under this RASR framework. We use $\mathcal{V}$ to denote the set of all bounded real-valued functions on $\mathcal{S}$. Let $\|v\|_\infty$ denote the L_∞ norm on $\mathcal{V}$. Then $(\mathcal{V}, \|\cdot\|_\infty)$ is a Banach space. We define the RASR Bellman operator $\mathfrak{T} : \mathcal{V} \to \mathcal{V}$ for a state s and transition posterior $\hat{P}$ as the best action with respect to the entropic risk measure over model and state transition distributions.

$$
(\mathfrak{T}_{\hat{P}}\hat{v})(s) = \max_{a\in\mathcal{A}} \rho^{\alpha}_{P^{\omega}_{t}\sim\hat{P},s'\sim P^{\omega}_{t}(\cdot|s_t,a_t)}\Big[r_{s_t,a_t,s'} + \gamma\hat{v}(s')\Big], \forall s \in \mathcal{S}, \forall \hat{v} \in \mathcal{V}\ . \tag{6.4}
$$

We now show that, the RASR Bellman operator $\mathfrak{T}$ as defined in (6.4) is a contraction mapping and therefore converges to a fixed point.

Theorem 6.2.2. *(Contraction) For any two bounded functions $u_{\hat{P}} : \mathcal{S} \to \mathbb{R}$, and $v_{\hat{P}} : \mathcal{S} \to \mathbb{R}$ under a posterior transition $\hat{P}$, and $\gamma \in [0, 1)$, the RASR Bellman operator $\mathfrak{T}$ is a contraction mapping. In particular, it holds for all $u_{\hat{P}}, v_{\hat{P}} \in \mathcal{V}$ that:*

$$
\|\mathfrak{T}u_{\hat{P}} - \mathfrak{T}v_{\hat{P}}\|_\infty \leq \gamma\|u_{\hat{P}} - v_{\hat{P}}\|_\infty \tag{6.5}
$$

Proof. We report the proof in Appendix A.4.2. □

Algorithm 6: RASR Value Iteration (RASR-VI)

Input: States $\mathcal{S}$, Actions $\mathcal{A}$, Transition posterior $\hat{P}$, Rewards r, discount factor γ and admissible maximum error ϵ.

Output: RASR value function $\hat{v}$

Initialize $\hat{v}(s)$ arbitrarily for all $s \in \mathcal{S}$;
repeat
  $v' \leftarrow \hat{v}, \Delta \leftarrow 0$;
  for *each* $s \in \mathcal{S}$ **do**
    $\hat{v}(s) \leftarrow \max_{a \in \mathcal{A}} \rho^{\alpha}_{P^{\omega} \sim \hat{P}, s' \sim P^{\omega}(\cdot|s,a)} \Big[r_{s,a,s'} + \gamma \hat{v}(s') \Big]$;
    $\Delta \leftarrow \max \big(\Delta, |\hat{v}(s) - v'(s)| \big)$
until $\Delta < \epsilon$;
return $\hat{v}$;

Algorithm 6 shows a value iteration algorithm based on the RASR Bellman operator $\mathfrak{T}$. The contraction property of $\mathfrak{T}$ shown in Theorem 6.2.2 ensures that Algorithm 6 converges to a fixed point of the optimal RASR value function $\hat{v}^{\star}$. Similar to regular value iteration, Algorithm 6 is *P-complete* as it does not require any additional computational step.

6.2.1 RASR Policy Parameterization

The value iteration algorithm proposed in previous section is good for tabular setting with discrete state and action spaces. But many real-world problems have large and continuous state spaces, which do not fit into the tabular setting. We therefore in this section extend the RASR framework beyond the tabular context by considering a class of parameterized stationary randomized policy $\pi_{\theta} : \mathcal{S} \to \Delta^{\mathcal{A}}$, where $\theta \subseteq \mathbb{R}^k$ is a $k-$dimensional parameter vector. We rewrite the trajectory based RASR objective as below:

$$\begin{aligned} J(\pi_{\theta}) &= \rho^{\alpha}_{\tau \sim p_{\theta}(\tau)} \Big[R(\tau) \Big] \\ &= -\frac{1}{\alpha} \log \left(\mathbb{E}_{\tau \sim p_{\theta}(\tau)} \Big[\exp \big(-\alpha R(\tau) \big) \Big] \right) \end{aligned} \tag{6.6}$$

We derive the gradient update formula of the RASR objective $J(\pi_{\theta})$ defined in (6.6) with respect to the policy parameter θ.

Theorem 6.2.3. *(RASR Policy-Gradient) The gradient of RASR objective $J(\pi_\theta)$ with respect to the parameter θ is:*

$$\nabla_\theta J(\pi_\theta) = \frac{-\sum_\tau p_\theta(\tau) \sum_{t=0}^{T} \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \cdot \exp\Big(-\alpha \sum_{t=0}^{T} r_{s_t,a_t}\Big)}{\alpha \sum_\tau p_\theta(\tau) \exp\big(-\alpha R(\tau)\big)}$$

Proof. The proof is deferred to Appendix A.4.3. □

The gradient derived in Theorem (6.2.3) then can be used to update the policy in the direction of the estimated gradient $\nabla_\theta J(\pi_\theta)$. Algorithm (7) shows an actor-critic (AC) algorithm for updating the parameterized policy optimizing the RASR objective.

Algorithm 7: RASR Actor Critic (RASR-AC) Algorithm with Entropic Risk

Input: A differentiable policy parameterization π_θ, a differentiable state-value function $\hat{v}^{\pi_\theta}_{\hat{P}}(s, w)$, confidence level α, step size schedule ζ_1 and ζ_2.

Output: Parameterized policy π_θ

Initialize policy parameter $\theta \in \mathbb{R}^k$ and state-value weights $w \in \mathbb{R}^{k'}$;

for $k \leftarrow 0, 1, 2, \ldots$ **do**

  Sample initial state: $s_0 \sim p_0$, set time-step $t \leftarrow 0$;

  while s_t *is not terminal* **do**

    Sample action $a_t \sim \pi_\theta(\cdot|s_t)$, then take action a_t and observe next state s_{t+1};

    TD error: $\delta_t \leftarrow \rho^{\alpha}_{P^\omega_t \sim \hat{P}, s_{t+1} \sim P^\omega_t(\cdot|s_t,a_t)} \Big[r_{s_t,a_t,s_{t+1}} + \gamma \hat{v}^{\pi_\theta}_{\hat{P}}(s_{t+1}, w) \Big] - \hat{v}^{\pi_\theta}_{\hat{P}}(s_t, w)$;

    θ update: $\theta \leftarrow \theta + \zeta_2(k) \delta_t \nabla_\theta J(\pi_\theta)$;

    w update: $w \leftarrow w + \zeta_1(k) \delta_t \nabla_w \hat{v}^{\pi_\theta}_{\hat{P}}(s_t, w)$;

    $t \leftarrow t + 1$;

return π_θ ;

The step size schedule of Algorithm (7) satisfy the standard conditions for stochastic approximation algorithms ensuring that θ update is on the fastest time-scale $\zeta_2(k)$ and the w update is on a slower time-scale $\zeta_1(k)$. This results in a two time-scale stochastic approximation algorithm and the convergence of it to a saddle point can be shown following standard proof techniques presented in [80].

6.3 Empirical Evaluation

In this section, we empirically evaluate the RASR framework on a set of different problem domains. All the experiments are run with risk parameter $\alpha = 0.9$ unless otherwise specified. We start with logged data $\mathcal{D}$ collected by running arbitrary baseline policies π_b from the underlying true distribution $P^\star$. We then use $\mathcal{D}$ to compute the Bayesian posterior from the prior. One can use conjugate distributions (e.g. Dirichlet) or MCMC sampling libraries like JAGS or Stan [63] to obtain this posterior.

6.3.1 Tabular Setting

We first evaluate the RASR framework in tabular MDP setting, for problems like river-swim, machine replacement and inventory management. We compare: i) *Nominal method* which only uses the expected model, ii) *Bayesian Confidence Region (BCR)* [76], iii) *Robustification with Sensible Value Functions (RSVF)* [76], iv) *RASR-VI with VaR* and v) *RASR-VI with CVaR* and vi) *RASR-VI with Entropic*. Note that RASR-VI algorithm presented in Algorithm 6 is specific to entropic risk measure and corresponds to our proposed RASR framework. The VaR and CVaR based RASR-VI methods are extensions to that algorithm, where the risk measure in line 5 of Algorithm 6 gets replaced with VaR or CVaR. These extensions are not theoretically sound, as they are difficult to interpret and violate the property of law invariance. We introduce them here for the sole purpose of comparison. We evaluate the learned policies on a test data-set and report the mean and RASR entropic return evaluated under the RASR framework.

RiverSwim We first take a modified version of the classic RiverSwim problem [103]. The states in this problem are arranged as a chain and labeled with an index increasing from left to right. The reward is assumed to be known and depends on the current state, action and the next state. There are two actions: going left or going right. The transition following action left is deterministic and leads to a next state that is on the left. Action right can lead

Table 6.2: Policy evaluation results for methods trained in RASR framework.

