Foundations and Trends® in Machine Learning

Risk-Sensitive Reinforcement Learning via Policy Gradient Search

Suggested Citation: Prashanth L. A. and Michael C. Fu (2022), "Risk-Sensitive Reinforcement Learning via Policy Gradient Search", Foundations and Trends® in Machine Learning: Vol. 15, No. 5, pp 536–692. DOI: 10.1561/2200000091.

Prashanth L. A.
Department of Computer Science and Engineering,
Indian Institute of Technology Madras
prashla@cse.iitm.ac.in

Michael C. Fu
Robert H. Smith School of Business &
Institute for Systems Research,
University of Maryland, College Park
mfu@umd.edu

now
the essence of knowledge
Boston — Delft

Contents

Risk-Sensitive Reinforcement Learning via Policy Gradient Search

Prashanth L. A.[1] and Michael C. Fu[2]

[1] *Indian Institute of Technology Madras, India; prashla@cse.iitm.ac.in*
[2] *University of Maryland, College Park, USA; mfu@umd.edu*

ABSTRACT

The objective in a traditional reinforcement learning (RL) problem is to find a policy that optimizes the expected value of a performance metric such as the infinite-horizon cumulative discounted or long-run average cost/reward. In practice, optimizing the expected value alone may not be satisfactory, in that it may be desirable to incorporate the notion of risk into the optimization problem formulation, either in the objective or as a constraint. Various risk measures have been proposed in the literature, e.g., exponential utility, variance, percentile performance, chance constraints, value at risk (quantile), conditional value-at-risk, prospect theory and its later enhancement, cumulative prospect theory.

In this monograph, we consider risk-sensitive RL in two settings: one where the goal is to find a policy that optimizes the usual expected value objective while ensuring that a risk constraint is satisfied, and the other where the risk measure is the objective. We survey some of the recent work in this area specifically where policy gradient search is the solution approach. In the first risk-sensitive RL setting, we cover popular risk measures based on variance, conditional value-at-risk, and chance constraints, and present a template for

Prashanth L. A. and Michael C. Fu (2022), "Risk-Sensitive Reinforcement Learning via Policy Gradient Search", Foundations and Trends® in Machine Learning: Vol. 15, No. 5, pp 536–692. DOI: 10.1561/2200000091.

policy gradient-based risk-sensitive RL algorithms using a Lagrangian formulation. For the setting where risk is incorporated directly into the objective function, we consider an exponential utility formulation, cumulative prospect theory, and coherent risk measures. This non-exhaustive survey aims to give a flavor of the challenges involved in solving risk-sensitive RL problems using policy gradient methods, as well as outlining some potential future research directions.

Preface

Reinforcement learning (RL) is one of the foundational pillars of artificial intelligence and machine learning. An important consideration in any optimization or control problem is the notion of risk, but its incorporation into RL has been a fairly recent development. This monograph surveys research on risk-sensitive RL that uses policy gradient search, i.e., policy optimization in a stochastic formulation, as opposed to robust optimization approaches and methods that focus on the value function.

We have tried to make the exposition completely self-contained but also organized in a manner that allows expert readers to skip background sections. In particular, those readers already familiar with Markov decision processes (MDPs), risk measures, and stochastic gradient-based search (specifically, stochastic approximation) can skip Sections 2, 3, and 4, respectively.

We have benefited from the feedback of many who read earlier drafts of the manuscript. We begin by thanking Prof. Vivek Borkar, who generously offered valuable detailed comments regarding the content, and provided material and references for the sections on the exponential cost formulation. Next, we thank Prof. Shalabh Bhatnagar for helpful discussions on the convergence analysis in the risk-as-constraint setting, and Prof. Armand Makowski for critical observations. We'd also like to thank two anonymous reviewers, whose comments and suggestions helped us improve the exposition considerably. Lastly, we thank several of our Ph.D. students — Xingyu Ren, Erfaun Noorani, Mehrdad Moharrami, Nithia Vijayan, Yi Zhou, and Mengting Chao, who read through various portions and stages of the manuscript and caught numerous typos. Any remaining errors are of course our responsibility alone.

One final note: We have chosen to include references at the end of the section in bibliographic remarks rather than cite them in the main text, so as not to interrupt the expositional flow.

1

Introduction

Markov decision processes (MDPs) provide a general framework for modeling a wide range of problems involving sequential decision making under uncertainty, which arise in many areas of applications, such as transportation, computer/communication systems, manufacturing, and supply chain management. MDPs transition from state to state probabilistically over time due to chosen actions taken by the decision maker, incurring state/action-dependent costs/rewards at each instant. The goal is to find a policy (sequence of decision rules) for choosing actions that optimizes a long-run objective function, e.g., the cumulative sum of discounted costs or the long-run average cost.

The traditional MDP setting assumes that (i) the transition dynamics (probabilities) and costs/rewards are fully specified/known, and (ii) the objective function and constraints involve standard expected value criteria. However, in a myriad of settings of practical interest, neither of these conditions holds, i.e., only ***samples*** of transitions (and costs/rewards) can be observed (e.g., in a black-box simulation model or an actual system) and/or performance measures that incorporate ***risk*** really need to be considered in the problem. In the case of the former, reinforcement learning (RL) techniques can be employed, and in the

latter setting, risk-sensitive approaches are appropriate. Although there is abundant research on both of these settings dating back decades, the work combining both aspects is more recent. Furthermore, the two settings have been predominantly pursued independently by different research communities, with RL a focus of CS/AI researchers and risk-sensitive MDPs a focus of stochastic control and operations research/management science/mathematical finance researchers.

Why risk? (Avoid merely expectations?)

The focus of this monograph is not on why risk is important nor on what is the best way to incorporate it into decision making but rather on finding good risk-sensitive policies via RL policy gradient algorithms. However, to provide some motivation for incorporating risk into decision making, we briefly describe two everyday illustrative examples. The first example has to do with financial investments, where the primary objective is generally to *maximize* ***expected*** *return*. Clearly, this is not sufficient for most decision makers, who would very much like to take into consideration the "risk" of the investments, in this case taken to mean mitigating the potential downside losses. The second example is your daily commute to work. In this case, your primary objective is likely to *minimize* ***expected*** *travel time*. However, if you have an important early morning meeting, you might want to reduce the "risk" of being late by choosing an alternative that has a higher expected travel time but is unlikely to suffer a huge delay from an unexpected but rare event such as an overturned tractor-trailer. A colleague of ours avoids taking the highway to/from work for this very reason (along with safety considerations). In other words, most decision makers consider more than merely expectations. Both of these examples also serve to illustrate the more general observation that real-world decisions involve multiple objectives, where at least one of them involves the notion of risk, extending beyond the usual expected value performance measures considered in standard MDP and RL models (including commonly used metrics for analysis purposes such as expected regret in multi-armed bandit models).

Types of risk and ways to incorporate risk

As in any multi-objective optimization problem, there are many ways to incorporate risk. Again, our focus is not on advocating for one formulation over another, but to provide several different alternatives, with a solution approach for each of them. Which formulation is "better" will depend on both the problem and the problem solver(s). We illustrate this concept by revisiting our two examples.

One way to address risk in the investment problem is to minimize some measure of volatility, which could take the form of putting an upper bound on the *variance* of return. Thus, the decision problem becomes a constrained optimization of maximizing the objective of expected return *subject to a constraint* on the variance of return. This is the classic mean-variance portfolio optimization problem in finance for which Harry Markowitz was awarded the 1990 Nobel Prize in Economics.

It can be easily argued that variance is not the best measure of risk for this problem, since it also penalizes excessive upside moves, so maybe focusing on one tail (the downside risk) is more appropriate. One way to address this would be to limit the probability of a high loss to some acceptable level such as 5% or 1% or even smaller. This is known as a *chance constraint*. Conversely, one might have an upper bound on the amount of loss that might occur at a certain low probability, i.e., putting a constraint on a quantile of the loss distribution, which the financial industry defines as *value-at-risk* (VaR). A more sophisticated extension of VaR is *conditional value-at-risk* (CVaR), which also has some other nice properties that VaR does not, most notably that it is a *coherent risk measure*. Exponential utility is another way of capturing risk preferences and implicitly capturing higher moments beyond the second moment. Section 3 provides a more formal review of all of these risk concepts and metrics.

Similarly, revisiting risk in the commuting problem where the objective is to minimize travel time, a constrained optimization problem formulation would be to minimize expected travel time subject to an upper bound on the variability of travel time, or alternatively, one could instead employ a chance constraint by specifying the probability of the travel time exceeding an acceptable threshold, e.g., requiring that at least 99% of the time the travel time will be less than an hour.

Realistic problems may involve multiple constraints that need to be satisfied concurrently, such as bounds on both the variability and the probability of a rare event. In our setting, this can be easily handled, but for the sake of simplicity we will only explicitly consider the case of a single constraint, as the extension using the policy gradient approach would just involve additional Lagrange multiplier gradient estimates, but the general approach would be the same.

Finally, rather than formulating the problem with risk as a constraint, another approach is to try and include it in the objective function. Perhaps the simplest way would be as a weighted combination of the multiple objectives. While we don't address the weighed objectives formulation explicitly, it should be clear how it could also be handled as an easy special case using the techniques of this monograph. Instead, we consider more general formulations: the use of expected utility (an exponential cost formulation), which modifies the output performance measure (corresponding to investment return or travel time in the two examples), and a risk measure called *cumulative prospect theory* (CPT) that "distorts" the perceived probabilities due to the decision maker's view of the world. Demonstrating that prospect theory and CPT are able to model certain aspects of actual observed human behavior that utility theory was unable to capture was a key contribution for which (behavioral psychologist) Daniel Kahneman was awarded the 2002 Nobel Prize in Economics. Our treatment also extends the CPT formulation to a framework encompassing general coherent risk measures.

Objectives of this monograph

The main purpose of this monograph is to introduce and survey research results on policy gradient methods for reinforcement learning with risk-sensitive criteria, as well as to outline some promising avenues for future research following the risk-sensitive RL framework. We consider both constrained formulations where the traditional expected value performance measure is augmented with a risk constraint and problem formulations where the risk measure is explicitly in the objective function being optimized. Some well-known examples of risk measures to be considered as constraints, most of which were illustrated

by the two earlier examples, include variance (or higher moments), probabilities (in the form of chance constraints), quantiles or value-at-risk (VaR), and conditional value-at-risk (CVaR). As also mentioned in the examples, risk measures used explicitly as the objective function include exponential utility and some very recent work on using CPT with RL.

To be specific, the constrained risk-sensitive RL problem will be an optimization problem of the following general form:

$$\min_{\theta \in \Theta} J(\theta) \triangleq \mathbb{E}\left[D(\theta)\right] \text{ subject to } \quad G(\theta) \leq \kappa, \tag{1.1}$$

where θ denotes the policy parameter, Θ represents the policy space, $D(\theta)$ is a (stochastic) cost function, $G(\theta)$ is a risk measure, and κ denotes the acceptable risk level. In the MDP setting, the quantities may also depend on the initial state of the MDP, which is not indicated here. The most common choices for $D(\theta)$ in the MDP setting include the infinite-horizon cumulative discounted cost, total cost in a stochastic shortest path problem, and the long-run average cost. Note that we will be minimizing cost (as in the commuting example), which is more common in MDP formulations than in the RL setting, which often focuses on maximizing reward (as in the investment example). The classic "risk-neutral" formulation simply minimizes $J(\cdot)$ without the risk constraint in (1.1). Also, in contrast to the traditional setting of risk-sensitive control where J and G functions are analytically available in the MDP model, in the RL setting, J and G are unknown or cannot be calculated directly, but noisy estimates of J and G are available, e.g., samples of D could provide an unbiased estimator of J. Thus, as in the usual RL setting, traditional MDP techniques cannot be applied, whereas RL algorithms suitably adapted provide one avenue to attack such risk-sensitive MDPs, i.e., a setting when the MDP model is unknown and all the information about the system is obtained from samples resulting from the decision maker's interaction with the environment.