Methods		Riverswim	Machine Replacement	Inventory Management
Nominal	Mean	221.90	−12.46	226.47
	RASR	16.54	−128.17	60.12
BCR	Mean	107.77	−15.68	208.73
	RASR	46.15	−127.53	74.40
RSVF	Mean	220.81	−14.14	216.54
	RASR	1.59	−129.03	65.44
RASR-VI with VaR	Mean	220.81	−14.14	222.19
	RASR	1.59	−129.03	62.45
RASR-VI with CVaR	Mean	132.92	−14.08	216.52
	RASR	43.56	−127.83	69.09
RASR-VI (Algorithm 6)	Mean	49.99	−24.11	118.54
	RASR	**49.99**	**−120.89**	**83.50**

to three possible next states (left, current, right) with uniform probabilities. We assume a Dirichlet prior for the transition distributions. Given some samples generated from the true distribution, we fit a JAGS [63] model to draw transition samples from Bayesian posterior.

Machine Replacement Next, we use an instance of the Machine Replacement problem (see e.g. Figure 3 of [104]) that consists of 10 states and 2 actions. States 0 to 7 describe the normal aging of the machine. States R1(index=9) and R2(index=8) represent two possible stages of repairs. R1 indicates for a normal repair with cost 2 and R2 indicates a harder repair with cost 10. Actions are labeled as 0 and 1, representing *do nothing* and *repair* respectively. An additional cost of 20 is incurred if the age of the machine reaches 8.

Inventory Management We then evaluate our RASR-entropic method on an instance of inventory management problem [3, 4]. This problem is formulated as an MDP with discrete state and action spaces. The state represents the inventory level and the action determines how much product to order to meet customer demands. The demand is stochastic and determines the transitions to next inventory levels. There is inherent stochasticity in transition

dynamics because of this randomness in demand realization and we model this as a Poisson distribution with a rate parameter λ. This demand distribution is moreover uncertain and is modeled with a Gamma distribution as prior with parameters: shape $k = 4$ and scale $\theta = 6$. We draw $n = 200$ samples from the true demand distribution and then compute the posterior Gamma distribution from the prior. The purchase cost and sale price for the problem are set to be 2.49 and 4.99 respectively. Ordering products to restock the inventory helps to meet demands, but unsold products incur a holding cost of 0.05.

Table 6.2 provides a comparison of different methods on all three tabular domains. The Nominal method provides the highest expected return estimates. But it does not take the variability of the model or transition into consideration and therefore performs poorly in the RASR entropic metric. The BCR and RSVF methods can provide certain robustness guarantees as discussed in [76], but they are only able to offer a very pessimistic estimate for the returns. Among methods involving risk measures, RASR VaR and RASR CVaR methods provide a slightly higher expected return estimates. But they are not time consistent and therefore cannot provide any performance guarantee. Their performance under RASR evaluation is also conservative. The RASR entropic method provides the best RASR performance in all problem domains.

6.3.2 Scaled-up Continuous Setting

We now extend our empirical study beyond tabular setting and evaluate our *RASR-AC with Entropic Risk Measure* algorithm on the classic *cart-pole* benchmark domain. We compare our algorithm with several baseline methods like: i) General AC [2, 83] ii) *Soft-Robust AC* [75], iii) *RASR-AC with VaR*, and iv) *RASR-AC with CVaR*.

Cart-Pole In this experiment, we evaluate our algorithm on cart-pole, a standard RL benchmark problem [2, 82]. The domain consists of a four dimensional and continuous state space. The task here is to balance a pole upright atop a cart by pushing the cart left or right

on a friction less track. We implement RASR-AC algorithm as described in Algorithm 7 with five different risk measures. We use a simple neural network of 1 fully connected hidden layer with 128 weights and *ReLu* activations. We use ADAM optimizer to minimize all the corresponding loss functions. We explicitly introduce a noise in the environment by altering the mass of the pole from a finite set of preset values. We train the agent on this perturbed version of the environment and then evaluate the policy on an environment perturbed with a different set of values for the pole mass.

Table 6.3: Evaluating AC policies for the cart-pole problem.

	General AC	**Soft-Robust AC**	**RASR-AC VaR**	**RASR-AC CVaR**	**RASR-AC Entropic**
RASR Return Estimates	112.11	102.49	105.18	127.82	**143.6**

A comparison is provided in Table 6.3. We run each algorithm for 10 different random seeds and then report the average return estimated for different policies. The General AC method [2, 83] optimizes for the expected value and therefore does not perform well in the RASR entropic metric that we care. The other three variants: Soft-Robust, VaR and CVaR based AC methods also perform reasonably well. But our entropic risk measure based AC method, which specifically optimizes a RASR entropic objective, outperforms all other variants in the evaluation by a good margin.

One important point to make here is that, the statistics presented in Table 6.3 for evaluating AC methods may depend on the random seed used in the experiment. This is a common reproducibility issue for many deep RL class of algorithms [105]. Also note that, the theoretical analysis presented in previous sections do not necessarily extend into this neural-network based AC setting. The main message of this experiment is that, our proposed RASR framework can be scaled up for larger problems with continuous state spaces to learn reasonable policies.

6.4 Contributions

In this chapter, I presented a unified Risk-Averse Soft-Robust (RASR) framework to simultaneously quantify and mitigate both model and transition uncertainties. I derived Bellman-style optimality equation for the RASR framework and presented a corresponding value iteration algorithm. To allow for scalability, I also derived gradient update formula to optimize the RASR objective and presented an actor-critic algorithm. I independently derived a finite sample convergence analysis for entropic risk measure and also empirically validated the usefulness of the RASR framework on several problem domains. The whole work has been done under close supervision of my advisor and thanks to Jia Lin Hau for joining this project recently.

CHAPTER 7
CONCLUSION

This thesis proposed several new approaches toward constructing tighter and more realistic robust solutions for problems involving sequential decision making. We proposed and examined the idea of incorporating weights into norm-bounded ambiguity sets to customize them for specific problems. We also have designed techniques to approximate near-optimal ambiguity sets and have validated their utilities. Though they operate in a restrictive s, a-rectangular setup and the empirical evidence indicate that they are still conservative, they still show significant improvement over prior methods while keeping the theoretical guarantees intact.

Incorporating robustness into CMDPs provide significant practical advantages in computing policies that are robust toward both objective and constraints. This thesis takes a step toward computing reasonable solutions for RCMDPs and contributed in theoretical and empirical developments.

In high-stakes practical problems, it is important to quantify and manage risk that arises from inherently stochastic transition probabilities or from uncertain models. Unlike other prior works that address each one of these sources of uncertainty independently, this thesis proposed a unified Risk-Averse Soft-Robust (RASR) framework that quantifies both model and transition uncertainties. Detailed theoretical and empirical analysis of RASR are also reported in this thesis.

We have evaluated all our methods on various problem domains that mimic the utilities and challenges of practical problems that we target. Our evaluation draws an encouraging

picture in all the benchmark domains and makes us pretty optimistic about what they are able to offer us. However, the full potentials of the methods remain yet to be discovered. Applying and deploying these methods into actual real-world applications can only lead to a *true* practical evaluation of these methods. With parallel progresses in many different technologies involving artificial intelligence, we are optimistic that such evaluation will become feasible soon enough.

This journey toward computing robust and practical solutions for reinforcement learning problems is by no means complete. More research needed about constructing even better ambiguity sets and also taking them beyond the rectangularity assumption while keeping them tractable and theoretically sound. The idea of robust-CMDPs is promising and it remains an interesting open direction to further advance our understanding about it. The RASR framework can deal with both epistemic and aleatory uncertainties together. Incorporation of risk measures other than entropic risk measure used in this thesis remains to be explored.

LIST OF REFERENCES

[1] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

[2] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. The MIT Press Cambridge, 2018.

[3] Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 2005.

[4] Paul H. Zipkin. *Foundations of Inventory Management*. 2000.

[5] Andrea Tirinzoni, Marek Petrik, Xiangli Chen, and Brian Ziebart. Policy-conditioned uncertainty sets for robust Markov decision processes. *Advances in Neural Information Processing Systems*, 2018.

[6] Marc Kery and Michael Schaub. *Bayesian Population Analysis Using WinBUGS*. Elsevier Science, 2012.

[7] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei a Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

[9] Hannes Eriksson and Dimitrakakis Christos. Epistemic Risk-Sensitive Reinforcement Learning. 2019.

[10] Aharon Ben-Tal, Dmitris Bertsimas, and David B. Brown. A Soft Robust Model for Optimization Under Ambiguity. *Operations Research*, 2010.

[11] Esther Derman, Daniel Mankowitz, Timothy Mann, and Shie Mannor. A Bayesian Approach to Robust Reinforcement Learning. *Uncertainty in Artificial Intelligence (UAI)*, 2019.

[12] Csaba Szepesvari. Algorithms for reinforcement learning. *Morgan and Claypool Publisher*, 2010.

[13] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. 1996.

[14] Richard Bellman. *Dynamic Programming*. 1957.

[15] Christos H Papadimitriou and John Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12:441–450, 1987.

[16] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the Complexity of Solving Markov Decision Problems. 2013.

[17] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control. *Athena Scientific*, 2012.

[18] Sasche Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. 2012.

[19] Marek Petrik, Mohammad Ghavamzadeh, and Yinlam Chow. Safe Policy Improvement by Minimizing Robust Baseline Regret. *Advances in Neural Information Processing Systems*, 2016.

[20] Philip S. Thomas, Georgios Teocharous, and Mohammad Ghavamzadeh. High Confidence Off-Policy Evaluation. In *Annual Conference of the AAAI*, 2015.

[21] Lihong Li, Rémi Munos, and Csaba Szepesvári. Toward Minimax Off-policy Value Estimation. 2015.

[22] Nan Jiang and Lihong Li. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2015.

[23] Shiau Hong Lim, Huan Xu, and Shie Mannor. Reinforcement Learning in Robust Markov Decision Processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[24] GA Hanasusanto and Daniel Kuhn. Robust Data-Driven Dynamic Programming. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[25] Kevin Murphy. *Machine Learning: A Probabilistic Perspective*. 2012.

[26] Romain Laroche and Paul Trichelair. Safe Policy Improvement with Baseline Bootstrapping, 2019.

[27] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[28] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Fast Bellman Updates for Robust MDPs. *Proceedings of Machine Learning Research (PMLR)*, 2018.

[29] Yann Le Tallec. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes*. PhD thesis, MIT, 2007.

[30] Wolfram Wiesemann, Daniel Kuhn, and Berc Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 2013.

[31] Shie Mannor, Ofir Mebel, and Huan Xu. Robust MDPs with k-rectangular uncertainty. *Mathematics of Operations Research*, 2016.

[32] Andrea Tirinzoni, Xiangli Chen, Marek Petrik, and Brian D Ziebart. Policy-Conditioned Uncertainty Sets for Robust Markov Decision Processes. 2018.

[33] Vineet Goyal and Julien Grand-Clement. Robust Markov Decision Process: Beyond Rectangularity. Technical report, 2018.

[34] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L_1 deviation of the empirical distribution. 2003.

[35] P Auer, Thomas Jaksch, and R Ortner. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 2010.

[36] Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 2005.

[37] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 2005.

[38] Dimitris Bertsimas, Nathan Kallus, and Vishal Gupta. *Data-driven robust optimization*. Springer Berlin Heidelberg, 2017.

[39] J Andrew Bagnell, Andrew Y Ng, and Jeff G Schneider. Solving Uncertain Markov Decision Processes. *Carnegie Mellon Research Showcase*, 2001.

[40] S Kalyanasundaram, E K P Chong, and N B Shroff. Markov decision processes with uncertain transition rates: Sensitivity and robust control. 2002.

[41] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial Policy Iteration for L1-Robust Markov Decision Processes. 2020.

[42] Majid Alkaee Taleghan, Thomas G. Dietterich, Mark Crowley, Kim Hall, and H. Jo Albers. PAC Optimal MDP Planning with Application to Invasive Species Management. *Journal of Machine Learning Research*, 2015.

[43] Alexander Strehl and Michael Littman. An analysis of model-based Interval Estimation for Markov Decision Processes. *Journal of Computer and System Sciences*, 2008.

[44] Vishal Gupta. Near-optimal Bayesian ambiguity sets for distributionally robust optimization. *Management Science*, 2019.

[45] Reazul Hasan Russel and Marek Petrik. Beyond confidence regions: Tight Bayesian ambiguity sets for robust MDPs. *Advances in Neural Information Processing Systems*, 2019.

[46] Arnab Nilim and Laurent El Ghaoui. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research*, 2005.

[47] Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Renner. Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Management Science*, 2013.

[48] Arnab Nilim and Laurent El Ghaoui. Robust solutions to Markov decision problems with uncertain transition matrices. *Operations Research*, 2004.

[49] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein Robust Reinforcement Learning. 2019.

[50] Marek Petrik, Yinlam Chow, and Mohammad Ghavamzadeh. Safe Policy Improvement by Minimizing Robust Baseline Regret. In *ICML Workshop on Reliable Machine Learning in the Wild*, 2016.

[51] Huan Xu and Shie Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 2012.

[52] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 2003.

[53] Dimitri P Bertsekas and John N Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific Belmont, MA, 2002.

[54] TG Dietterich, MA Taleghan, and Mark Crowley. PAC optimal planning for invasive species management: Improved exploration for reinforcement learning from simulator-defined MDPs. *The AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[55] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 2008.

[56] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[57] Vishal Gupta. Near-Optimal Bayesian Ambiguity Sets for Distributionally Robust Optimization. 2015.

[58] Huan Xu and Shie Mannor. Parametric regret in uncertain Markov decision processes. *Proceedings of the IEEE Conference on Decision and Control*, 2009.

[59] Philip S. Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference of Machine Learning (ICML)*, 2016.

[60] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and Efficient Off-Policy Reinforcement Learning. 2016.

[61] Alexander L Strehl. *Probably Approximately Corrct (PAC) Exploration in Reinforcement Learning*. PhD thesis, 2007.

[62] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 2010.

[63] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3rd edition, 2014.

[64] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on stochastic programming: Modeling and theory*. 2014.

[65] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[66] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming*. SIAM, 2009.

[67] Marek Petrik and Dharmashankar Subramanian. RAAM : The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2014.

[68] Eitan Altman. Constrained Markov Decision Processes. 2004.

[69] Shih Chieh Lin, Yunqi Zhang, Chang Hong Hsu, Matt Skach, Md E. Haque, Lingjia Tang, and Jason Mars. The architectural implications of autonomous driving: Constraints and acceleration. *ACM SIGPLAN Notices*, 2018.

[70] Mahmoud El Chamiea, Yue Yu, and Behcet Acikmese. Convex synthesis of randomized policies for controlled markov chains with density safety upper bound constraints. 2016.

[71] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces, 2018.

[72] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 2005.

[73] Jeroen Van Baar, Alan Sullivan, Radu Cordorel, Devesh Jha, Diego Romeres, and Daniel Nikovski. Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019.

[74] Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via Sampling. 2014.

[75] Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, and Shie Mannor. Soft-robust actor-critic policy-gradient. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[76] Reazul Hasan Russel and Marek Petrik. Beyond Confidence Regions: Tight Bayesian Ambiguity Sets for Robust MDPs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[77] Santiago Paternain, Luiz F.O. Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Conference on Neural Information Processing Systems*, 2019.

[78] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR optimization in MDPs. *Advances in Neural Information Processing Systems*, 2014.

[79] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv*, 2019.

[80] Vivek S. Borkar. Stochastic Approximation: A Dynamical Systems Viewpoint. *International Statistical Review*, 2009.

[81] Bahram Behzadian, Reazul Hasan Russel, and Marek Petrik. High-Confidence Policy Optimization: Reshaping Ambiguity Sets in Robust MDPs. 2019.

[82] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. Technical report, arXiv:1606.01540v1, 2016.

[83] Vijay R. Konda and John N. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 2003.

[84] Vivek Borkar and Rahul Jain. Risk-Constrained Markov Decision Processes. *IEEE Transactions on Automatic Control*, 2014.

[85] R. Tyrrell Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2000.

[86] Berend Roorda, Hans Schumacher, and Jacob Engwerda. Coherent acceptability measures in multiperiod models. *Mathematical Finance*, 2005.

[87] L.A. Prashanth and Mohammad Ghavamzadeh. Variance-constrained Actor-Critic Algorithms for Discounted and Average Reward MDPs. *Machine Learning Journal*, 2016.

[88] Aviv Tamar, Dotan Di Castro, and Shie Mannor. Temporal Difference Methods for the Variance of the Reward To Go. *International Conference on Machine Learning*, 2013.

[89] Elita A. Lobo, Mohammad Ghavamzadeh, and Marek Petrik. Soft-Robust Algorithms for Batch Reinforcement Learning. *Arxiv*, 2021.

[90] Takuya Hiraoka, Takahisa Imagawa, Tatsuya Mori, Takashi Onishi, and Yoshimasa Tsuruoka. Learning Robust Options by Conditional Value at Risk Optimization. *Neural Information Processing Systems*, 2019.

[91] E. Delage and S. Mannor. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *Operations Research*, 2010.

[92] David Nass, Boris Belousov, and Jan Peters. Entropic Risk Measure in Policy Search. *Investment Management and Financial Innovations*, 2020.

[93] Yingjie Fei, Zhuoran Yang, Yudong Chen, Zhaoran Wang, and Qiaomin Xie. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. *arXiv*, 2020.

[94] Hannes Eriksson and Christos Dimitrakakis. Epistemic risk-sensitive reinforcement learning. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2020.

[95] Aviv Tamar, Dotan Di Castro, and Shie Mannor. Policy gradients with variance related risk criteria. *International Conference on Machine Learning*, 2012.

[96] Daniel J. Mankowitz, Nir Levine, Rae Jeong, Yuanyuan Shi, Jackie Kay, Abbas Abdolmaleki, Jost Tobias Springenberg, Timothy Mann, Todd Hester, and Martin Riedmiller. Robust Reinforcement Learning for Continuous Control with Model Misspecification. 2019.

[97] Hans Follmer and Alexander Schied. *Stochastic Finance: An Introduction in Discrete Time*. 2004.

[98] Kai Detlefsen and Giacomo Scandolo. Conditional and dynamic convex risk measures. *Finance and Stochastics*, 2005.

[99] Hans Föllmer and Thomas Knispel. Entropic Risk Measures: Coherence Vs. Convexity, Model Ambiguity and Robust Large Deviations. *Stochastics and Dynamics*, 2011.

[100] H. Follmer and A. Schied. Convex and coherent risk measures. *Preprint*, 2008.

[101] Aharon Ben-Tal. An old-new concept of convex risk measures: The optimized certainty equivalent. *Mathematical Finance*, 2007.

[102] David B. Brown. Large deviations bounds for estimating conditional value-at-risk. *Operations Research*, 2007.