We propose to solve the constrained optimization problem (1.1) by performing gradient descent search on the Lagrangian objective function. As depicted in Figure 1.1, the risk-sensitive policy gradient algorithm requires estimators $\widehat{\nabla} J(\theta)$, $\widehat{\nabla} G(\theta)$ and $\widehat{G}(\theta)$ of $\nabla J(\theta)$, $\nabla G(\theta)$ and $G(\theta)$, respectively. Then, two-timescale gradient-based search algorithms

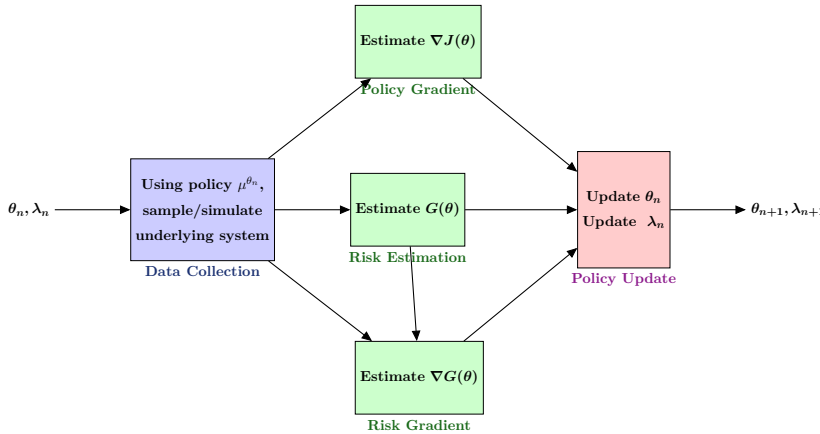


Figure 1.1: Schematic of risk-sensitive policy gradient algorithm for constrained optimization (underlying system could be a simulation model or a real system).

taking the following form will be developed (where λ is the Lagrange multiplier to be optimized along with the policy parameter θ):

$$\lambda_{n+1} = \left[\lambda_n + \zeta_1(n)\left(\widehat{G}(\theta_n) - \kappa\right)\right]^+,$$
$$\theta_{n+1} = \Gamma\left[\theta_n - \zeta_2(n)\left(\widehat{\nabla} J(\theta_n) + \lambda_n \widehat{\nabla} G(\theta_n)\right)\right],$$

where $[x]^+ = \max(0, x)$, Γ is a projection into Θ, and $\{\zeta_1(n), \zeta_2(n)\}$ are step-size sequences selected such that the θ update is on the faster timescale and the λ update is on the slower timescale (see Section 5.2 for details).

In addition to the risk-constrained problem (1.1), we also consider a risk-sensitive problem where the risk measure is explicitly incorporated into the objective function, i.e., the following optimization problem:

$$\min_{\theta \in \Theta} G(\theta), \tag{1.2}$$

where G is a risk objective function involving exponential utility, CPT, or a coherent risk measure. For solving the problem (1.2), we propose a policy gradient algorithm that incorporates the following iterative update:

$$\theta_{n+1} = \Gamma\big[\theta_n - \zeta(n)\widehat{\nabla} G(\theta_n)\big],$$

where $\{\zeta(n)\}$ is a step-size sequence, $\widehat{\nabla} G(\theta_n)$ is an estimate of $\nabla G(\theta_n)$, and Γ is a projection operator that keeps the iterate θ_n bounded within the set Θ as in the case of the risk-constrained policy gradient algorithm above (see Section 5.1 for details).

Challenges in risk-sensitive RL

Risk-sensitive RL is generally more challenging than its risk-neutral counterpart. For instance, for a discounted-cost MDP, there exists a Bellman equation for the variance of the return, but the underlying Bellman operator is not necessarily monotone, so that policy iteration is no longer guaranteed to lead to an optimal policy. Moreover, finding a globally mean-variance optimal policy in a discounted-cost MDP is NP-hard, even in the classic MDP setting where the transition model is known. Average-cost MDP problems also are generally NP-hard, e.g., consider a risk measure that is not the plain variance of the average cost and instead is a variance of a quantity that measures the deviation of the single-stage cost from the average cost. Finally, in comparison to variance/CVaR, CPT is a non-coherent and non-convex measure, ruling out the usual Bellman equation-based dynamic programming (DP) approaches when optimizing the MDP CPT-value.

The computational complexity results summarized in the previous paragraph imply that finding guaranteed global optima of risk-sensitive MDP formulations described by (1.1) or (1.2) is not computationally practical, motivating the need for algorithms that approximately solve such MDP formulations. In this monograph, we focus on policy gradient-type learning algorithms where the policies are parameterized in a continuous space, and an iterative search for a better policy occurs through a gradient-descent update. Actor-critic methods are a popular subclass of policy gradient methods and were among the earliest to be investigated in RL. They are comprised of an *Actor* that improves the current policy via gradient descent (as in policy gradient schemes) and a *Critic* that incorporates feature-based representations to approximate

the value function. The latter approximation is necessary to handle the curse of dimensionality. Regular policy gradient schemes usually rely on Monte Carlo methods for policy evaluation, an approach that suffers from high variance as compared to actor-critic schemes. On the other hand, function approximation introduces a bias in the policy evaluation. A policy gradient/actor-critic scheme with provable convergence to a locally risk-optimal policy would require careful synthesis of techniques from stochastic approximation, stochastic gradient estimation approaches, and importance sampling.

Several of the constituent solution pieces require significant research for various risk measures. For example, consider the "policy evaluation" part of the overall algorithm in a risk-sensitive MDP, which requires estimating $J(\theta)$ and $G(\theta)$, given samples obtained by simulating the MDP with policy θ. If $J(\theta)$ is one of the usual MDP optimization objectives such as discounted total cost, long-run average cost, or total cost (in a finite-horizon MDP), then estimating $J(\theta)$ can be performed using one of the existing algorithms. Temporal difference (TD) learning is a well-known algorithm that can learn the objective value along a sample path for a given θ. However, estimating $G(\theta)$ using TD-type learning algorithms is infeasible in many cases. For instance, consider variance as the risk measure in a discounted-cost MDP. In this case, even though there is a Bellman equation, the operator underlying this equation is not monotone, ruling out a TD-type learning algorithm. More recently, CVaR-constrained MDPs have been considered, though a variance-reduced CVaR estimation algorithm is still needed. In other words, there is no algorithm in an RL context that incorporates a variance reduction technique such as importance sampling and is provably convergent. Note that variance reduction is necessary, because CVaR is based on the tail of the distribution.

Going beyond the prediction problem, designing policy gradient algorithms is challenging for a risk-sensitive MDP, as it requires estimating the (policy) gradient of the risk measure considered, a nontrivial task in the RL context. For instance, in a discounted-cost MDP context, the policy gradient theorem variant that accounts for the variance of the cumulative discounted cost does not lend itself to an RL algorithm. An alternative is to apply a finite differences method such as simultaneous

perturbation stochastic approximation (SPSA), which amounts to treating the MDP as a black box, and such an approach would ignore the underlying Markovian structure of the problem, which is the case with the existing policy gradient algorithms for optimizing the CPT-value in any of the MDP settings.

Outline of the remaining sections

Section 2 provides an overview of MDPs and outlines the standard formulations for discounted-cost and average-cost MDPs and stochastic shortest path total-cost MDP problems. Examples and basic theoretical results are included for the benefit of readers less familiar with MDPs. Section 3 introduces all of the risk measures used in the monograph, namely exponential cost, variance, CVaR, coherent risk measures, chance constraints, and CPT. Section 4 provides an introduction to temporal difference learning and two gradient estimation techniques, namely simultaneous perturbation (stochastic approximation) and the likelihood ratio method. Section 5 presents two templates for risk-sensitive policy gradient algorithms, one for the setting where the risk measure is the objective, and the other for the setting where the risk measure is featured in the constraint. This chapter also presents a convergence analysis of the template algorithms for both settings. Section 6 develops policy gradient algorithms for four special cases of risk-sensitive MDPs for the constrained optimization problem posed in (1.1), with variance, CVaR, and a chance constraint used as the risk measure constraint. Section 7 develops policy gradient algorithms for three risk-sensitive MDP formulations in the unconstrained optimization setting of (1.2) with risk explicitly as the objective: exponential cost, CPT, and coherent risk measures. Finally, Section 8 provides concluding remarks and identifies some interesting future research directions.

A brief note on notation

Throughout the monograph, the functions J, G, and D may show one, two, or no arguments, depending on the context. Specifically, the two possible arguments would be θ, the policy parameter, as in (1.1) or (1.2),

or a state of the MDP (e.g., x_0, x, i, j), as described in Section 2. This is particularly relevant to Sections 5, 6, and 7. The same "convention" is used for other analogous counterparts such as the variance and squared versions of these quantities. On the other hand, dependence on an entire MDP policy μ is represented as subscript, e.g., J_μ. Gradients represented by ∇ are assumed to be with respect to θ unless otherwise indicated, e.g., ∇_λ denoting a gradient with respect to the Lagrange multiplier λ. Finally, all vectors will be assumed to be column vectors, and superscript "T" will be used to denote the matrix/vector transpose operation.

Bibliographic remarks

MDPs have a long history dating back to the work of Richard E. Bellman. For a rigorous introduction, the reader is referred to the books by Puterman (1994) and Bertsekas (2007), and for reinforcement learning, the books by Bertsekas and Tsitsiklis (1996), Sutton and Barto (2018), and Szepesvári (2011). Material in this book drawn from our own research includes Prashanth and Ghavamzadeh (2013), Prashanth and Ghavamzadeh (2016), Prashanth (2014), Prashanth *et al.* (2016), and Gopalan *et al.* (2017). Cumulative prospect theory (CPT) was introduced by Tversky and Kahneman (1992) as a successor to prospect theory, which was one of the central contributions cited for Daniel Kahneman receiving the Nobel Memorial Prize in Economic Sciences in 2002.

References for the various risk measures include the following: mean-variance tradeoff (Markowitz, 1952), exponential utility (Arrow, 1971; Howard and Matheson, 1972), the percentile performance (Filar *et al.*, 1995), the use of chance constraints (Prekopa, 2003), stochastic dominance constraints (Dentcheva and Ruszczynski, 2003), value at risk (VaR), and conditional value-at-risk (CVaR) (Rockafellar and Uryasev, 2000; Ruszczyński, 2010; Shen *et al.*, 2013). The concept of a coherent risk measure was introduced by Artzner *et al.* (1999), see also Föllmer and Schied (2004), with the extension to multi-period settings treated in Riedel (2004), Ruszczyński and Shapiro (2006), Ruszczyński (2010), Cavus and Ruszczynski (2014), Tallec (2007), and Choi (2009).

The large body of literature utilizing the exponential utility formulation includes the classic formulation by Howard and Matheson (1972); related work includes Whittle (1990), Browne (1995), Fleming and McEneaney (1995), Hernández-Hernández and Marcus (1996), Marcus *et al.* (1997), Fernández-Gaucherand and Marcus (1997), Hernández-Hernández and Marcus (1999), Coraluppi and Marcus (1999a), Coraluppi and Marcus (1999b), Coraluppi and Marcus (2000), Borkar and Meyn (2002), and Bäuerle and Rieder (2014). For a survey of risk-sensitive RL under the exponential utility formulation, the reader is referred to Borkar (2010).

Another approach to risk/uncertainty is the robust optimization approach. In the setting of Markov decision processes, Iyengar (2005) is an early seminal work in this area, where a robust optimal policy is defined relative to uncertainty in the underlying transition probabilities. We do not pursue the robust approach in this monograph.

The existence of a Bellman equation for the variance of the return, where the underlying Bellman operator is not necessarily monotone, can be found in Sobel (1982). The result that finding a globally mean-variance optimal policy in a discounted-cost MDP is NP-hard can be found in Mannor and Tsitsiklis (2013). The use of variance of a quantity that measures the deviation of the single-stage cost from the average cost can be found in Filar *et al.* (1989). The result that solving an average-cost MDP under this notion of variance is NP-hard is shown in Filar *et al.* (1989).

Actor-critic methods investigated in RL are found in Barto *et al.* (1983) and Sutton (1984). Temporal difference (TD) learning can be found in Sutton (1988). More recently, CVaR-constrained MDPs have been considered in Borkar and Jain (2010), Prashanth (2014), and Tamar *et al.* (2014a), though a variance-reduced CVaR estimation algorithm is still needed.

The application of simultaneous perturbation stochastic approximation (SPSA) to policy gradient search for mean-variance optimization in discounted-cost MDPs is considered in Prashanth and Ghavamzadeh (2016) and for optimizing CPT-value in Prashanth *et al.* (2016).

Prospect theory (PT) was introduced in Kahneman and Tversky (1979), and cumulative prospect theory (CPT) in Tversky and Kahne-

man (1992), with experiments on humans reported in Starmer (2000) and Tversky and Kahneman (1992). More work adopting this approach includes Lin (2013), Lin and Marcus (2013b), Lin and Marcus (2013a), and Lin *et al.* (2018); see also Cavus and Ruszczynski (2014).

Variance as a risk measure in a discounted-cost and average-cost MDP, respectively, are based on Prashanth and Ghavamzadeh (2013) and Prashanth and Ghavamzadeh (2016). CVaR as a risk measure is based on Prashanth (2014). CPT as the risk measure is based on Prashanth *et al.* (2016) and Jie *et al.* (2018).

A sampling but nowhere near exhaustive list of other risk-sensitive RL work includes the following. In Tamar *et al.* (2012), variance as risk is considered in a stochastic shortest path context, and a policy gradient algorithm using the likelihood ratio method is provided. In Mihatsch and Neuneier (2002), a modified temporal differences algorithm is proposed and connected to the exponential utility approach. A general policy gradient algorithm that handles a class of risk measures that includes CVaR is presented in Tamar *et al.* (2015b). An early work that considers a constrained MDP setting similar to that in (1.1) is Borkar (2005), where the objective is average cost and the constraint is also an average-cost function different from the objective function. A modification of this formulation to a discounted-cost MDP, incorporating function approximation, was treated in Bhatnagar (2010). CVaR optimization in a constrained MDP setup was also explored in Borkar and Jain (2010), but the algorithm proposed there requires that the single-stage cost be separable. Optimization of risk measures that include CVaR in an unconstrained MDP setting using RL algorithms with function approximation can be found in Jiang and Powell (2017).