[103] Alexander L Strehl and Michael L Littman. An Analysis of Model-Based Interval Estimation for Markov Decision Processes. *Elsevier*, 2008.

[104] Erick Delage and Shie Mannor. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *Investment Management and Financial Innovations*, 2014.

[105] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *AAAI Conference on Artificial Intelligence*, 2018.

[106] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013.

APPENDIX A

APPENDICES

A.1 Supplementary Materials for Chapter 3

A.1.1 Dual Norm of Weighted L_1-norm

Lemma A.1.1. *Let $\|\cdot\|_{1,w}$ be the weighted L_1 norm on $\mathbb{R}^n$. The associated dual norm $\|\cdot\|_{\infty,\frac{1}{w}}$ is defined as:*

$$\|z\|_{\infty,\frac{1}{w}} = \sup\{z^\mathsf{T} x | \|x\|_{1,w} \leq 1, w \in \mathbb{R}^n_{++}\}.$$

Proof. Assume we are given a set of positive weights $w \in \mathbb{R}^n_{++}$ for the following weighted L_1 optimization problem:

$$\begin{aligned} \max_x \quad & z^\mathsf{T} x \\ \text{s.t.} \quad & \|x\|_{1,w} \leq 1 \ . \end{aligned} \tag{A.1}$$

we have:

$$\begin{aligned} x^\mathsf{T} z = \sum_{i=1}^n x_i z_i &\leq \sum_{i=1}^n |x_i z_i| \\ &\overset{(a)}{\leq} \sum_{i=1}^n |x_i||z_i| = \sum_{i=1}^n w_i |x_i| \frac{1}{w_i} |z_i| \\ &\leq \max_{i=1,\ldots,n} \left\{ \frac{1}{w_i} |z_i| \right\} \cdot \sum_{i=1}^n w_i |x_i| = \max_{i=1,\ldots,n} \left\{ \frac{1}{w_i} |z_i| \right\} \cdot \|x\|_{1,w} \\ &\overset{(b)}{\leq} \max_{i=1,\ldots,n} \left\{ \frac{1}{w_i} |z_i| \right\} = \|z\|_{\infty,\frac{1}{w}} \ . \end{aligned}$$

Here, (a) follows from the Cauchy-Schwarz inequality and (b) follows from the constraint

$\|x\|_{1,w} \leq 1$ of (A.1). $\square$

A.1.2 Proof of Theorem 3.3.1

Proof. The inner optimization objective function for RMDPs for L_p-constrained ambiguity sets are defined as follows:

$$q(z) = \min_{p \in \Delta^{\mathcal{S}}} \left\{ p^{\mathsf{T}} z : \|p - \bar{p}\| \leq \psi \right\} \ .$$

Let $q = p - \bar{p}$. We can reformulate the optimization problem using the new variable q:

$$\begin{aligned} \min_{q} \quad & (q + \bar{p})^{\mathsf{T}} z \\ \text{s.t.} \quad & \|q\| \leq \psi \\ & \mathbf{1}^{\mathsf{T}}(q + \bar{p}) = 1 \implies \mathbf{1}^{\mathsf{T}} q = 0 \\ & q \geq -\bar{p} \ . \end{aligned}$$

If ψ is sufficiently small and $\bar{p}$ is sufficiently large, we can relax the problem by dropping the $q \geq -\bar{p}$ constraint. Since $\bar{p}^{\mathsf{T}} z$ is a fixed number, we continue with:

$$\begin{aligned} \bar{p}^{\mathsf{T}} z + \min_{q} \quad & q^{\mathsf{T}} z \\ \text{s.t.} \quad & \|q\| \leq \psi \\ & \mathbf{1}^{\mathsf{T}} q = 0 \end{aligned}$$

We then change the minimization form to maximization:

$$\begin{aligned} \bar{p}^{\mathsf{T}} z - \max_{q} \quad & - q^{\mathsf{T}} z \\ \text{s.t.} \quad & \|q\| \leq \psi \\ & \mathbf{1}^{\mathsf{T}} q = 0 \end{aligned}$$

By applying the method of Lagrange multipliers, we obtain:

$$\min_{\lambda} \max_{q} \quad -q^{\mathsf{T}} z - \lambda(q^{\mathsf{T}}\mathbf{1}) = q^{\mathsf{T}}(-z - \lambda\mathbf{1})$$
$$\text{s.t.} \quad \|q\| \leq \psi$$

Letting $x = \frac{q}{\psi}$, we get:

$$\min_{\lambda} \max_{x} \quad \psi \cdot x^{\mathsf{T}}(-z - \lambda\mathbf{1})$$
$$\text{s.t.} \quad \|x\| \leq 1$$

Given the definition of the *dual norm*, $\|z\|_\star = \sup\{z^\intercal x \mid \|x\| \leq 1\}$, we have:

$$q(z) \geq \bar{p}^{\mathsf{T}} z - \min_{\lambda} \quad \psi\|z + \lambda\mathbf{1}\|_\star \ .$$

□

A.1.3 Proof of Theorem 3.4.1 (Weighted L_1 Error Bound)

In this section, we describe a proof of a bound on the $L_{1,w}$ distance between the estimated transition probabilities $\bar{p}$ and the true one $p^\star$ over each state $s \in \mathcal{S} = \{1, \ldots, S\}$ and action $a \in \mathcal{A} = \{1, \ldots, A\}$. The proof is an extension to Lemma C.1 (L1 error bound) in [45].

Proof. Let $q_{s,a} = \bar{p}_{s,a} - p^\star_{s,a}$. To shorten notation in the proof, we omit the s, a indexes when there is no ambiguity. We assume that all weights are non-negative. First, we will express the $L_{1,w}$ norm of q in terms of an optimization problem. It is worth noting that $\mathbf{1}^{\mathsf{T}} q = 0$. Let $\mathbf{1}_{\mathcal{Q}_1}, \mathbf{1}_{\mathcal{Q}_2} \in \mathbb{R}^{\mathcal{S}}$ be the indicator vectors for some subsets $\mathcal{Q}_1, \mathcal{Q}_2 \subset \mathcal{S}$ where $\mathcal{Q}_2 = \mathcal{S} \setminus \mathcal{Q}_1$. According to Lemma A.1.1 we have:

$$\begin{aligned}
\|q\|_{1,w} &= \max_{z} \left\{ z^{\mathsf{T}} q : \|z\|_{\infty,\frac{1}{w}} \leq 1 \right\} \\
&= \max_{\mathcal{Q}_1, \mathcal{Q}_2 \in 2^{\mathcal{S}}} \left\{ \mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W q + \mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W(-q) : \mathcal{Q}_2 = \mathcal{S} \setminus \mathcal{Q}_1 \right\} \ .
\end{aligned}$$

Here weights are on the diagonal entries of W. Using the expression above, we can bound

the probability as follows:

$$
\begin{aligned}
&\mathbb{P}\left[\max_{\mathcal{Q}_1,\mathcal{Q}_2\in 2^{\mathcal{S}}}\left\{\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W q + \mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W(-q)\right\} \geq \psi\right] \\
&\overset{(a)}{\leq} \mathbb{P}\left[\max_{\mathcal{Q}_1\in 2^{\mathcal{S}}}\left\{\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W q\right\} \geq \frac{\psi}{2}\right] + \mathbb{P}\left[\max_{\mathcal{Q}_2\in 2^{\mathcal{S}}}\left\{\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W(-q)\right\} \geq \frac{\psi}{2}\right] \\
&\leq \sum_{\mathcal{Q}_1\in 2^{\mathcal{S}}} \mathbb{P}\left[\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W q \geq \frac{\psi}{2}\right] + \sum_{\mathcal{Q}_2\in 2^{\mathcal{S}}} \mathbb{P}\left[\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W(-\boldsymbol{q}) \geq \frac{\psi}{2}\right] \\
&= \sum_{\mathcal{Q}_1\in 2^{\mathcal{S}}} \mathbb{P}\left[\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W(\bar{p} - p^\star) \geq \frac{\psi}{2}\right] + \sum_{\mathcal{Q}_2\in 2^{\mathcal{S}}} \mathbb{P}\left[\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W(-\bar{p} + p^\star) \geq \frac{\psi}{2}\right] \\
&\overset{(b)}{\leq} \sum_{\mathcal{Q}_1\in 2^{\mathcal{S}}} \exp\left(-\frac{\psi^2 n}{2\|\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W\|_\infty^2}\right) + \sum_{\mathcal{Q}_2\in 2^{\mathcal{S}}} \exp\left(-\frac{\psi^2 n}{2\|\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W\|_\infty^2}\right) \\
&\overset{(c)}{=} 2\sum_{i=1}^{S-1} 2^{S-i} \exp\left(-\frac{\psi^2 n}{2w_i^2}\right) \ .
\end{aligned}
$$

(a) follows from union bound, and (b) follows from Hoeffding's inequality. (c) follows by $\mathcal{Q}_1^c = \mathcal{Q}_2$ and sorting weights $w = \{w_1, \ldots, w_n\}$ in non-increasing order.

□

Theorem A.1.2 (weighted L_1 error bound using Bernstein's inequality)**.** *Suppose that $\bar{p}_{s,a}$ is the empirical estimate of the transition probability obtained from $n_{s,a}$ samples for some $s \in \mathcal{S}$ and $a \in \mathcal{A}$. If the weights $w \in \mathbb{R}_{++}^S$ are sorted in non-increasing order $w_i \geq w_{i+1}$, then the following holds when using Bernstein's inequality:*

$$
\mathbb{P}\left[\|\bar{p}_{s,a} - p_{s,a}^\star\|_{1,w} \geq \psi_{s,a}\right] \leq 2\sum_{i=1}^{S-1} 2^{S-i} \exp\left(-\frac{3\psi^2 n}{6w_i^2 + 4\psi w_i}\right)
$$

where $w \in \mathbb{R}_{++}^S$ is the vector of weights. The weights are sorted in non-increasing order.

Proof. The proof is similar to the proof of 3.4.1 until section b. The proof continues from section (b) as follows:

$$
\overset{(b)}{\leq} \sum_{\mathcal{Q}_1\in 2^{\mathcal{S}}} \exp\left(-\frac{3\psi^2 n}{24\sigma^2 + 4c\psi}\right) + \sum_{\mathcal{Q}_2\in 2^{\mathcal{S}}} \exp\left(-\frac{3\psi^2 n}{24\sigma^2 + 4c\psi}\right)
$$

$$\overset{(c)}{\leq} \sum_{\mathcal{Q}_1 \in 2^{\mathcal{S}}} \exp\left(-\frac{3\psi^2 n}{6\|\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W\|_\infty^2 + 4\psi\|\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W\|_\infty}\right) + \sum_{\mathcal{Q}_2 \in 2^{\mathcal{S}}} \exp\left(-\frac{3\psi^2 n}{6\|\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W\|_\infty^2 + 4\psi\|\mathbf{1}_{\mathcal{Q}_2}^{\mathsf{T}} W\|_\infty}\right)$$
$$\overset{(d)}{=} 2\sum_{i=1}^{S-1} 2^{S-i} \exp\left(-\frac{3\psi^2 n}{6w_i^2 + 4\psi w_i}\right) .$$

Here (b) follows from Bernstein's inequality where σ^2 is the mean of variance of random variables, and c is their upper bound [106]. In the weighted case, with conservative estimate of variance $\sigma^2 = \|\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W\|_\infty^2/4$, and $c = \|\mathbf{1}_{\mathcal{Q}_1}^{\mathsf{T}} W\|_\infty$, because the random variables are drawn from *Bernoulli* distribution with the maximum possible variance of $1/4$. (d) follows by sorting weights w in non-increasing order.

□

A.2 Supplementary Materials for Chapter 4

The following proposition shows that the guarantee of a safe estimate on the return is achieved when the true transition model is contained in the ambiguity set.

Lemma A.2.1. *Suppose that an ambiguity set $\mathcal{P}$ satisfies $\mathbb{P}_{\mathcal{D}}\left[p_{s,a}^\star \in \mathcal{P}_{s,a} \mid P^\star\right] \geq 1-\delta/(SA)$ for each state s and action a. Then:*

$$\mathbb{P}_{\mathcal{D}}\left[\hat{v}_{\mathcal{P}}^\pi \leq v_{P^\star}^\pi,\ \forall \pi \in \Pi \mid P^\star\right] \geq 1-\delta .$$

Proof. We omit $\mathcal{P}$ and $P^\star$ from the notation in the proof since they are fixed. From Proposition (2.4.1), we have that $\hat{v}^\pi \leq v^\pi$ if

$$\widehat{T}^\pi \hat{v}^\pi \leq T^\pi \hat{v}^\pi .$$

That is, for each state s and action a:

$$\min_{p \in \mathcal{P}_{s,a}} p^{\mathsf{T}} \hat{v}^\pi \leq (p_{s,a}^\star)^{\mathsf{T}} \hat{v}^\pi.$$

Using the identity above, the probability that the robust value function is a lower bound can be bounded as follows:

$$\mathbb{P}_{\mathcal{D}}\left[\hat{v}^{\pi}_{\mathcal{P}} \leq v^{\pi}_{P},\ \forall \pi \in \Pi \mid P^{\star}\right] = \mathbb{P}_{\mathcal{D}}\left[\min_{p \in \mathcal{P}_{s,a}} p^{\mathsf{T}}\hat{v}^{\pi} \leq (p^{\star}_{s,a})^{\mathsf{T}}\hat{v}^{\pi},\ \forall \pi \in \Pi, s \in \mathcal{S}, a \in \mathcal{A} \mid P^{\star}\right] \geq$$

$$\geq \mathbb{P}_{\mathcal{D}}\left[(p^{\star}_{s,a})^{\mathsf{T}}\hat{v}^{\pi} \leq (p^{\star}_{s,a})^{\mathsf{T}}\hat{v}^{\pi},\ \forall \pi \in \Pi, s \in \mathcal{S}, a \in \mathcal{A} \mid P^{\star} \in \mathcal{P}, P^{\star}\right]\mathbb{P}_{\mathcal{D}}\left[P^{\star} \in \mathcal{P} \mid P^{\star}\right] +$$

$$+\mathbb{P}_{\mathcal{D}}\left[P^{\star} \notin \mathcal{P} \mid P^{\star}\right] \geq 1\,\mathbb{P}_{\mathcal{D}}\left[P^{\star} \in \mathcal{P} \mid P^{\star}\right] + 0\,\mathbb{P}_{\mathcal{D}}\left[P^{\star} \notin \mathcal{P} \mid P^{\star}\right] \geq$$

$$\geq \mathbb{P}_{\mathcal{D}}\left[P^{\star} \in \mathcal{P} \mid P^{\star}\right]\ .$$

Now, from the union bound over all states and actions, we get:

$$\mathbb{P}_{\mathcal{D}}\left[\hat{v}^{\pi} > v^{\pi} | P^{\star}\right] \leq \mathbb{P}_{\mathcal{D}}\left[P^{\star} \notin \mathcal{P} \mid P^{\star}\right] \leq \sum_{s \in \mathcal{S}}\sum_{a \in \mathcal{A}} \mathbb{P}_{\mathcal{D}}\left[p^{\star}_{s,a} \notin \mathcal{P}_{s,a} \mid P^{\star}\right] \leq \delta\ ,$$

which completes the proof. □

The next proposition is the Bayesian equivalent of Lemma (A.2.1).

Lemma A.2.2. *Suppose that an ambiguity set $\mathcal{P}$ satisfies $\mathbb{P}_{P^{\star}}\left[p^{\star}_{s,a} \in \mathcal{P}_{s,a} \mid \mathcal{D}\right] \geq 1-\delta/(SA)$ for each state s and action a. Then:*

$$\mathbb{P}_{P^{\star}}\left[\hat{v}^{\pi}_{\mathcal{P}} \leq v^{\pi}_{P^{\star}},\ \forall \pi \in \Pi \mid \mathcal{D}\right] \geq 1 - \delta\ .$$

Proof. We omit $\mathcal{P}$ and $P^{\star}$ from the notation in the proof since they are fixed. From (2.4.1), we have that $\hat{v}^{\pi} \leq v^{\pi}$ if

$$\widehat{T}^{\pi}\hat{v}^{\pi} \leq T^{\pi}\hat{v}^{\pi}\ .$$

That is, for each state s and action a:

$$\min_{p \in \mathcal{P}_{s,a}} p^{\mathsf{T}}\hat{v}^{\pi} \leq (p^{\star}_{s,a})^{\mathsf{T}}\hat{v}^{\pi}.$$

Using the identity above, the probability that the robust value function is a lower bound can

be bounded as follows:

$$\mathbb{P}_{P^\star}\left[\hat{v}^\pi_{\mathcal{P}} \leq v^\pi_P,\ \forall \pi \in \Pi \mid \mathcal{D}\right] = \mathbb{P}_{P^\star}\left[\min_{p \in \mathcal{P}_{s,a}} p^\mathsf{T}\hat{v}^\pi \leq (p^\star_{s,a})^\mathsf{T}\hat{v}^\pi,\ \forall \pi \in \Pi, s \in \mathcal{S}, a \in \mathcal{A} \mid \mathcal{D}\right] \geq$$

$$\geq \mathbb{P}_{P^\star}\left[(p^\star_{s,a})^\mathsf{T}\hat{v}^\pi \leq (p^\star_{s,a})^\mathsf{T}\hat{v}^\pi,\ \forall \pi \in \Pi, s \in \mathcal{S}, a \in \mathcal{A} \mid P^\star \in \mathcal{P}, \mathcal{D}\right]\mathbb{P}_{P^\star}\left[P^\star \in \mathcal{P} \mid \mathcal{D}\right] +$$

$$+\mathbb{P}_{P^\star}\left[P^\star \notin \mathcal{P} \mid \mathcal{D}\right] \geq 1\,\mathbb{P}_{P^\star}\left[P^\star \in \mathcal{P} \mid \mathcal{D}\right] + 0\,\mathbb{P}_{P^\star}\left[P^\star \notin \mathcal{P} \mid \mathcal{D}\right] \geq$$

$$\geq \mathbb{P}_{P^\star}\left[P^\star \in \mathcal{P} \mid \mathcal{D}\right]\ .$$

Now, from the union bound over all states and actions, we get:

$$\mathbb{P}_{P^\star}\left[\hat{v}^\pi > v^\pi | \mathcal{D}\right] \leq \mathbb{P}_{P^\star}\left[P^\star \notin \mathcal{P} \mid \mathcal{D}\right] \leq \sum_{s \in \mathcal{S}}\sum_{a \in \mathcal{A}} \mathbb{P}_{P^\star}\left[p^\star_{s,a} \notin \mathcal{P}_{s,a} \mid \mathcal{D}\right] \leq \delta\ ,$$

which completes the proof. □

A.2.1 L_1 Concentration Inequality Bounds

In this section, we describe a new elementary proof of a bound on the L_1 distance between the estimated transition probability distribution and the true one. It simplifies the proofs of [34] but also leads to coarser bounds. Note that in the frequentist setting the ambiguity set $\mathcal{P}$ is a random variable that is a function of the dataset $\mathcal{D}$.