2

Markov Decision Processes

A Markov decision process (MDP) is a discrete-time stochastic process that transitions from one state to the next in a probabilistic fashion after the execution of an action, on the way possibly accumulating costs (or rewards). The objective of an MDP is to minimize some cost function (or maximize some reward function) over the horizon of interest, which may be finite (possibly random) or infinite. In this section, we provide a basic overview of MDP theory for several of the most commonly used settings. A reader familiar with the theory of MDPs can skip this section and move to the description of risk measures and risk-sensitive RL algorithms in the subsequent sections.

We consider an MDP $\{x_t, t = 0, 1, \ldots\}$ with state space $\mathcal{X}$ and action space $\mathcal{A}$ (both assumed to be finite), and starting in state x_0. Let $P(\cdot|x, a)$ denote the state transition probability distribution from state x under action a, $k(x, a)$ denote the single-stage cost incurred in state x under action a, and $\mathcal{A}(x) \subset \mathcal{A}$ denote the set of feasible actions in state x (all assumed to be stationary). The sequence of actions $\{a_t, t = 0, 1, \ldots\}$ follows a policy μ (also assumed to be stationary). For simplicity, we consider nonrandomized policies, i.e., $\mu : \mathcal{X} \longrightarrow \mathcal{A}$. We extend to randomized policies at the end of the section when we introduce the policy parameterization to be considered in Section 5. We introduce MDPs under three different (risk-neutral) objectives/settings: infinite-horizon discounted cost, stochastic shortest path total cost, and infinite-horizon average cost, described in Sections 2.1, 2.2, and 2.3, respectively.

2.1 Discounted-cost MDP

The infinite-horizon cumulative discounted cost for an MDP trajectory (or sample path) under policy μ, starting in state x_0, is given by

$$D_\mu(x_0) = \sum_{m=0}^{\infty} \gamma^m k(x_m, a_m), \tag{2.1}$$

where $\gamma \in [0, 1)$ is the discount factor and $a_m = \mu(x_m)$.

In a risk-neutral setting, the performance measure (or value function) associated with a policy μ is the expected total discounted cost denoted by

$$J_\mu(x_0) \triangleq \mathbb{E}\left[D_\mu\left(x_0\right)\right],$$

and the objective is to find the optimal cost or value function

$$J^* \triangleq \min_{\mu \in \Xi} \{J_\mu\} \tag{2.2}$$

and/or the associated optimal policy $\mu^* \triangleq \arg\min_{\mu \in \Xi} \{J_\mu\}$, where Ξ denotes the set of admissible policies. A policy μ is *admissible* if it considers only feasible actions in any given state, i.e., $\mu(x) \in \mathcal{A}(x)$.

For an $|\mathcal{X}|$-dimensional vector $J \triangleq [J(x)]_{x \in \mathcal{X}}$, define the (Bellman optimality) operator T as follows:

$$(TJ)(x) \triangleq \min_{a \in \mathcal{A}(x)} \left\{ k(x, a) + \gamma \sum_{y \in \mathcal{X}} P(y|x, a) J(y) \right\}, \ \forall x \in \mathcal{X}, \tag{2.3}$$

where $\mathcal{A}(x)$ denotes the set of feasible actions in state x. Next, we define another (Bellman policy) operator T_μ specific to a given policy μ:

$$(T_\mu J)(x) \triangleq k(x, \mu(x)) + \gamma \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) J(y), \ \forall x \in \mathcal{X}. \tag{2.4}$$

The first important result is the following, which forms the basis for the value-iteration algorithm for policy evaluation.

Proposition 2.1. $T_\mu^m J \longrightarrow J_\mu$ as $m \to \infty$, $\forall J \triangleq [J(x)]_{x \in \mathcal{X}}$, where $J_\mu \triangleq [J_\mu(x)]_{x \in \mathcal{X}}$ is the unique solution of

$$J_\mu = T_\mu J_\mu. \tag{2.5}$$

Value iteration relies on the result in Proposition 2.1, as it repeatedly applies the T_μ operator for the expected cost J_μ defined by (2.4), starting with an arbitrary J_0, i.e., the following update rule for computing J_μ:

$$J_{k+1}(x) = k(x, \mu(x)) + \gamma \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) J_k(y), \ \forall x \in \mathcal{X}.$$

Important properties of the Bellman optimality operator T and its relationship to the optimal value function J^* are summarized here.

Proposition 2.2.
(i) $J^* \triangleq [J^*(x)]_{x \in \mathcal{X}}$ is the unique solution to $J^* = TJ^*$.
(ii) For any $J \triangleq [J(x)]_{x \in \mathcal{X}}$, $T^m J \longrightarrow J^*$ as $m \to \infty$,
(iii) A policy μ is optimal if and only if $T_\mu J^* = TJ^*$.

The fixed-point equation in (i) is referred to as the Bellman (optimality) equation, and the corresponding value iteration update for computing the optimal value J^* is the following:

$$J^*_{k+1}(x) = \min_{a \in \mathcal{A}(x)} \left(k(x, a) + \gamma \sum_{y \in \mathcal{X}} P(y|x, a) J^*_k(y) \right), \ \forall x \in \mathcal{X}.$$

We now present a simple illustrative example where the optimal policy is found using the Bellman equation.

Example 2.1. ***(machine replacement problem)***
A machine can be in any of n states, denoted $1, 2, \ldots, n$, where state 1 corresponds to a machine in perfect condition, and higher-valued states indicate deteriorating conditions (e.g., age of the machine), for which it is more costly to operate the machine. The goal is to decide when to replace the machine to minimize the long-run discounted operating cost.

Let $k(i)$ denote the cost of operating the machine in state i in any time period (in this case with no direct dependence on the action), with

$$k(1) \leq k(2) \ldots \leq k(n).$$

The machine deteriorates stochastically according to transition matrix $\mathbf{P} = [p_{ij}]$, where p_{ij} is the probability that the machine goes from

state i to j. Assume the machine can never improve its state without intervention, which implies $p_{ij} = 0$ for $j < i$. The state of the machine at the beginning of each time period is known, and there are just two possible actions: (i) do nothing, or (ii) replace the machine with a new machine (state 1) at cost R. The problem is to choose the actions to minimize the infinite-horizon discounted cost. Intuitively, it makes little sense to replace the machine when it is almost new, i.e., the state is close to 1, and it turns out that a policy that uses a threshold to determine whether to replace or not is optimal.

For a machine in state i, if the action chosen is to replace the machine, then the machine will go to state 1 for the next period, incurring immediate cost $R + k(1)$ and a future discounted (optimal) expected cost from state 1; otherwise, it deteriorates to state $j \geq i$ according to the state transition probabilities p_{ij}, incurring immediate cost $k(i)$ and a corresponding future discounted (optimal) expected cost. Putting these together for the RHS of (2.3), the Bellman equation is obtained by applying Proposition 2.2(i):

$$J^*(i) = \min \left\{ R + k(1) + \gamma J^*(1), k(i) + \gamma \sum_{j=i}^{n} p_{ij} J^*(j) \right\}.$$

Letting $\mathbf{P}_{i,:}$ denote the ith row of $\mathbf{P}$, so $\sum_{j=1}^{n} p_{ij} J^*(j) = \mathbf{P}_{i,:} \cdot J^*$, the optimal action is to replace a machine in state i if

$$R + k(1) + \gamma J^*(1) \leq k(i) + \gamma \mathbf{P}_{i,:} \cdot J^*,$$

and do nothing otherwise.

To establish that the optimal policy is threshold-based requires the following additional assumption (besides the previous assumption that the machine cannot get better on its own):

$$p_{ij} \leq p_{(i+1)j}, \ i < j,$$

which implies that the machine is more likely to jump to a worse state from a state that is closer to it than from a state that is farther away. For J satisfying $J(1) \leq J(2) \leq \ldots \leq J(n)$, and $\forall i \in \{1, 2, \ldots, n-1\}$,

$$\mathbf{P}_{i,:} \cdot J \leq \mathbf{P}_{i+1,:} \cdot J \implies k(i) + \gamma \mathbf{P}_{i,:} \cdot J \leq k(i+1) + \gamma \mathbf{P}_{i+1,:} \cdot J,$$

where the RHS inequality used the assumption that $k(i)$ is nondecreasing. Thus, $(TJ)(i)$ is nondecreasing in i if J is nondecreasing. Using this observation together with the fact that T is a monotone operator, $(T^m J)(i)$ is nondecreasing in i for any $m \geq 1$, which implies that $J^*(i) = \lim_{m\to\infty}(T^m J)(i)$ is nondecreasing in i.

The foregoing implies that the function $h(i) = k(i) + \gamma \mathbf{P}_{i,:} \cdot J^*$ is nondecreasing in i. Let i^* denote the smallest state satisfying $R + k(1) + \gamma J^*(1) \leq k(i) + \gamma \mathbf{P}_{i,:} \cdot J^*$. As illustrated in Figure 2.1, it is easy to infer that the following policy is optimal for the machine replacement problem: replace if $i \geq i^*$; else, do nothing.

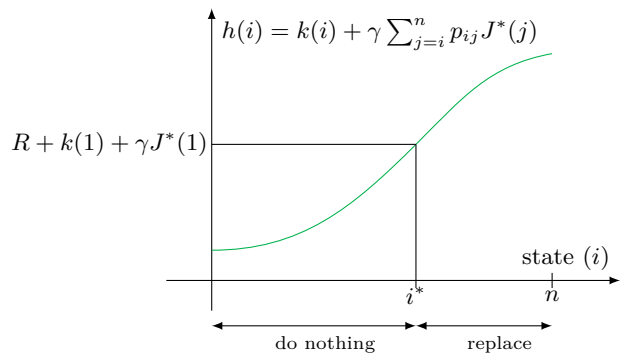


Figure 2.1: The optimal threshold-based policy for the machine replacement problem (For ease of illustration, $h(i)$ is plotted as a continuous function.).

2.2 Stochastic shortest path MDP

In a stochastic shortest path (SSP) problem, the horizon is finite but unknown (random), as the MDP terminates when it enters a predetermined state (or possibly set of states in a generalized setting). The basic version of a shortest path problem is to traverse from a source (origin) node to a sink (destination) node, hence the term stochastic shortest path problem.

An *episode* is a sample path using a policy μ that starts in state $x_0 \in \mathcal{X}$, visits $\{x_1, \ldots, x_{\tau-1}\}$ before ending in the (cost-free) *absorbing* state $0 \in \mathcal{X}$, where τ is the first passage time to state $0 \in \mathcal{X}$. Let $D_\mu(x_0) = \sum_{m=0}^{\tau-1} k(x_m, a_m)$ denote the total cost from an episode, with

the actions $\{a_m\}$ chosen according to policy μ, i.e., $a_m = \mu(x_m)$. In a risk-neutral setting, the performance measure (or value function) associated with a policy μ is

$$J_\mu(x_0) \triangleq \mathbb{E}\left[D_\mu\left(x_0\right)\right].$$

We consider policies that ensure that the absorbing state 0 is recurrent and the remaining states transient in the underlying Markov chain. Such policies are referred to as "proper", and we formalize this notion in the following definition:

Definition 2.1. A stationary policy μ is *proper* if $\forall x \in \mathcal{X}$, there exists $M > 0$ s.t.

$$\rho_\mu = \max_{x \in \mathcal{X}} \mathbb{P}\left(x_M \neq 0 \mid x_0 = x, \mu\right) < 1.$$

The condition in Definition 2.1 ensures that there exists a path of positive probability from any state to the absorbing state 0. To understand the need for the notion of a proper policy, consider a three-state deterministic shortest path example shown in Figure 2.2, where the costs (rather than transition probabilities) are shown on the arcs.

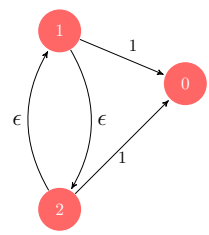


Figure 2.2: Three-state deterministic shortest path example (costs on the arcs).

It is obvious that the least-cost path from node 1 (or 2) to 0 follows the edge connecting it to 0 in one step, with a total cost of 1. However, one improper policy is a policy that leads to a path that loops between 1 and 2. Since the edge costs are positive, it is clear that such an improper policy would incur infinite cost. On the other hand, having zero cost edges between 1 and 2 (i.e., letting $\epsilon \to 0$) would mean the improper policy would incur zero cost in the long run – one type

of pathological scenario that we would like to avoid, motivating the following assumptions for the analysis of SSPs:

A2.1. There exists at least one proper policy.

A2.2. For every improper policy μ, the associated cost $J_\mu(x_0)$ is infinite for at least one state x_0.

The risk-neutral objective in an SSP context is to minimize the expected total cost, i.e.,

$$\min_{\mu \in \Xi} \{J_\mu(x_0)\}, \tag{2.6}$$

where Ξ denotes the set of admissible policies that satisfy A2.1 and A2.2.

As in the discounted-cost setting, for an $|\mathcal{X}|$-dimensional vector $J \triangleq [J(x)]_{x \in \mathcal{X}}$, define the (Bellman optimality) operator T:

$$(TJ)(x) \triangleq \min_{a \in \mathcal{A}(x)} \left\{ k(x,a) + \sum_{y \in \mathcal{X}} P(y|x,a) J(y) \right\}, \ \forall x \in \mathcal{X}, \tag{2.7}$$

and also the operator specific to a given policy μ, T_μ, defined as

$$(T_\mu J)(x) \triangleq k(x, \mu(x)) + \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) J(y), \ \forall x \in \mathcal{X}. \tag{2.8}$$

Now, we state a few important properties of the T_μ operator in the proposition below.

Proposition 2.3. Assume A2.1 and A2.2. Then, for any proper policy μ and any $J \triangleq [J(x)]_{x \in \mathcal{X}}$,

$$\lim_{m \to \infty} (T_\mu^m J)(x) = J_\mu(x), \ \forall x \in \mathcal{X}, \tag{2.9}$$

where $J_\mu \triangleq [J_\mu(x)]_{x \in \mathcal{X}}$ is the unique solution of the fixed-point equation

$$J_\mu = T_\mu J_\mu.$$

As in the discounted case, the convergence result given by (2.9) can be used to establish convergence of the value iteration algorithm, which

repeatedly applies the T_μ operator given by (2.8), starting with an arbitrary J_0.

We now turn our attention to the Bellman optimality operator T, and state a few important properties concerning this operator. First we define the optimal cost (value) function:

$$J^*(x_0) \triangleq \min_{\mu \in \Xi} J_\mu(x_0), \forall x_0 \in \mathcal{X}.$$

Proposition 2.4. Assume A2.1 and A2.2. Then,

1. $J^* \triangleq [J^*(x)]_{x \in \mathcal{X}}$ is the unique solution to the fixed-point equation

$$J^* = TJ^*. \tag{2.10}$$

2. For any $J \triangleq [J(x)]_{x \in \mathcal{X}}$,

$$\lim_{m \to \infty} (T^m J)(x) = J^*(x), \ \forall x \in \mathcal{X}.$$

3. A stationary policy μ is optimal if and only if

$$T_\mu J^* = TJ^*.$$

We illustrate the usage of Bellman equation (2.10) for finding the optimal policy in the example below.

Example 2.2. ***(the spider and the fly)***
A spider hunts a fly on the one-dimensional line of integers $..., -1, 0, +1, ...$ In each period/stage, the fly jumps forward or backward 1 unit with probability p and remains in the same position with probability $1 - 2p$. The spider jumps (1 unit) towards the fly if the distance between them is greater than 1 unit. If the distance between them is exactly 1 unit, the spider can choose to stay in its position hoping the fly will come to it or go 1 unit forward. The game (and the fly) ends when the spider is in the same position as the fly. The goal is to decide the actions of the spider to minimize the expected time to catch the fly.

This problem can be formulated as an SSP, with the state as the distance between the spider and the fly. The terminal state is state

0, which is reached when the spider and fly are in the same position. Note that given an initial distance between the spider and the fly, the subsequent distance between them can never be greater than this distance, so that the number of states is finite. Specifically, assuming that the spider and fly are initially at a separation of n, the state space is $\mathcal{X} = \{0, 1 \dots n\}$. The transition probabilities are obtained as follows: When the state is 2 or more, the spider has to jump towards the fly, leading to

$$p_{i,i-2} = p, \; p_{i,i-1} = 1 - 2p, \text{ and } p_{i,i} = p, \text{ for } i \geq 2,$$

where $p_{i,j}$ denotes the probability that the system transitions from state i to j, when the spider jumps, for $i \geq 2$. Since the optimal action for the spider is to jump in this situation, we drop the dependence on the spider's action in the transition probabilities for $i \geq 2$.

When the state is 1, the spider has two possible actions. Denoting M and $\overline{M}$ as the actions "move" and "don't move", respectively, the transition probabilities as a function of the spider's action are given by

$$\begin{aligned} &p_{1,1}(M) = 2p, \quad p_{1,0}(M) = 1 - 2p, \text{and} \\ &p_{1,2}(\overline{M}) = p, \; p_{1,0}(\overline{M}) = p, \text{ and } p_{1,1}(\overline{M}) = 1 - 2p. \end{aligned}$$

To find the minimum expected time to catch the fly, we set all the single-stage costs to 1.

We now derive the Bellman equation (2.10) for this problem by finding the RHS of (2.7) for each state. As there is only one action to take for states $i > 1$, the Bellman equation for $i > 1$ is

$$\begin{aligned} J^*(i) &= 1 + p_{i,i-2} J^*(i-2) + p_{i,i-1} J^*(i-1) + p_{i,i} J^*(i) \\ &= 1 + pJ^*(i-2) + (1-2p) J^*(i-1) + pJ^*(i). \end{aligned}$$

For state 1, we have to take the minimum over two actions, leading to

$$J^*(1) = 1 + \min\left(2pJ^*(1), \; pJ^*(2) + (1-2p)J^*(1)\right).$$

Therefore, we have

$$J^*(i) = \begin{cases} 1 + pJ^*(i-2) + (1-2p)J^*(i-1) + pJ^*(i) & i \geq 2, \\ 1 + \min(2pJ^*(1), \; pJ^*(2) + (1-2p)J^*(1)) & i = 1, \\ 0 \;\; \text{(cost-free absorbing state)} & i = 0. \end{cases}$$

Straightforward calculations involving $J^*(2)$ and $J^*(1)$ lead to the spider's optimal action in state 1 being to move if the probability of the fly jumping is under 1/3, leading to the spider's overall optimal policy (for $p = 1/3$, either action is optimal in state $i = 1$):

$$\mu(i) = \begin{cases} M & i = 1 \text{ and } p < \frac{1}{3}, \\ \overline{M} & i = 1 \text{ and } p > \frac{1}{3}, \\ M & i \geq 2. \end{cases}$$

2.3 Average-cost MDP

The average cost under policy μ is defined as

$$J_\mu(x) \triangleq \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{m=0}^{T-1} k(x_m, a_m)\middle| x_0 = x\right], \tag{2.11}$$

where $a_m = \mu(x_m)$.

Under the following unichain assumption, the infinite-horizon average cost in (2.11) is identical for all initial states $x_0 \in \mathcal{X}$:

A2.3. The Markov chain generated by any policy μ is irreducible and positive recurrent.

Under A2.3, we can drop the dependence in (2.11) on the initial state and simply use J_μ to denote a ***scalar*** rather than a vector as in the two previous settings.

The goal in the standard (risk-neutral) average-cost formulation is

$$\min_{\mu\in\Xi} J_\mu,$$

where, as before, Ξ denotes the set of admissible policies.

For any policy μ, we associate an expected differential value function defined as follows:

$$Q_\mu(x,a) \triangleq \sum_{m=0}^{\infty} \mathbb{E}[k(x_m, a_m) - J_\mu \mid x_0 = x, a_0 = a, \mu], \tag{2.12}$$

$$V_\mu(x) \triangleq Q_\mu(x, \mu(x)), \tag{2.13}$$

where $V_\mu(x)$ is the expected sum of the differences between the single-stage cost and the average cost J_μ under the policy μ with initial state $x \in \mathcal{X}$, and $Q_\mu(x,a)$ has an analogous connotation. V_μ and Q_μ are referred to as the differential value and Q-value functions, respectively.

Proposition 2.5. Assume A2.3. The differential value and Q-value functions satisfy the following Poisson equations:

$$J_\mu + V_\mu(x) = k(x, \mu(x)) + \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) V_\mu(y), \tag{2.14}$$
$$J_\mu + Q_\mu(x, a) = k(x, a) + \sum_{y \in \mathcal{X}} P(y|x, a) V_\mu(y).$$

A well-known algorithm for computing the differential value function of a given policy using the Poisson equation, is 'relative value iteration', which employs the following update rule:

$$V_{m+1}(x) = k(x, \mu(x)) + \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) V_m(y) - V_m(x_f), \ \forall x \in \mathcal{X},$$

where x_f is a fixed state. V_m converges asymptotically to the differential value function V_μ, although the proof is more delicate than in the discounted-cost or SSP settings, where the T_μ operator is a contraction, which is not the case for the average-cost setting. In addition to computing the differential value, relative value iteration can be used to obtain the average cost J_μ, since the term $V_m(x_f)$ converges to J_μ.

We now turn our attention to the optimal average cost, and its associated Bellman (optimality) equation.

Proposition 2.6. Assume A2.3. Let $J^* = \min_{\mu \in \Xi} J_\mu$ denote the optimal average cost. Let μ^* denote the optimal policy, and let $V^* = V_{\mu^*}$ denote the differential value function associated with the optimal policy μ^*. Then, $(J^*, V^*(x)), x \in \mathcal{X}$, satisfy the following Bellman equation:

$$J^* + V^*(x) = \min_{a \in \mathcal{A}(x)} \left\{ k(x, a) + \sum_{y \in \mathcal{X}} P(y|x, a) V^*(y) \right\}, \forall x \in \mathcal{X}.$$

The 'relative value iteration' for computing the optimal average cost employs the following update rule:

$$V^*_{m+1}(x) = \min_{a \in \mathcal{A}(x)} \left(k(x, a) + \sum_{y \in \mathcal{X}} P(y|x, a) V^*_m(y) \right) - V^*_m(x_f), \ \forall x \in \mathcal{X},$$

where x_f is a fixed state. Again, V_m converges asymptotically to the optimal differential value function V^*, and $V_m(x_f)$ converges to J^*.

Example 2.3. ***(machine replacement problem revisited)*** Consider the machine replacement example again, with the average-cost objective in place of discounted cost. Using the same notation, the Bellman equation for the average-cost problem is given by

$$J^* + V^*(i) = \min\left\{R + k(1) + V^*(1), k(i) + \sum_{j=1}^{n} p_{ij} V^*(j)\right\}, i = 1, \ldots, n,$$

where the action that minimizes the RHS of the Bellman equation is the optimal action in state i.

What is needed to ensure that the Bellman equation is solvable for the average-cost version of this problem? First, observe that not all polices necessarily satisfy the unichain assumption now, i.e., there exists policies that do not satisfy A2.3. To see this, consider a policy that replaces the machine in all states $i < n$, and in state n does nothing. Such a policy clearly results in two disjoint recurrent classes, and hence the underlying Markov chain is not unichain. A more general assumption that ensures solvability of the Bellman equation is the following:

A2.4. The state space can be partitioned into two disjoint classes $\mathcal{X}_1$ and $\mathcal{X}_2$ such that (i) all states in $\mathcal{X}_1$ are transient in the Markov chain generated by any policy μ; and (ii) there exists a policy, say $\tilde{\mu}$, and a positive integer M such that $\mathbb{P}(x_M = x' | x_0 = x, \tilde{\mu}) > 0, \ \ \forall x, x' \in \mathcal{X}_2$.

This assumption is an intuitive extension of the unichain assumption, allowing the presence of transient states, whereby the MDP eventually acts as if it were unichain once it leaves the set of transient states.

Under the assumption above, the average cost J is identical for all initial states, as in the unichain setting. Moreover, it can be easily verified that the machine replacement problem satisfies A2.4, implying the Bellman equation is solvable. Furthermore, the arguments used for discounted-cost MDPs to show that the form of the optimal policy is threshold-based can be extended to the average-cost setting, as well, and we omit the details.

2.4 Randomized policies and policy parameterization

Nonrandomized policies μ specify a single action for a given state, i.e., $\mu(x),\ x \in \mathcal{X}$, whereas a randomized policy μ specifies a *probability distribution* over the feasible action space, i.e., $\mu(\cdot|x)$ will be used to denote a distribution over $\mathcal{A}(x)$. For the discrete state/action space setting, $\mu(a|x)$ is simply the probability of taking action a in state x, and a randomized policy μ is admissible if the policy puts nonzero probability on only feasible actions in any given state, i.e., $\mu(a|x) > 0$ implies $a \in \mathcal{A}(x)\ \forall x, a$. Various definitions would be adjusted accordingly, by summing over the action space. For instance, for the discounted-cost setting, the T_μ operator given by (2.4) becomes

$$(T_\mu J)(x) \triangleq \sum_{a \in \mathcal{A}(x)} \mu(a|x) \Big[k(x,a) + \gamma \sum_{y \in \mathcal{X}} P(y|x,a) J(y) \Big],\ \forall x \in \mathcal{X},$$

and for the average-cost setting, Equations (2.13) and (2.14) become

$$V_\mu(x) \triangleq \sum_{a \in \mathcal{A}(x)} \mu(a|x) Q_\mu(x,a)$$

and

$$J_\mu + V_\mu(x) = \sum_{a \in \mathcal{A}(x)} \mu(a|x) \Big[k(x,a) + \sum_{y \in \mathcal{X}} P(y|x,a) V_\mu(y) \Big],$$

respectively.