Recall that our ambiguity sets are defined as L_1 balls around the expected transition probabilities $\bar{p}_{s,a}$:

$$\mathcal{P}_{s,a} = \{p \in \Delta^S\ :\ \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a}\}\ . \tag{A.2}$$

Lemma A.2.1 implies that the size of the L_1 balls must be chosen as follows:

$$\mathbb{P}\left[\|\bar{p}(s,a) - p^\star(s,a)\|_1 \leq \psi_{s,a}\right] \geq 1 - \delta/(SA)\ . \tag{A.3}$$

We can now express the necessary size $\psi_{s,a}$ of the ambiguity sets in terms of $n_{s,a}$, which denotes the number of samples in $\mathcal{D}$ that originate with a state s and an action a.

Lemma A.2.3 (L_1 Error bound)**.** *Suppose that $\bar{p}_{s,a}$ is the empirical estimate of the transition probability obtained from $n_{s,a}$ samples for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then:*

$$\mathbb{P}\left[\|\bar{p}_{s,a} - p^\star_{s,a}\|_1 \geq \psi_{s,a}\right] \leq (2^S - 2)\exp\left(-\frac{\psi^2_{s,a} n_{s,a}}{2}\right) .$$

Therefore, for any $\delta \in [0,1]$:

$$\mathbb{P}\left[\|\bar{p}_{s,a} - p^\star_{s,a}\|_1 \leq \sqrt{\frac{2}{n_{s,a}}\log\frac{SA(2^S-2)}{\delta}}\right] \leq 1 - \delta/(SA) .$$

Proof. To shorten the notation, we omit the indexes s, a throughout the proof; for example $\bar{p}$ is used instead of the full $\bar{p}_{s,a}$. First, express the L_1 distance between two distributions $\bar{p}$ and $p^\star$ in terms of an optimization problem. Let $\mathbf{1}_{\mathcal{Q}} \in \mathbb{R}^{\mathcal{S}}$ be the indicator vector for some subset $\mathcal{Q} \subset \mathcal{S}$. Then:

$$\begin{aligned}
\|\bar{p} - p^\star\|_1 &= \max_z \left\{z^\mathsf{T}(\bar{p} - p^\star) \ : \ \|z\|_\infty \leq 1\right\} = \\
&= \max_{\mathcal{Q}\in 2^{\mathcal{S}}} \left\{\mathbf{1}_{\mathcal{Q}}^\mathsf{T}(\bar{p} - p^\star) - (\mathbf{1} - \mathbf{1}_{\mathcal{Q}})^\mathsf{T}(\bar{p} - p^\star) \ : \ 0 < |\mathcal{Q}| < m\right\} \\
&\overset{\text{(a)}}{=} 2\max_{\mathcal{Q}\in 2^{\mathcal{S}}} \left\{\mathbf{1}_{\mathcal{Q}}^\mathsf{T}(\bar{p} - p^\star) \ : \ 0 < |\mathcal{Q}| < m\right\} .
\end{aligned}$$

Here, (a) holds because $\mathbf{1}^\mathsf{T}(\bar{p} - p^\star) = 0$. Using the expression above, the target probability can be bounded as follows:

$$\begin{aligned}
\mathbb{P}\left[\|\bar{p} - p^\star\|_1 > \psi\right] &= \mathbb{P}\left[2\max_{\mathcal{Q}\in 2^{\mathcal{S}}} \left\{\mathbf{1}_{\mathcal{Q}}^\mathsf{T}(\bar{p} - p^\star) \ : \ 0 < |\mathcal{Q}| < m\right\} > \psi\right] \\
&\overset{\text{(a)}}{\leq} (|\mathcal{Q}| - 2)\max_{\mathcal{Q}\in 2^{\mathcal{S}}} \left\{\mathbb{P}\left[\mathbf{1}_{\mathcal{Q}}^\mathsf{T}(\bar{p} - p^\star) > \frac{\psi}{2}\right] \ : \ 0 < |\mathcal{Q}| < m\right\} \\
&\overset{\text{(b)}}{\leq} (|\mathcal{Q}| - 2)\exp\left(-\frac{\psi^2 n}{2}\right) = (2^S - 2)\exp\left(-\frac{\psi^2 n}{2}\right) .
\end{aligned}$$

The inequality (a) follows from union bound and the inequality (b) follows from the Hoeffding's inequality since $\mathbf{1}_{\mathcal{Q}}^\mathsf{T}\bar{p} \in [0,1]$ for any $\mathcal{Q}$ with the mean of $\mathbf{1}_{\mathcal{Q}}^\mathsf{T}\bar{p}^\star$. □

A.3 Supplementary Materials for Chapter 5

A.3.1 Proof of Theorem 5.2.1

Proof. The objective as specified in (5.4):

$$\mathfrak{L}(\pi_\theta, \lambda) = \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) - \lambda \beta$$

We first derive the gradient update rule of $\mathfrak{L}(\pi_\theta, \lambda)$ with respect to θ as below:

$$\begin{aligned}
\nabla_\theta \mathfrak{L}(\pi_\theta, \lambda) &= \sum_{\xi \in \Xi} \nabla_\theta p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \nabla_\theta \log p^{\pi_\theta}(\xi) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \nabla_\theta \log \left(p_0(s_0) \prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t) \pi_\theta(a_t|s_t) \right) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \nabla_\theta \left(\log p_0(s_0) + \sum_{t=0}^{T-1} \log p(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t) \right) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) \sum_{t=0}^{T-1} \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}
\end{aligned}$$

Next, we derive the gradient update rule for $\mathfrak{L}(\pi_\theta, \lambda)$ with respect to λ:

$$\begin{aligned}
\nabla_\lambda \mathfrak{L}(\pi_\theta, \lambda) &= \nabla_\lambda \left(\sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) \Big(g(\xi, r) + \lambda g(\xi, d) \Big) - \lambda \beta \right) \\
&= \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi) g(\xi, d) - \beta
\end{aligned}$$

□

A.3.2 Convergence Analysis of Algorithm

A.3.3 Assumptions

(A1) For any state s, policy $\pi_\theta(.|s)$ is continuously differentiable with respect to parameter θ and $\nabla_\theta \pi_\theta(.|s)$ is a Lipschitz function in θ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

(A2) The step size schedules $\{\zeta_2(t), \zeta_1(t)\}$ satisfy:

$$\sum_t \zeta_1(t) = \sum_t \zeta_2(t) = \sum_t \zeta_3(t) = \infty \tag{A.4}$$

$$\sum_t \zeta_1(t)^2, \sum_t \zeta_2(t)^2 \leq \infty \tag{A.5}$$

$$\zeta_1(t) = o\big(\zeta_2(t)\big) \tag{A.6}$$

These assumptions are basically standard step-size conditions for stochastic approximation algorithms [80]. Equation (A.4) ensures that the discretization covers the entire time axis. (A.5) ensures that the errors resulting from the discretization of the Ordinary Differential Equation (ODE) and errors due to the noise both becomes negligible asymptotically with probability one [80]. Equations (A.4) and (A.5) together ensures that the iterates asymptotically captures the behavior of the ODE. (A.6) mandates that, updates corresponding to $\zeta_1(t)$ is on a slower time scale than $\zeta_2(t)$.

A.3.4 Policy Gradient Algorithm

The general stochastic approximation scheme used by [80] is of the form:

$$x_{t+1} = t_n + a(t)[h(x_t) + \Delta_{t+1}] \tag{A.7}$$

where $\{\Delta_t\}$ are a sequence of integrable random variables representing the noise sequence

and $\{a_t\}$ are step sizes (e.g. $\zeta(t)$). The expression $h(x_t) + \Delta_{t+1}$ inside the square bracket is the noisy measurement where $h(x_t)$ and Δ_{t+1} are not separately available, only their sum is available. The terms of (A.7) need to satisfy below additional assumptions:

(A3) The function $h : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz. That is $\|h(x) - h(y)\| \leq L\|x - y\|$ for some $0 \leq L \leq \infty$.

(A4) $\{\Delta_t\}$ are martingale difference sequence:

$$\mathbb{E}[\Delta_{t+1}|x_n, \Delta_n, n \leq t] = 0$$

In addition to that, $\{\Delta_t\}$ are square-integrable:

$$\mathbb{E}[\|\Delta_{t+1}\|^2|x_n, \Delta_n, n \leq t] \leq K(1 + \|x_t\|^2) \text{ a.s. for } t \geq 0,$$

and for some constant $K > 0$.

Our proposed policy gradient algorithm is a two time-scale stochastic approximation algorithm. The parameter update iterations of the policy gradient algorithm are defined as below:

$$\theta_{t+1} = \theta_t + \zeta_2(t)\nabla_\theta \mathfrak{L}(\pi_\theta, \lambda) \tag{A.8}$$

$$\lambda_{t+1} = \lambda_t + \zeta_1(t)\nabla_\lambda \mathfrak{L}(\pi_\theta, \lambda) \tag{A.9}$$

These gradient update rules defined in (A.8) and (A.9) are in a special form as:

$$x_{t+1} = x_t + a(t)f(x_t, \epsilon_t), t \geq 0 \tag{A.10}$$

Where $\{\epsilon\}$ is a zero mean i.i.d. random variable representing noise. To apply general

convergence analysis techniques derived for (A.7) in [80], we take the special form in (A.10) and transform it to the general format of (A.7) as below:

$$h(x) = \mathbb{E}\left[f(x, \epsilon_1)\right] \text{ and } \Delta_{n+1} = f(x_n, \epsilon_{n+1}) - h(x_n) \tag{A.11}$$

With these transformation techniques, we obtain the general update for θ from (A.8):

θ update:

$$\theta_{t+1} = \theta_t + \zeta_2(t)\big[h(\theta_t, \lambda_t) + \Delta^{(1)}_{t+1}\big] \tag{A.12}$$

where, $f^{(1)}(\theta_t, \lambda_t) = \nabla_\theta L(\pi_\theta, \lambda)$ is the gradient w.r.t θ, $h(\theta_t, \lambda_t) = \mathbb{E}[f^{(1)}(\theta_t, \lambda_t)]$, and $\Delta^{(1)}_{t+1} = f^{(1)}(\theta_t, \lambda_t) - h(\theta_t, \lambda_t)$. Note that, the noise term ϵ is omitted because the noise is inherent in our sample based iterations.

Proposition A.3.1. *$h(\theta_t, \lambda_t)$ is Lipschitz in θ.*

Proof. Recall that the gradient of $\mathfrak{L}(\pi_\theta, \lambda)$ with respect to θ is:

$$\nabla_\theta \mathfrak{L}(\pi_\theta, \lambda) = \sum_{\xi \in \Xi} p^{\pi_\theta}(\xi)\Big(g(\xi, r) + \lambda g(\xi, d)\Big) \sum_{t=0}^{T-1} \frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \tag{A.13}$$

Assumption (A1) implies that, $\nabla_\theta \pi_\theta(a_t|s_t)$ in the equation (A.13) is a Lipschitz function in θ for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$. As the expectation of sum of $|T|$ number of Lipschitz functions is also Lipschitz, we conclude that $h(\theta_t, \lambda_t)$ is Lipschitz in θ.

□

Proposition A.3.2. *$\Delta^{(1)}_{t+1}$ of (A.12) satisfies assumption (A4).*

We transform our update rule of (A.9) as:

λ update:

$$\lambda_{t+1} = \lambda_t + \zeta_1(t)\big[g(\theta_t, \lambda_t) + \Delta^{(2)}_{t+1}\big] \tag{A.14}$$

where, $f^{(2)}(\theta_t, \lambda_t) = \nabla_\lambda L(\pi_\theta, \lambda)$ is the gradient w.r.t λ, $g(\theta_t, \lambda_t) = \mathbb{E}_M[f^{(2)}(\theta_t, \lambda_t)]$, and $\Delta^{(2)}_{t+1} = f^{(2)}(\theta_t, \lambda_t) - h(\theta_t, \lambda_t)$.

Notice that $\nabla_\lambda \mathfrak{L}(\pi_\theta, \lambda) = \sum_\xi \hat{p}^\theta(\xi) g(\xi, d) - \beta$ is a constant function of λ. And therefore, $g(\theta_t, \lambda_t)$ is a constant function of λ.

Proposition A.3.3. *$\Delta^{(2)}_{t+1}$ of (A.14) satisfies assumption (A4).*

We now focus on the singularly perturbed ODE obtained from (A.12) and (A.14).

$$\dot{\theta} = \zeta_2(t) h(\theta_t, \lambda_t) \tag{A.15}$$

$$\dot{\lambda} = \zeta_1(t) g(\theta_t, \lambda_t) \tag{A.16}$$

With assumption (A2), $\lambda(\cdot)$ is quasi-static from the perspective of $\theta(\cdot)$ turning (A.15) into an ODE. where λ is held fixed:

$$\dot{\theta} = \zeta_2(t) h(\theta_t, \lambda) \tag{A.17}$$

We additionally assume that:

(A5) (A.17) has a globally asymptotically stable equilibrium $x(\lambda)$ such that x is a Lipschitz map.

Assumption (A5) turns (A.16) into:

$$\dot{\lambda}(t) = g(x(\lambda_t), \lambda_t) \tag{A.18}$$

Let's further assume that:

(A6) The ODE (A.18) has a globally asymptotically stable equilibrium $\lambda^\star$.

(A7) $\sup_t(\|\theta_t\| + \|\lambda_t\|) < \infty$ almost surely.

Proof of Theorem 5.2.4

Proof. Above are the necessary conditions to apply Theorem 2 from chapter 6 of [80], which shows that $(\theta_t, \lambda_t) \to (x(\lambda^\star), \lambda^\star)$. Now the saddle point theorem assures that $\theta^\star = x(\lambda^\star)$ maximizes the Lagrange optimization problem stated in (5.4). □

A.4 Supplementary Materials for Chapter 6

A.4.1 Finite-Sample Convergence of Entropic Risk Measure

In this section, we first derive some auxiliary results that will be helpful in obtaining the finite sample guarantee for entropic risk measure. In this regard, we first formulate the Optimized Certainty Equivalent (OCE) for entropic risk measure and then show that the finite sample convergence properties of OCE can be used to derive the convergence bound for entropic risk measure.

With ϕ in Definition 6.1.2 being continuously differentiable and strictly concave, [101] shows that the supremum in (6.2) is uniquely obtained at $\eta_s \in \mathbb{R}$ as solution of:

$$\mathbb{E}\left[\phi'(X - \eta_s)\right] = 1 \tag{A.19}$$

Where ϕ' is the first derivative of ϕ w.r.t η_s. Therefore the optimal value of $S_\phi(X)$ is:

$$S_\phi(X) = \eta_s + \mathbb{E}\left[\phi(X - \eta_s)\right] \tag{A.20}$$

Proposition A.4.1. *The Optimal Certainty Equivalent (OCE) for utility function* $\phi(t) = \frac{1}{\alpha}\big(1 - \exp\left(-\alpha t\right)\big)$, $\forall t \in \mathbb{R}$ *and* $\alpha \in (0, 1]$ *is* $S_\phi(X) = -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]$. *Moreover, the negative of* $S^\alpha_\phi(X)$ *defines the entropic risk measure with confidence level* α.

Proof.

$$\begin{aligned}
&\phi(X - \eta_s) = \frac{1}{\alpha}\big(1 - \exp\left(\alpha\eta_s - \alpha X\right)\big) \\
&\Rightarrow \phi'(X - \eta_s) = \frac{1}{\alpha} \cdot \exp\left(\alpha\eta_s - \alpha X\right) \cdot \alpha \\
&\Rightarrow \phi'(X - \eta_s) = \exp\left(\alpha\eta_s - \alpha X\right)
\end{aligned}$$

From (A.19), we have:

$$
\begin{aligned}
&\mathbb{E}\left[\phi'(X-\eta_s)\right] = 1 \\
&\Rightarrow \mathbb{E}\left[\exp\left(\alpha\eta_s - \alpha X\right)\right] = 1 \\
&\Rightarrow \mathbb{E}\left[\exp\left(\alpha\eta_s\right).\exp\left(-\alpha X\right)\right] = 1 \\
&\Rightarrow \exp\left(\alpha\eta_s\right)\mathbb{E}\left[\exp\left(-\alpha X\right)\right] = 1 \\
&\Rightarrow \mathbb{E}\left[\exp\left(-\alpha X\right)\right] = \exp\left(-\alpha\eta_s\right) \\
&\Rightarrow \log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] = \log\exp\left(-\alpha\eta_s\right) \\
&\Rightarrow \eta_s = -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]
\end{aligned}
$$

Now, from (A.20), the optimal OCE $S_\phi(X)$ for utility function ϕ is:

$$
\begin{aligned}
S_\phi(X) &= -\frac{1}{\alpha}\log\mathbb{E}[\exp\left(-\alpha X\right)] + \mathbb{E}\left[\frac{1}{\alpha}(1-\exp\left(\alpha\eta_s - \alpha X\right))\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \mathbb{E}\left[\frac{1}{\alpha}\right] - \frac{1}{\alpha}\mathbb{E}\left[\exp\left(\alpha\eta_s\right)\exp\left(-\alpha X\right)\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \frac{1}{\alpha} - \frac{1}{\alpha}\exp\left(\alpha\eta_s\right)\mathbb{E}\left[\exp\left(-\alpha X\right)\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \frac{1}{\alpha} - \frac{1}{\alpha}\exp\left(-\alpha\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]\right)\mathbb{E}\left[\exp\left(-\alpha X\right)\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \frac{1}{\alpha} - \frac{1}{\alpha}\frac{1}{\exp\left(\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]\right)}\mathbb{E}\left[\exp\left(-\alpha X\right)\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \frac{1}{\alpha} - \frac{1}{\alpha}\frac{1}{\mathbb{E}\left[\exp\left(-\alpha X\right)\right]}\mathbb{E}\left[\exp\left(-\alpha X\right)\right] \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right] + \frac{1}{\alpha} - \frac{1}{\alpha} \\
&= -\frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]
\end{aligned}
$$