Our focus in policy gradient risk-sensitive RL will be on parameterized randomized policies μ^θ, where θ denotes the policy parameter, which appears in the distribution. For example, in the machine replacement problem, a randomized policy would specify the probability of replacing the machine (or doing nothing), and in 'the spider and the fly' example, a randomized policy would specify the probability of moving (or not moving) in each state, e.g., specified as follows:

$$\mu^\theta(M|i) = \theta^i, \quad \mu^\theta(\overline{M}|i) = 1 - \theta^i, \quad \theta = [\theta^0\ \theta^1\ \cdots\ \theta^n],$$

where θ^i denotes the probability of the spider moving when in state i. Policy gradient methods require estimators for $\nabla J(\theta)$, the gradient of the objective function (i.e., total cost or average cost) with respect to the policy parameters.

2.5 Bibliographic remarks

MDPs have a long history dating back to the seminal work of Richard E. Bellman in the 1950s (Bellman, 1957). For a more complete treatment, the reader may refer to the books by Derman (1970), Ross (1983), Puterman (1994), Bertsekas (2007), Bertsekas (2012), and Sutton and Barto (2018), where the last text focuses on reinforcement learning. We assume finite state and action spaces throughout, as the infinite, especially uncountable, space setting involves deeper mathematics beyond the scope of this work; a comprehensive review in the average-cost setting is Arapostathis *et al.* (1993). The presentation of the discounted-cost MDP in Section 2.1 is based on material from Chapter 1 of Bertsekas (2012). In Section 2.2, the notion of a proper policy is based on the treatment of SSPs from Bertsekas (2012, Chapter 2). The latter chapter is also the source for the presentation of Bellman operator and 'the spider and the fly' example in Section 2.2. For the Poisson and Bellman equations in the average-cost MDP from Section 2.3, the reader can refer to either Puterman (1994) or Bertsekas (2012, Chapter 4). The unichain Assumption A2.3, as well as its relaxation in Assumption A2.4, is also based on material from Chapter 4 of Bertsekas (2012). For simulation-based approaches, see Chang *et al.* (2007), Chang *et al.* (2013), and Gosavi (2003).

3

Risk Measures

In this section, we introduce risk measures that can be incorporated into risk-sensitive RL, whether explicitly in the objective function or implicitly as a constraint. The first risk measure involves a modified functional form of exponential utility serving as the objective function in an average-cost MDP, which we will refer to throughout as exponential cost. This measure has been analyzed in depth in the risk-sensitive stochastic control community, but there remains a dearth of computationally practical policy gradient algorithms for high-dimensional state-space RL settings. The second and third risk measures are based on two different interpretations of variance in discounted-cost and average-cost MDPs, described in Sections 3.2 and 3.3, respectively. The fourth risk measure, described in Section 3.4, is conditional Value-at-Risk (CVaR), widely used in finance. The fifth risk measure, detailed in Section 3.5, are chance constraints commonly used in stochastic optimization problem formulations, as briefly mentioned in the commuting example of Section 1. The sixth risk measure, described in Section 3.6, is the class of coherent risk measures, which includes CVaR. The final risk measure, described in Section 3.7, is based on cumulative prospect theory (CPT), which has been found to model human decision making well.

3.1 Exponential cost in average-cost MDPs

As mentioned in Section 1, an exponential utility function is one way commonly used to capture risk preferences. For average-cost MDPs, the corresponding risk measure takes the following form (referred to henceforth as ***exponential cost***):

$$G_\mu \triangleq \limsup_{T\to\infty} \frac{1}{T}\frac{1}{\beta} \log \mathbb{E}\left[\exp\left(\beta \sum_{n=0}^{T-1} k(x_n, a_n)\right)\right],$$

where β is a parameter that controls risk sensitivity. Assuming that the single-stage costs are positive, i.e., $k(\cdot,\cdot) > 0$, $\beta > 0$ corresponds to the risk-averse setting and $\beta < 0$ to the risk-seeking setting, and in the limit $\beta \to 0$, the exponential cost approaches the classic (risk-neutral) average cost. If the Markov chain generated by the policy μ is irreducible and positive recurrent, i.e., satisfies A2.3, then the lim sup in the definition (3.1) can be replaced by an ordinary limit.

3.2 Variance in discounted-cost MDPs

For discounted MDPs with starting state x_0, we consider a *variability* risk measure defined as follows:

$$G_\mu(x_0) \triangleq \mathbf{Var}\left[D_\mu(x_0)\right] = U_\mu(x_0) - J_\mu(x_0)^2, \qquad (3.1)$$
$$\text{where } J_\mu(x) \triangleq \mathbb{E}\left[D_\mu(x)\right] \text{ and } U_\mu(x) \triangleq \mathbb{E}\left[\left(D_\mu(x)\right)^2\right].$$

This risk measure is the *overall* variance of the cumulative discounted cost. An alternative is to consider *per-period* variance, i.e., the deviations of the single-stage costs, which we will consider for the average-cost MDP in Section 3.3. Setting aside the question of which is the most appropriate notion of variability for the underlying MDP, we shall design algorithms for overall variance in the discounted-cost case and per-period variance in the average-cost case in Sections 6.1–6.2, although the constituent pieces of these algorithms can also be easily incorporated

to handle the cases of per-period variance in a discounted cost MDP and overall variance in an average cost MDP.

The variance risk measure $G(x)$ defined by (3.1) satisfies the following fixed-point equation for any $x \in \mathcal{X}$:

$$G_\mu(x) = \chi_\mu(x) + \gamma^2 \sum_{y \in \mathcal{X}} P(y|x, \mu(x)) G_\mu(y), \tag{3.2}$$

where
$\chi_\mu(x) = \gamma^2 \left(\sum_{y \in \mathcal{X}} P(y|x, \mu(x)) J_\mu(y)^2 - \left(\sum_{y \in \mathcal{X}} P(y|x, \mu(x)) J_\mu(y) \right)^2 \right)$.

Remark 3.1. The exponential cost risk measure implicitly incorporates the mean-variance trade-off. To see this, let $D = \sum_{n=0}^{T-1} k(x_n, a_n)$ denote the total cost r.v. Using a Taylor's series expansion, we have $\frac{1}{\beta} \log \mathbb{E}[e^{\beta D(\theta)}] = \mathbb{E}[D(\theta)] + \frac{\beta}{2} \mathrm{Var}[D(\theta)] + O(\beta^2)$; hence, the exponential cost risk measure incorporates all higher moments of D. On the other hand, a mean-variance constrained optimization formulation would involve a variance constraint in (1.1), while minimizing the usual expected cost objective function. Each formulation has its advantages and disadvantages. For example, the constraint formulation eschews choosing the risk parameter β, although its value can be imputed from the choice of the constraint threshold κ in (1.1), where κ may have a more intuitive meaning in many practical applications. However, the fixed-point relation in (3.2) lacks monotonicity, ruling out policy iteration as a candidate for optimizing variance, whereas the exponential cost satisfies a multiplicative form of Bellman equation (see Section 7.1), making it more amenable to dynamic programming algorithms.

3.3 Variance in average-cost MDPs

For average-cost MDPs, we consider the variance defined by deviations of single-stage cost from the average cost (as opposed to the variance of the average cost itself), viz.:

$$G_\mu \triangleq \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{n=0}^{T-1} \left(k(x_n, a_n) - J_\mu\right)^2\right], \tag{3.3}$$

where the actions a_n are governed by policy μ, and J_μ is the average cost of the policy μ. To see the rationale behind the definition above for variability, consider two stream of cost: a policy μ^1 results in $(0, 0, 0, 0, \ldots)$, while another policy μ^2 gives $(100, -100, 100, -100, \ldots)$. The average cost as well as the variance of the average cost is zero for both policies. On the other hand, from the point of the variance as defined by (3.3), policy μ^1 is better than μ^2.

A straightforward calculation yields

$$G_\mu = \eta_\mu - J_\mu^2,$$

where $\eta_\mu = \sum_{x,a} \pi(x, a)k(x, a)^2$ is the average squared cost, with $\pi(x, a)$ denoting the stationary distribution of the state-action pair (x, a) under policy μ.

Along the lines of (2.12) and (2.13), the squared-cost counterparts W and U of Q and V are defined as follows:

$$\begin{aligned} W_\mu(x, a) &\triangleq \sum_{n=0}^{\infty} \mathbb{E}\big[k(x_n, a_n)^2 - \eta_\mu \mid x_0 = x, a_0 = a, \mu\big], \\ U_\mu(x) &\triangleq \sum_a \mu(a|x) W_\mu(x, a). \end{aligned}$$

These differential-value and Q-value functions U and W for the square cost, which are defined above, satisfy the Poisson equations given by

$$\begin{aligned} \eta_\mu + U_\mu(x) &= \sum_a \mu(a|x)\big[k(x, a)^2 + \sum_y P(y|x, a)U_\mu(y)\big], \\ \eta_\mu + W_\mu(x, a) &= k(x, a)^2 + \sum_y P(y|x, a)U_\mu(y). \end{aligned}$$

3.4 Conditional Value-at-Risk (CVaR)

For any random variable (r.v.) X, the Value-at-Risk (VaR) at level $\alpha \in (0, 1)$ is defined as

$$\mathrm{VaR}_\alpha(X) \triangleq \inf\left\{\xi \mid \mathbb{P}\left(X \le \xi\right) \ge \alpha\right\},$$

which mathematically is just an α-quantile, since if F is the cumulative distribution function (c.d.f.) of X, VaR is equivalently defined as

$$\mathrm{VaR}_\alpha(X) \triangleq \inf\{\xi \mid F(\xi) \geq \alpha\} = F^{-1}(\alpha).$$

VaR is a commonly used risk measure in the financial industry, where it represents a level of assets needed to cover a potential loss. VaR as a risk measure has several drawbacks, which precludes using standard stochastic optimization methods; most prominently, VaR is not a coherent risk measure (see Section 3.6 for a definition). On the other hand, another closely related risk measure also widely used in the financial industry called CVaR is coherent and thus lends itself to stochastic programming techniques. CVaR is a conditional mean over the tail distribution as delineated by the VaR, defined as follows:

$$\mathrm{CVaR}_\alpha(X) \triangleq \mathbb{E}\left[X | X \geq \mathrm{VaR}_\alpha(X)\right].$$

In a stochastic shortest path problem, CVaR is defined as:

$$G_\mu(x_0) \triangleq \mathrm{CVaR}_\alpha\left[\sum_{m=0}^{\tau-1} k(x_m, a_m) \middle| x_0\right], \tag{3.4}$$

where τ is the first visit time to absorbing state 0 and $a_m \sim \mu(\cdot|x_m)$.

3.5 Chance constraints

A chance constraint takes the form

$$\mathbb{P}(g(X) \geq 0) \leq \alpha, \tag{3.5}$$

where X is again some random variable of interest and α is a small number (e.g., 0.1, 0.05, 0.001) referred to as the risk tolerance level. This fits into the general constrained formulation given by (1.1) by taking $G \triangleq \mathbb{P}(g(X) \geq 0)$.

3.6 Coherent risk measures

A risk measure $\rho(\cdot)$ is coherent if it satisfies the following conditions (where X and Y are random variables):

- Monotonicity: If $X \leq Y$ a.s. (almost surely), then $\rho(X) \leq \rho(Y)$.
- Sub-additivity: $\rho(X+Y) \leq \rho(X) + \rho(Y)$.
- Positive homogeneity: $\rho(\lambda X) = \lambda \rho(X)$ for any $\lambda \geq 0$.
- Translation invariance: For constant $a > 0$, $\rho(X+a) = \rho(X) + a$.

The sub-additivity requirement is vital, and in the context of portfolio optimization implies diversification cannot lead to increased risk. Note that VaR violates this condition, whereas CVaR is a coherent risk measure.

In the context of an MDP, the r.v. X could correspond to the total cost in an SSP problem, or the cumulative cost in a discounted MDP, or the long-run average cost.

3.7 Cumulative prospect theory (CPT)

CPT is a risk measure that captures human attitudes towards risk. For any r.v. X, the CPT-value is defined as

$$\mathcal{C}(X) \triangleq \int_0^\infty w^+\left(\mathbb{P}\left(u^+(X) > z\right)\right) dz - \int_0^\infty w^-\left(\mathbb{P}\left(u^-(X) > z\right)\right) dz, \tag{3.6}$$

where $u^+, u^- : \mathbb{R} \to \mathbb{R}_+$ are the utility functions that are assumed to be continuous, with $u^+(x) = 0$ when $x \leq 0$ and increasing otherwise, and with $u^-(x) = 0$ when $x \geq 0$ and decreasing otherwise, $w^+, w^- : [0,1] \to [0,1]$ are weight functions assumed to be continuous, non-decreasing and satisfy $w^+(0) = w^-(0) = 0$ and $w^+(1) = w^-(1) = 1$. Note that CPT-value is a generalization of the classic expected value, which can be seen by taking w^+, w^- as the identity functions, $u^+(x) = x^+$ and $u^-(x) = x^-$, where $x^+ \triangleq \max(x, 0)$, $x^- \triangleq \max(-x, 0)$, leading to $\mathcal{C}(X) = \mathbb{E}\left[X^+\right] - \mathbb{E}\left[X^-\right]$.

The human preference to play safe with gains and take risks with losses is captured by a concave gain-utility u^+ and a convex disutility $-u^-$. The weight functions w^+, w^- capture the observed empirical behavior that the value seen by a human subject is nonlinear in the

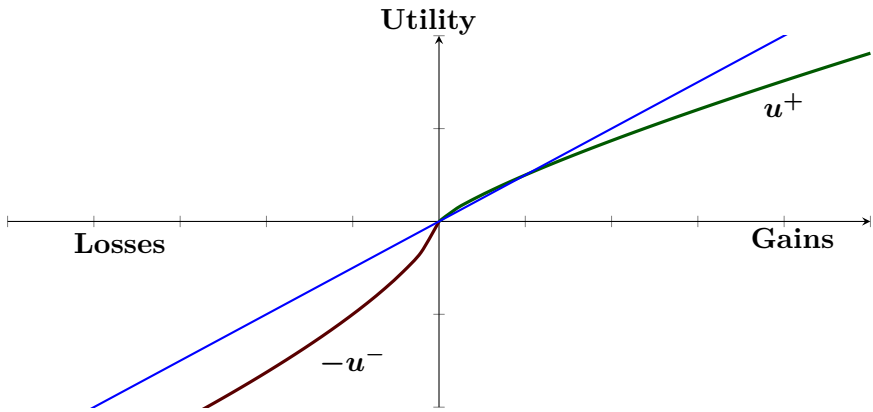


Figure 3.1: An example of a utility function. A reference point on the x axis serves as the point of separating gains and losses. For losses, the disutility $-u^-$ is typically convex and for gains, the utility u^+ is typically concave; both functions are non-decreasing and take the value of zero at the reference point.

underlying probabilities. In particular, humans deflate high probabilities and inflate low probabilities. Examples of utility and weight functions used in practice are shown in Figures 3.1 and 3.2, respectively.

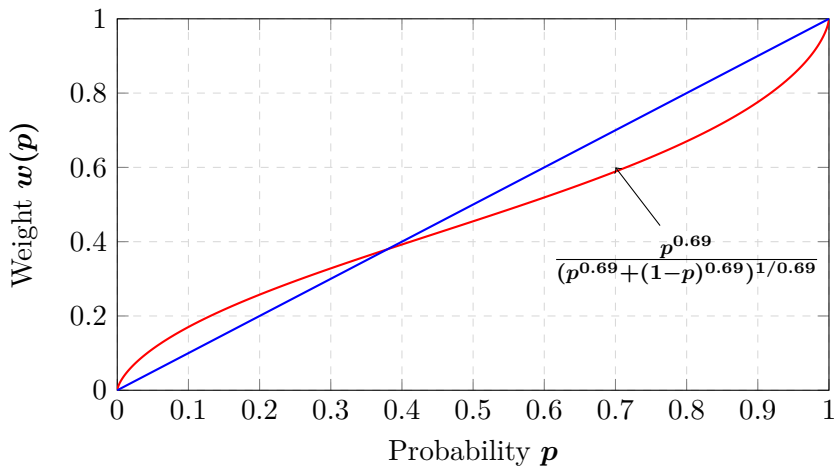


Figure 3.2: An example of a weight function. A typical CPT weight function inflates small probabilities and deflates large probabilities, capturing the tendency of humans doing the same when faced with decisions of uncertain outcomes.

A risk measure based on CPT in a typical MDP setting could apply the CPT-functional to a risk-neutral objective. For instance, take the r.v. X in (3.6) to be either the total cost in an SSP problem or the infinite-horizon cumulative cost in a discounted MDP. However, to carry out dynamic programming would require a Bellman optimality equation, which is not readily available, given the non-convex structure of the CPT-value. A different approach is to use a nested formulation, together with the CPT-style probability distortion. Basically, the formulation is equivalent to optimizing the sum of CPT-value period costs rather than the CPT-value of the sum, and by doing so guarantees the existence of a Bellman optimality equation. Intuitively, it makes sense to incorporate CPT for the total reward rather than applying it separately to the reward in each period.

3.8 Bibliographic remarks

What we defined as the "exponential cost" risk measure in Section 3.1 to be used primarily in Section 7 has an extensive literature in the control and the finance/economics/operations research communities, usually referred to as simply risk-sensitive control and exponential utility, respectively; see the bibliographic remarks in Section 1 for numerous references.

The risk measure of Section 3.2 for discounted-cost MDPs was introduced by Sobel (1982), who also derived a fixed-point equation for it. However, as shown there, the operator underlying this equation for variance lacks the monotonicity property. The approach of deriving a fixed-point equation for the square value function U and using it to estimate the variance was introduced in an SSP context in Tamar *et al.* (2013), and later extended to the discounted MDP context in Prashanth and Ghavamzadeh (2016). Establishing T as a contraction mapping can be found in Prashanth and Ghavamzadeh (2016, Lemma 2). For the average-cost MDP, the single-stage variance definition can be found in Filar *et al.* (1989).

For CVaR and chance constraints, see Rockafellar and Uryasev (2000) and Nemirovski and Shapiro (2007). Chance-constrained optimization problems were introduced in Charnes *et al.* (1958); see also Miller and

Wagner (1965) and Prékopa (1970). The seminal paper introducing coherent risk measures is Artzner *et al.* (1999).

Empirical evidence on human behavior such as the observation that they deflate high probabilities and inflate low probabilities can be found in Tversky and Kahneman (1992) and Barberis (2013). For the weight functions, Tversky and Kahneman (1992) recommend $w(p) = \frac{p^\eta}{(p^\eta+(1-p)^\eta)^{1/\eta}}$, whereas Prelec (1998) recommends $w(p) = \exp(-(-\ln p)^\eta)$, with $0 < \eta < 1$. Figure 3.2 is an example of the former with $p = 0.69$. In both forms, the weight function has an inverted-s shape, which is seen to be a good fit from empirical tests on human subjects, as reported by numerous researchers (Conlisk, 1989; Camerer, 1989; Camerer, 1992; Harless, 1992; Sopher and Gigliotti, 1993; Camerer and Ho, 1994; Gonzalez and Wu, 1999; Abdellaoui, 2000).

The nested formulation approach for CPT was adopted by Lin in his PhD dissertation (Lin, 2013); see also Lin and Marcus (2013b), Lin and Marcus (2013a), Lin *et al.* (2018), and Cavus and Ruszczynski (2014).

Finally, we note that there are several risk measures that have not been explored directly in the risk-sensitive RL literature. For instance, spectral risk measure (SRM) (Acerbi, 2002) and utility-based shortfall risk (UBSR) (Föllmer and Schied, 2002). SRM generalizes CVaR, while retaining coherency. In particular, SRM employs a 'risk-aversion' function to weigh losses. Note that VaR gives a zero weight for each loss beyond a given quantile, while CVaR assigns a constant weight in the tail region beyond VaR. On the other hand, the risk-aversion function in SRM allows one to assign larger weights to higher losses, and thus model a user's risk attitude better. Moreover, a positive, increasing risk-aversion function that integrates to one would imply coherency of SRM. Finally, SRMs are also equivalent to the class of distortion risk measures (DRMs), when the risk-aversion function satisfies the aforementioned properties (Balbás *et al.*, 2009). DRM employs a weight function to distort probabilities, and a concave weight function ensures coherency of DRMs. The CPT risk measure that we presented is more general than DRMs, as the weight function employed there is neither convex nor concave. Next, UBSR is a special instance of a convex risk measure, which is a generalization of coherency. This can be inferred from the fact

that subadditivity and positive homogeneity – properties imposed for coherency – imply convexity. UBSR involves a utility function that can be chosen to encode the risk associated with each value of the r.v. X, allowing more flexibility in modeling a user's risk attitude, as compared to CVaR. See Föllmer and Schied (2016) for a textbook introduction to the class of convex risk measures in general and UBSR in particular.

4

Background on Policy Evaluation and Gradient Estimation

TD-learning and gradient estimation serve as building blocks for policy gradient algorithms, in both the risk-neutral and risk-sensitive MDP contexts. This section provides the basic background on these two topics needed to understand the remainder of the monograph. Two of the most commonly used gradient estimation approaches in policy gradient algorithms, simultaneous perturbation stochastic approximation (SPSA) and the likelihood ratio (LR) method, are covered, as they are employed in the policy gradient algorithms of Sections 5, 6, and 7. A reader familiar with this background material can skip this section.

4.1 Stochastic approximation (SA)

The goal of stochastic approximation (SA) is to find a root of an unknown real-valued function, denoted here by $H : \mathbb{R}^d \longrightarrow \mathbb{R}$. Specifically, SA aims to find a $\theta^* \in \mathbb{R}^d$ that solves the equation $H(\theta^*) = 0$, where only noisy estimates of H are available, i.e., an estimator $\hat{H} = H + \xi$, where ξ is a random variable representing the noise. If ξ is zero mean, then the estimator $\hat{H}$ is said to be unbiased. The primary setting of interest in this monograph is where H represents a gradient such that a zero of $H(\theta)$, say θ^*, corresponds to a (local) optimum.

4.1.1 Basic algorithm

The seminal Robbins-Monro (RM) algorithm solved this problem by employing the following SA iterative update rule:

$$\theta_{n+1} = \theta_n + \zeta(n)(H(\theta_n) + \xi_n), \tag{4.1}$$

where $\{\zeta(n)\}$ is a step-size sequence; commonly used choices include $\zeta(n) = c/n$ and $\zeta(n) = c$, for some constant $c > 0$. In the original RM setting, $\{\xi_n\}$ is an i.i.d. zero-mean sequence, but more generally it could be a martingale-difference sequence (see A4.4 below). In the optimization setting where an unbiased estimator can be obtained such as through the LR method described in Section 4.5, the parameter update will take the form (4.1), with the gradient estimator given by $\hat{H} = H + \xi$.

The RM algorithm can converge even if the function measurements contain an additional bias term that vanishes asymptotically, in which case the SA update iteration takes the following form in place of (4.1):

$$\theta_{n+1} = \theta_n + \zeta(n)(H(\theta_n) + \xi_n + \beta_n), \tag{4.2}$$

where β_n is an asymptotically vanishing bias term (see A4.2 below). In applications involving biased function measurements, such as SPSA discussed in Section 4.4, the parameter update will take the form (4.2), with (biased) gradient estimator $\hat{H} = H + \xi + \beta$, where the bias term β_n vanishes asymptotically.

4.1.2 Asymptotic convergence

For the asymptotic analysis of (4.1) and (4.2), we follow the ordinary differential equation (ODE) approach, where the main idea is to show that the algorithm in (4.1) or (4.2) is a noisy discretization of the following ODE:

$$\dot{\theta}(t) = H(\theta(t)). \tag{4.3}$$

In the absence of ξ_n and β_n, it is apparent that (4.2) is a Euler discretization of the ODE defined above, and hence the algorithm (4.2)

would converge to the equilibria of the aforementioned ODE. The analysis under the ODE approach would show that the algorithm (4.2) tracks the above ODE, even in the presence of noise ξ_n and bias β_n. For establishing this claim, we make the following assumptions:

A4.1. $H : \mathbb{R}^d \longrightarrow \mathbb{R}$ is Lipschitz continuous.

A4.2. The sequence $\{\beta_n\}$ is a bounded random sequence with $\beta_n \to 0$ almost surely (a.s.) as $n \to \infty$.

A4.3. The sequence $\{\zeta(n)\}$ satisfies $\zeta(n) \to 0$ and $\sum_{n=0}^{\infty} \zeta(n) = \infty$.

A4.4. $\{\xi_n\}$ is a sequence such that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{m\geq n} \left\| \sum_{i=n}^{m} \zeta(i)\xi_i \right\| \geq \epsilon \right) = 0.$$

A4.5. $\sup_n \|\theta_n\| < \infty$ a.s.

We now discuss these assumptions. A4.1 ensures that the ODE (4.3) is well-posed. A4.2 ensures that the bias β_n vanishes asymptotically. A4.3 contains standard stochastic approximation conditions on the step sizes $\{\zeta(n)\}$. The condition $\sum_n \zeta(n) = \infty$ ensures that the entire time axis is covered, since $\zeta(n)$ can be seen as the discrete time steps, while $\zeta(n) \to 0$ ensures the discretization errors can be ignored. A4.4 imposes conditions on the noise ξ_n that ensure the effect of noise is asymptotically negligible.

A typical SA convergence result for the RM algorithm is as follows.

Theorem 4.1. Assume A4.1–A4.5. Then θ_n governed by (4.2) converges a.s. to the set $\{\theta^* \mid H(\theta^*) = 0\}$.