Ben-Tal et al. [101] shows that, the negative of the OCE is a convex risk measure. The particular utility function $\phi(t)$ that we used here yields the convex entropic risk measure:

$$
\rho^{\alpha}_{ent} = \frac{1}{\alpha}\log\mathbb{E}\left[\exp\left(-\alpha X\right)\right]
$$

□

Proof of Corollary 6.1.5

Proof. Proposition A.4.1 shows that entropic risk measure is an OCE with utility function $\phi(t) = \frac{1}{\alpha}\big(1 - \exp\left(-\alpha t\right)\big)$. The deviation bound for any OCE follows from Theorem 6.1.4.

$$\begin{aligned} P(|\hat{\rho}^{\alpha}_{ent}(X_1, \ldots, X_N) - \rho^{\alpha}_{ent}(X)| \geq \varepsilon) &\leq 2\exp\Big(-2\big(\varepsilon/\phi(U)\big)^2 N\Big) \\ &= 2\exp\left(-2\Big(\varepsilon/\frac{1}{\alpha}\big(1 - \exp\left(-\alpha U\right)\big)\Big)^2 N\right) \\ &= 2\exp\Big(-2\alpha^2\varepsilon^2\Big(\exp\left(\alpha U\right)/\exp\left(\alpha U\right) - 1\Big)^2 N\Big) \\ &\overset{\text{(a)}}{\leq} 2\exp\left(-2\alpha^2\varepsilon^2 N\right) \end{aligned}$$

Here (a) follows because $\frac{\exp(\alpha U)}{\exp(\alpha U)-1} \geq 1$ when $U \geq 0$. □

A.4.2 RASR Bellman Update

In this section, we first derive some auxiliary results needed to prove the contraction property of RASR Bellman operator.

Definition A.4.1. (Translation subvariance) For any function $v : \mathcal{S} \to$, a scalar $c \in \mathbb{R}$ and $\gamma \in (0, 1)$, an operator T satisfies the translation subvariance property if

$$\begin{aligned} \big(T(v + c)\big)(s) &= (T(s) + \gamma c \\ &\leq (Tv)(s) + c \end{aligned}$$

Proposition A.4.2. *Operator $T = \log(\cdot)$ is translation subvariant for $c \geq 1$ and $\gamma \in [0, 1)$.*

Proof.

$$\log(X + c) \leq \log(X) + \log(c) \tag{A.21}$$

Here (A.21) follows from Jensen's inequality. From definition A.4.1, for translation subvariance to hold we need:

$$\log(c) = \gamma c$$

and therefore, we have:

$$\gamma = \frac{\log(c)}{c}$$

which satisfies $0 \leq \gamma < 1$ and this completes the proof. □

Lemma A.4.3. *For any two bounded functions $u : \mathcal{S} \to$, $v : \mathcal{S} \to$, and $\gamma \in [0, 1)$, an operator T is a non-expansive mapping if it satisfies monotonicity and translation invariance properties. In particular, it holds for all $u, v \in \mathcal{V}$ that:*

$$\|Tu - Tv\|_\infty \leq \|u - v\|_\infty \ .$$

Proof. Denote

$$c = \max_{s \in S} |u(s) - v(s)| \ .$$

We therefore have, for all $s \in \mathcal{S}$,

$$u(s) - c \leq v(s) \leq u(s) + c \ . \tag{A.22}$$

Applying T on (A.22) and using the monotonicity and translation invariance properties, we obtain for all $s \in \mathcal{S}$,

$$(Tu)(s) - c \leq (Tv)(s) \leq (Tu)(s) + c \ .$$

It therefore follows that for all $s \in \mathcal{S}$,

$$|(Tv)(s)) - (Tu)(s)| \leq c \ ,$$

and as a result, we have:

$$\|Tu - Tv\|_\infty \leq c$$

. □

Lemma A.4.4. *For any two bounded functions $u : \mathcal{S} \to$, $v : \mathcal{S} \to$, and $\gamma \in [0, 1)$, an operator T is a contraction mapping if it satisfies monotonicity and translation subvariance properties. In particular, it holds for all $u, v \in \mathcal{V}$ that:*

$$\|Tu - Tv\|_\infty \leq \gamma \|u - v\|_\infty \ . \tag{A.23}$$

Proof. We first denote a scalar c as:

$$c = \max_{s \in S} |u(s) - v(s)| \ .$$

We therefore have, for all $s \in \mathcal{S}$:

$$u(s) - c \leq v(s) \leq u(s) + c \ . \tag{A.24}$$

Applying T on (A.24) and using the monotonicity and translation subvariance properties, we obtain for all $s \in \mathcal{S}$,

$$(Tu)(s) - \gamma c \leq (Tv)(s) \leq (Tu)(s) + \gamma c \ .$$

It therefore follows that for all $s \in \mathcal{S}$,

$$|(Tv)(s)) - (Tu)(s)| \leq \gamma c \ ,$$

And as a result, we have:

$$\|Tu - Tv\|_\infty \leq \gamma c$$

. □

Proof of Theorem 6.2.2

Proof. The RASR Bellman operator from (6.4) is:

$$
\begin{aligned}
(\mathfrak{T}v_{\hat{P}})(s) &= \max_{a\in\mathcal{A}} \rho^{\alpha}_{P_t^{\omega}\sim\hat{P},s'\sim P_t^{\omega}(\cdot|s_t,a_t)}\Big[r_{s_t,a_t,s'} + \gamma v^{\pi}_{\hat{P}}(s')\Big] \\
&= \max_{a\in\mathcal{A}} \underbrace{\Bigg(\underbrace{-\frac{1}{\alpha}\log\Bigg(\underbrace{\mathbb{E}_{P_t^{\omega}\sim\hat{P},s'\sim P_t^{\omega}(\cdot|s_t,a_t)}\Big[\exp\big(-\alpha(r_{s_t,a_t,s'} + \gamma v^{\pi}_{\hat{P}}(s'))\big)\Big]}_{T_1}\Bigg)}_{T_2}\Bigg)}_{T_3}
\end{aligned}
$$

The Bellman operator $\mathfrak{T}_{\hat{P}}$ is composed of three operators: $T_1 = \mathbb{E}[\cdot]$, $T_2 = \log(\cdot)$ and $T_3 = \max(\cdot)$. All these operators independently satisfy the monotonicity property [3, 17]. Operator T_1 is known to be translation subvariant [3]. Lemma A.4.2 shows that operator T_2 is translation subvariant. And operator T_3 is known to be translation invariant [17]. We then have:

$$
\begin{aligned}
\|\mathfrak{T}u_{\hat{P}} - \mathfrak{T}v_{\hat{P}}\|_{\infty} &= \|T_3T_2T_1u_{\hat{P}} - T_3T_2T_1v_{\hat{P}}\|_{\infty} \\
&\overset{(a)}{\leq} \|T_2T_1u_{\hat{P}} - T_2T_1v_{\hat{P}}\|_{\infty} \\
&\overset{(b)}{\leq} \gamma\|T_1u_{\hat{P}} - T_1v_{\hat{P}}\|_{\infty} \\
&\overset{(c)}{\leq} \gamma^2\|u_{\hat{P}} - v_{\hat{P}}\|_{\infty} \\
&\leq \gamma\|u_{\hat{P}} - v_{\hat{P}}\|_{\infty} \ .
\end{aligned}
$$

Here (a) follows from Lemma A.4.3, (b) and (c) follows from Lemma A.4.4. □

A.4.3 Proof of Theorem 6.2.3

Proof. We compute the gradient $\nabla_{\theta}J(\pi_{\theta})$ of (6.6) with respect to θ as:

$$
\begin{aligned}
&\nabla_\theta J(\pi_\theta) \\
&= -\frac{1}{\alpha}\nabla_\theta \log\left(\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)\right) \\
&= \frac{-\sum_\tau p_\theta(\tau)\nabla_\theta \log p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)}{\alpha\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)} \\
&= \frac{-\sum_\tau p_\theta(\tau)\nabla_\theta \log\Big(p_0(s_0)\prod_{t=0}^T \pi_\theta(a_t|s_t)\hat{P}_t^\omega(s_{t+1}|s_t,a_t)f_t^\omega\Big)\exp\big(-\alpha R(\tau)\big)}{\alpha\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)} \\
&= \frac{-\sum_\tau p_\theta(\tau)\nabla_\theta\Big(\log p_0(s_0)+\sum_{t=0}^T\Big(\log\pi_\theta(a_t|s_t)+\log\hat{P}_t^\omega(s_{t+1}|s_t,a_t)+\log f_t^\omega\Big)\Big)\exp\Big(-\alpha R(\tau)\Big)}{\alpha\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)} \\
&= \frac{-\sum_\tau p_\theta(\tau)\sum_{t=0}^T \nabla_\theta\log\pi_\theta(a_t|s_t)\cdot\exp\Big(-\alpha\sum_{t=0}^T r_{s_t,a_t}\Big)}{\alpha\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)} \\
&= \frac{-\sum_\tau p_\theta(\tau)\sum_{t=0}^T \frac{\nabla_\theta\pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\cdot\exp\Big(-\alpha\sum_{t=0}^T r_{s_t,a_t}\Big)}{\alpha\sum_\tau p_\theta(\tau)\exp\big(-\alpha R(\tau)\big)}
\end{aligned}
$$

□