Note that for the original RM algorithm given by (4.1), where the bias term is absent, A4.2 is automatically satisfied. This particular result is often referred to in the SA literature as the Kushner-Clark Lemma. Convergence to a set is interpreted as follows. If the set consists of a single point, then the convergence would be to that point. If all the elements in the set are disconnected, then convergence would be to a single point in the set, with the specific point to which the algorithm

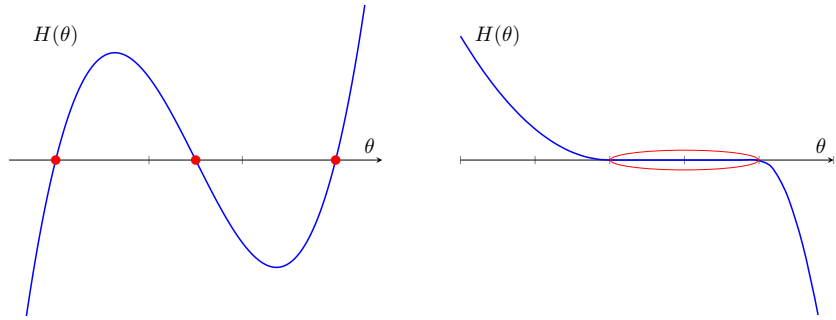


Figure 4.1: Functions illustrating the two main types of SA convergence to the zero(s) of the function. In the left graph, the SA algorithm would converge to one of the three points at which the function crosses the x-axis (indicated by the large red circles), where which one it reaches depends on the starting point and the noise. In the right graph, the SA algorithm would eventually bounce between points in the interval (circled in red) on the x-axis unless the noise goes to zero.

converges depending on the initial condition, the step-size sequence, and the noise. If some of the points are connected, then the algorithm could "bounce" between such points and not converge to a single point. These possibilities are illustrated in Figure 4.1.

The noise sequence $\{\xi_n\}$ is generally assumed to be a martingale difference, i.e., $\mathbb{E}(\xi_n \mid \mathcal{F}_n) = 0$, where $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denotes the underlying σ-field, in which case, A4.3 and A4.4 can be replaced with the following:

A4.6. The sequence $\{\zeta(n)\}$ satisfies $\sum_{n=0}^{\infty} \zeta(n) = \infty, \sum_{n=0}^{\infty} \zeta(n)^2 < \infty$.

A4.7. $\{\xi_n\}$ is a square-integrable martingale-difference sequence satisfying $\mathbb{E}[\|\xi_n\|^2 \mid \mathcal{F}_n] \leq C_0(1 + \|\theta_n\|^2)\ \forall n \geq 0$, for some constant C_0.

To see that the above two assumptions in conjunction with A4.5 imply A4.4, we use Doob's martingale inequality, given as follows: For a martingale sequence $\{W_m\}$,

$$\mathbb{P}\left(\sup_{m \geq 0} \|W_m\| \geq \epsilon\right) \leq \frac{1}{\epsilon^2} \lim_{m \to \infty} \mathbb{E}\,\|W_m\|^2 . \tag{4.4}$$

Apply the inequality above to $\{\sum_{n=k}^{l} \zeta(n)\xi_n\}_{l\geq k}$ to obtain

$$\lim_{k\to\infty} \mathbb{P}\left(\sup_{l\geq k}\left\|\sum_{n=k}^{l}\zeta(n)\xi_n\right\| \geq \epsilon\right) \leq \frac{1}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\zeta(n)^2\mathbb{E}\left\|\xi_n\right\|^2$$
$$\leq \frac{\text{const}}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\zeta(n)^2 = 0,$$

where the final inequality used the following facts: (i) $\mathbb{E}\left\|\xi_n\right\|^2$ is bounded above since the iterate θ_n is bounded a.s. from A4.5 and the noise satisfies a linear growth condition as specified in A4.7; (ii) the step sizes are square summable from A4.6. Thus, A4.4 is satisfied.

SA is useful in solving several subproblems in risk-sensitive RL. For example, TD-learning is an instance of an SA algorithm that incorporates a fixed-point iteration. While regular TD-learning is useful in estimating $J(\theta)$, a variant will be useful in estimating variance indirectly (see Section 6.1). Moreover, VaR estimation is performed using an SA scheme that features a stochastic gradient descent-type update iteration, while CVaR estimation is a plain averaging rule that can be done through SA, as well.

If H represents a gradient, say $H = \nabla h$, the RM algorithm becomes a stochastic gradient descent (for minimization problems) scheme with the following SA iterate updating equation:

$$\theta_{n+1} = \theta_n - \zeta(n)\widehat{\nabla} h(\theta_n), \tag{4.5}$$

where $\widehat{\nabla} h(\theta_n) = \nabla h(\theta_n) + \xi_n + \beta_n$ denotes the gradient estimator.

In the context of RL, the policy parameter updates in a policy gradient algorithm for solving risk-neutral/risk-sensitive MDPs are of this form. A straightforward specialization of the result in Theorem 4.1 leads to the following result:

Theorem 4.2. Assume A4.1–A4.5. Then θ_n governed by (4.5) converges a.s. to the set $\{\theta^* \mid \nabla h(\theta^*) = 0\}$.

As in the root-finding setting, if the set consists of a single point, then the convergence would be to that point. Otherwise, the meaning of

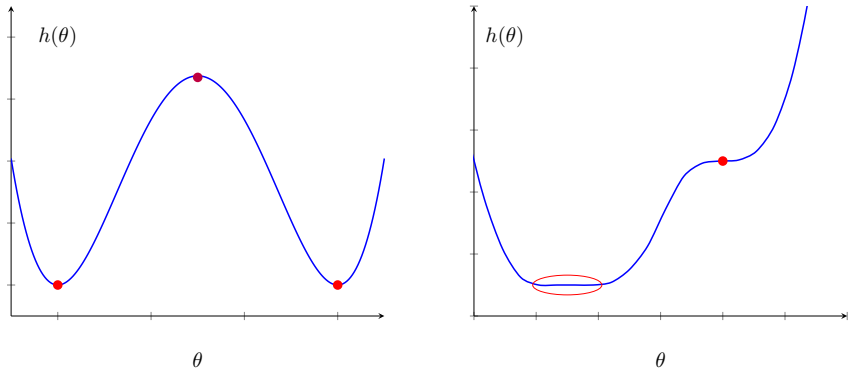


Figure 4.2: Two graphs illustrating the types of SA convergence for stochastic optimization. In the left graph, an SA algorithm for minimization would converge to one of the two local minima or the local maximum indicated by the filled (red) circles, where which one it reaches depends on the starting point and the noise. In the right graph, the SA algorithm could converge to the saddle point indicated by the filled (red) circle or would eventually bounce between points in the circled (in red) interval unless the noise goes to zero. As long as the gradient estimate remains appropriately noisy, the SA algorithm would eventually move away from the local maximum in the left graph and away from the saddle point in the right graph.

convergence to a set is depicted by two graphs in Figure 4.2. If all the elements in the set are disconnected, then convergence would be to a single point in the set, with the specific point to which the algorithm converges depending on the initial condition, the step-size sequence, and the noise, as illustrated in the left graph of Figure 4.2, which contains two local minima and one local maximum. If some of the points are connected, then the algorithm could "bounce" between such points and not converge to a single point, as illustrated in in the right graph of Figure 4.2, which contains a flat local minimal region and a saddle point. "Unstable" points such as local maxima (in minimization problems) and saddle points can be avoided by ensuring that the gradient estimate is suitably noisy, to be described in more detail now.

Since the ODE tracked by the iteration (4.5) is $\dot{\theta}(t) = \nabla h(\theta(t))$, we know that its stationary points will be local maxima or minima, saddle points, or points of inflection. If these points are isolated, then the algorithm (4.5) will a.s. converge to a sample path-dependent stationary

point. Under additional assumptions, one can ensure convergence to a local minimum, i.e., avoid local maxima and saddle points. One such assumption is that the stationary points are hyperbolic, i.e., the Hessian $\nabla^2 h$ does not have eigenvalues on the imaginary axis. Then locally, it has a 'stable manifold' of dimension equal to the number of eigenvalues in the left half plane and an unstable manifold with the complementary dimension. A trajectory on the former converges to the stationary point along the stable manifold, whereas one on the latter moves away from it on the unstable manifold. A trajectory initiated anywhere else also eventually moves away. Thus, if there is at least one unstable eigenvalue, the trajectories move away from the stationary point except on the stable manifold, a set of zero Lebesgue measure. Hence, if the noise is omnidirectional, i.e., rich in all directions in a certain precise sense, the iterations will be pushed away from the stable manifold often enough for the iterates to move away from the stationary point for good, a.s. Then the iterates will a.s. converge to a local minimum, where there are no unstable directions. In case the conditions on noise cannot be verified for the problem at hand, one can always add extraneous i.i.d. zero mean noise, i.e., an SA update iteration of the form

$$\theta_{n+1} = \theta_n - \zeta(n)(\widehat{\nabla} h(\theta_n) + \varphi_n), \tag{4.6}$$

where φ_n is extraneous noise added to ensure that the algorithm avoids saddle points/local maxima. A simple choice is to sample φ_n from the d-dimensional unit sphere uniformly. In practice, it may not be necessary to add such a noise factor extraneously, since the algorithm has an inherent noise component in the gradient estimates.

4.1.3 Projected stochastic approximation

Theorem 4.1 imposes a stability requirement on the iterates, i.e., the condition $\sup_n \|\theta_n\| < \infty$. This requirement is not easy to ensure in RL applications, where one considers a policy gradient-type algorithm for finding the optima of a non-convex objective function. In such situations, an alternative is to employ projections to artificially ensure stability of iterates.

A projected stochastic approximation algorithm would involve the following update iteration:

$$\theta_{n+1} = \Gamma(\theta_n + \zeta(n)(H(\theta_n) + \xi_n + \beta_n)), \tag{4.7}$$

where Γ is a projection into a compact and convex set, say $\Theta \subset \mathbb{R}^d$.

The ODE associated with (4.7) is given by

$$\dot{\theta} = \check{\Gamma}(H(\theta)), \tag{4.8}$$

where $\check{\Gamma}$ is a projection operator that keeps the ODE evolution within the set Θ, defined as follows: For any bounded continuous function $f(\cdot)$,

$$\check{\Gamma}(f(\theta)) = \lim_{\tau \to 0} \frac{\Gamma(\theta + \tau f(\theta)) - \theta}{\tau}.$$

The limit defined above exists because Θ is convex. Furthermore, for θ in the interior of Θ, the projection $\check{\Gamma}(f(\theta)) = f(\theta)$, while for θ on the boundary of Θ, $\check{\Gamma}(f(\theta))$ is the projection of $f(\theta)$ onto the tangent space of the boundary of Θ at θ.

The following theorem presents an asymptotic convergence result for the projected SA iteration (4.7), with assumptions similar to those used in Theorem 4.1 (sans the stability requirement A4.5).

Theorem 4.3. *(Projected stochastic approximation)* Assume A4.1–A4.4. Let $\Theta^* = \{\theta \mid \check{\Gamma}(H(\theta))) = 0\}$ denote the set of limit points of the ODE (4.8). Then θ_n governed by (4.7) converges a.s. to the set Θ^*.

This result is the projected form of the Kushner-Clark Lemma.

4.1.4 A stability result

Recall that the asymptotic convergence result in Theorem 4.1 requires that the iterate θ_n remains bounded a.s. The variant in Theorem 4.3 ensured stability through a projection operator Γ. However, one can do away with the projection operator under certain conditions, and infer both boundedness as well as convergence. The result in this section presents conditions for ensuring stability, and these conditions are usually satisfied in the context of policy evaluation, esp. through TD learning methods (see Section 4.3).

A4.8. For any $\eta \in \mathbb{R}$, define

$$H_\eta(\theta) = H(\eta\theta)/\eta. \tag{4.9}$$

Then, there exists a continuous function H_∞ such that $H_\eta \to H_\infty$ as $\eta \to \infty$ uniformly on compact sets. Furthermore, θ^* is the (unique) globally asymptotically stable equilibrium for the ODE

$$\dot{\theta}(t) = H_\infty(\theta(t)).$$

Assumption A4.8 can be interpreted intuitively as follows: Consider the scaled ODE (4.9), which arises by scaling the iterate to lie within a unit ball and linearly interpolating between the scaled iterate values. The assumption requires that the limit of these scaled functions H_η exist, and the limiting ODE has a globally asymptotically stable equilibrium. Under these conditions, together with A4.7, which implies the effects of the noise is asymptotically negligible, we obtain the following stability result for the original (unprojected) SA algorithm.

Theorem 4.4. Assume A4.1, A4.7, and A4.8. Then for θ_n governed by (4.1), $\sup_n \|\theta_n\| < \infty$ a.s. for any θ_0. Furthermore, θ_n converges a.s. to the set $\{\theta^* \mid H(\theta^*) = 0\}$.

4.2 Contractive stochastic approximation

In many RL problems, the underlying operator is contractive in nature, and the goal is to find the fixed point of such a contraction mapping by observing a sample path of the underlying MDP. Stochastic approximation facilitates finding such a fixed point, and we formalize this claim below.

Given a vector $\nu = (\nu(1), ..., \nu(|\mathcal{X}|))$, with $\nu(i) > 0, \ \forall i$, define the weighted maximum norm of a vector $\theta = (\theta(1), \ldots, \theta(|\mathcal{X}|))$ by

$$\|\theta\|_\nu = \max_i \frac{|\theta(i)|}{\nu(i)}.$$

If $\nu(i) = 1, \ \forall i$, then $\|\cdot\|_\nu$ is the max-norm or ℓ_∞ norm.

Suppose that H is a weighted max-norm contraction, i.e. $\exists$ a positive vector $\nu = (\nu(1), ..., \nu(|\mathcal{X}|))$, and a constant $\beta \in [0, 1)$ such that

$$\|H(\theta) - H(\theta')\|_\nu \leq \beta \|\theta - \theta'\| \qquad \forall \theta, \theta' \in \mathbb{R}^{|\mathcal{X}|}.$$

It is well known that there exists an θ^* that is the unique fixed point of the contraction mapping H, i.e. $H(\theta^*) = \theta^*$.

To see the connection of the norm defined above, recall from the theory of MDPs in Section 2 that (i) in an SSP with all policies proper, the Bellman operator T, as well as the policy-specific operator T_μ, are contractions under a weighted-max norm; and (ii) in a discounted MDP with bounded single-stage cost, the Bellman operator T, as well the policy-specific operator T_μ, are contractions under the max-norm. For the Bellman operator, the fixed point θ^* would correspond to the optimal cost, while θ^* would be the expected total cost for the policy-specific operator T_μ.

An SA iteration for finding θ^* would take the following form:

$$\theta_{n+1} = \theta_n + \zeta(n)(H(\theta_n) + \xi_n - \theta_n), \tag{4.10}$$

where ξ_n is the noise element and $\zeta(n)$ is the step size, as in the previous section. The SA algorithm uses the noisy observation $H(\theta_n)(i) + \xi_n(i)$ to perform an incremental update in each component i. In RL settings, each component would denote a state, and θ_n would be an estimate of the value function or the optimal value function, based on whether the problem is value prediction or control, respectively.

We now state an asymptotic convergence result for the iterate θ_n to the fixed point θ^* of H.

Theorem 4.5. Assume A4.3, A4.4, A4.5, and that H is a weighted max-norm contraction. Then θ_n governed by (4.10) converges a.s. to the fixed point of H, i.e., $\theta_n \longrightarrow \theta^*$ a.s., where $\theta^* = H(\theta^*)$.

4.3 Temporal-difference (TD) learning

A key algorithm for policy evaluation in RL is TD learning. The objective of TD-learning is to estimate the value function $J_\mu(x)$ for a given policy μ. In this section, we first present tabular TD learning, i.e., a setting where the state space is small allowing one to store a lookup table with an entry for each state. Subsequently, we cover TD learning with

linear function approximation – an algorithm that can handle large state spaces by employing feature-based representations.

4.3.1 Tabular TD learning

For ease of exposition, we consider the TD(0) algorithm in a discounted-cost MDP setting. Recall that the value function $J_\mu(x)$ satisfies the following fixed point relation:

$$J_\mu(x) = \mathbb{E}\left[k(x,a) + \gamma J_\mu(y)\right], \tag{4.11}$$

where the expectation is over the random action a chosen according to $\mu(\cdot|x)$, and the next state y, which is sampled from $P(\cdot|x,a)$. Now, in order to estimate the expectation, we take a sample of the expression within the expectation, and apply the update rule as shown below: Starting with any J_0, the TD(0) algorithm iteratively updates an estimate J_{n+1} at iteration $n+1$ using the observed sample cost $k(x_n,a_n)$, $a_n \sim \mu^\theta(\cdot|x_n)$, and previous estimate J_n as follows:

$$J_{n+1}(x) = J_n(x) + \zeta(\nu(x,n))\mathbb{I}\{x_n = x\}[k(x_n,a_n) + \gamma J_n(x_{n+1}) - J_n(x_n)], \tag{4.12}$$

where $x_{n+1} \sim P(\cdot|x_n,a_n)$, $\nu(x,n) = \sum_{m=0}^{n} \mathbb{I}\{x_m = x\}$ is the number of visits to state x, and $\{\zeta(\cdot)\}$ is a step-size sequence. Note that (4.12) is an iterative means of trying to find the zero of the fixed-point equation given by (4.11) by taking the difference between estimates of the value function given by $J_n(x_n)$ for the LHS of (4.11) and that given by $k(x_n,a_n) + \gamma J_n(x_{n+1})$ for the RHS of (4.11).

Since $\mathbb{E}\left[k(x,a) + \gamma J(y)\right] = T_\mu J(x)$, the TD(0) update rule (4.12) is equivalent to

$$J_{n+1}(x) = J_n(x) + \zeta(\nu(x,n))\mathbb{I}\{x_n = x\}\left(T_\mu J_n(x) - J_n(x) + \xi_n(x)\right),$$

where ξ_n is the noise term. From the equation above, we can draw the parallel to the stochastic approximation algorithm presented in the previous section, in particular, to observe the ξ_n term is conditionally zero mean, and satisfies the linear growth condition in Theorem 4.5, leading to the following convergence result:

Theorem 4.6. For a discounted MDP with bounded single-stage cost, the TD(0) algorithm (4.12) using step sizes satisfying A4.6 converges a.s., i.e.,

$$J_n \to J_\mu \text{ a.s as } n \to \infty.$$

A similar claim can be made for the case of other MDPs (SSP and average cost), and we omit the details.

4.3.2 TD learning with linear function approximation

While the TD(0) algorithm described above is provably convergent to the true value $J_\mu(x_0)$, this algorithm employs full-state representation, i.e., it requires a lookup table entry for each state $x \in \mathcal{X}$, and thus would be subject to the curse of dimensionality, in terms of potentially intractable growth of the size of the state space. A practical approach to address this problem is to employ feature-based representations and function approximation by approximating the value function as follows:

$$J_\mu(x) \approx v^\intercal \phi(x),$$

where $\phi(x)$ is a d-dimensional feature (column) vector corresponding to the state x, with $d \ll |\mathcal{X}|$ and v is a tunable d-dimensional parameter. Given this approximation architecture, an important question is how to choose v so that we obtain a good enough approximation of J_μ within a linear space. The TD approach is to find a v that solves the following projected system of equations:

$$\Phi v = \Pi T_\mu(\Phi v), \tag{4.13}$$

where Φ is a matrix with rows $\phi(x)^\intercal \ \forall x \in \mathcal{X}$, $T_\mu J = k + \gamma P_\mu J$ is the discounted-cost MDP Bellman operator (2.4) underlying the fixed-point equation for policy μ given in Proposition 2.1, with P_μ representing the transition probability matrix of the Markov chain generated by μ, and Π is an operator that projects onto the linear space $\mathcal{S} = \{\Phi v | v \in \mathbb{R}^d\}$. More precisely, assuming a stationary distribution, say ψ, exists for the Markov chain generated by policy μ, we have $\Pi = \Phi(\Phi^\intercal D \Phi)^{-1}(\Phi^\intercal D)$,

with D denoting a diagonal matrix with entries from the distribution ψ. Define a weighted ℓ_2-norm as

$$\|J\|_\psi^2 = \sum_{i=1}^{|\mathcal{X}|} \psi(i) J(i)^2, \text{ for any } J \in \mathbb{R}^{|\mathcal{X}|}.$$

Then, Π can be seen as the orthogonal projection operator onto the set $\mathcal{S}$ under the norm defined above, i.e., for any J,

$$\Pi J = \arg\min_{\bar{J} \in \mathcal{S}} ||J - \bar{J}||_\psi^2.$$

The projected fixed-point relation in (4.13) can be written equivalently as a linear system of equations, i.e.,

$$\begin{aligned} \Phi v = \Pi T_\mu(\Phi v) \Leftrightarrow Cv = d, \text{ where} \\ C = \Phi^\mathsf{T} D(I - \gamma P_\mu)\Phi, d = \Phi^\mathsf{T} D\mathbf{k}, \end{aligned}$$

and $\mathbf{k}$ is a $|\mathcal{X}|$-dimensional vector with elements $\sum_a k(x,a)\mu(a|x)$. The above equivalence can be seen by noting the following:

$$\begin{aligned} Cv - d &= \Phi^\mathsf{T} D(I - \gamma P_\mu)\Phi v - \Phi^\mathsf{T} D\mathbf{k} \\ &= \sum_{x,y} \psi(x) P_{xy} \phi(x) \left(\phi(x)^\mathsf{T} v - \gamma \phi(y)^\mathsf{T} v - \sum_a k(x,a)\mu(a|x) \right) \\ &= \mathbb{E}_\psi \left[\phi(x) \left(\phi(x)^\mathsf{T} v - \gamma \phi(y)^\mathsf{T} v - \sum_a k(x,a)\mu(a|x) \right) \right], \end{aligned}$$

where $\mathbb{E}_\psi$ denotes expectation with respect to one step in a Markov chain generated by policy μ that starts in the stationary distribution ψ and P_{xy} is the x-yth entry in the one-step transition matrix P_μ.

Thus, finding the TD fixed point is equivalent to obtaining a v such that the expectation on the RHS above is zero. The obvious method for finding such a v requires sampling a state x from the stationary distribution ψ and the next state y from $P_{x\cdot} \triangleq P(\cdot|x, \mu(x))$. However, in practice, the stationary distribution ψ is unknown, so sampling from it may not be feasible. Instead, assuming an initial distribution ν_0, the samples seen by TD(0) would be coming from $\nu_0 P_\mu^n$, which under suitable mixing assumptions converges to the stationary distribution ψ. This motivates the following update rule for the TD(0) algorithm:

$$v_{n+1} = v_n + \zeta(n)\phi(x_n)(k(x_n, a_n) + \gamma v_n^{\mathsf{T}}\phi(x_{n+1}) - v_n^{\mathsf{T}}\phi(x_n)), \tag{4.14}$$

where v_0 is set arbitrarily, $a_n \sim \mu(\cdot|x_n)$ and $\{\zeta(n)\}$ is a step-size sequence satisfying standard SA conditions.

The convergence analysis of the TD(0) algorithm utilizing (4.14) is more complicated than the usual RM-based TD algorithms presented previously. The stochastic approximation schemes presented in earlier sections assumed that the noise elements came from a martingale difference sequence, whereas iteration (4.14) has a transient mixing phase before the samples are seen from the stationary distribution ψ.

To show that TD(0) with linear function approximation converges to solution of $Cv - d = 0$, we make the following assumptions:

A4.9. The Markov chain induced by the policy μ is irreducible and aperiodic. Moreover, there exists a stationary distribution $\psi(= \psi_\mu)$ for this Markov chain. Let $\mathbb{E}_\psi$ denote the expectation w.r.t. this distribution.

A4.10. The matrix Φ with rows $\phi(x)^{\mathsf{T}}, \forall x \in \mathcal{X}$ has full column rank.

A4.11. The single-period cost function satisfies $\mathbb{E}_\psi(k^2(x, \mu(x))) < \infty$, $\forall x \in \mathcal{X}$.

A4.12. The feature vector $\phi_i(x)$ satisfies $\mathbb{E}_\psi(\phi_i^2(x)) < \infty$, $\forall x \in \mathcal{X}$, $i = 1, \ldots, d$.

A4.13. For the Markov chain $\{x_t\}$ with stationary distribution ψ induced by policy μ, there exists a non-negative bounded function $B(\cdot)$ such that for any $q > 1$, there exists a non-negative constant $K_q < \infty$ satisfying $\mathbb{E}[B^q(x_m) \mid x_0] \leq K_q B^q(x_0)$ for all $x_0 \in \mathcal{X}$. Furthermore,

$$\sum_{n=0}^{\infty} \|\mathbb{E}[k(x_n, \mu(x_n))\ \phi(x_n)|\, x_0] - \mathbb{E}_\psi[k(x_n, \mu(x_n))\phi(x_n)]\| \leq B(x_0),$$

$$\sum_{n=0}^{\infty} \|\mathbb{E}[\phi(x_n)\phi(x_{n+m})^{\mathsf{T}} \mid x_0] - \mathbb{E}_\psi[\phi(x_n)\phi(x_{n+m})^{\mathsf{T}}]\| \leq B(x_0).$$

We briefly discuss the assumptions. A4.9 is an ergodicity requirement that is necessary to ensure that the operator ΠT_μ is a contraction

mapping w.r.t. the weighted max-norm $\|\cdot\|_\psi$, with the weights coming from the stationary distribution vector ψ. A full column rank feature matrix (assumed in A4.10) together with the fact that ΠT_μ is a contraction mapping imply that the fixed point v is unique. Next, A4.11 and A4.12 are integrability requirements, which ensure that the effect of noise in the TD(0) update vanishes asymptotically. The conditions in A4.13 are related to the mixing of the Markov chain generated by the given policy μ. These conditions ensure that taking expectation of quantities relevant to the update (4.14) w.r.t. the distribution $\nu_0 P^n$, where ν_0 is the initial distribution, is close to the one with the stationary distribution.

The main result establishing asymptotic convergence of TD(0) with linear function approximation is given below.

Theorem 4.7. For a discounted MDP, assume A4.9–A4.13. Then the TD(0) algorithm for v_n governed by (4.14) using step sizes satisfying A4.6 converges a.s. to the fixed-point solution of the following projected Bellman equation:

$$\Phi v = \Pi T_\mu(\Phi v),$$

where T_μ is the Bellman operator corresponding to policy μ and Π is the orthogonal projection onto the linearly parameterized space $\{\Phi v \mid v \in \mathbb{R}^d\}$, with Φ denoting the feature matrix with rows $\phi(x)^\intercal, \forall x \in \mathcal{X}$.

4.3.3 Average-cost TD learning

TD-learning can be employed to estimate the differential value function $V(\theta, x)$ in an average-cost MDP setting, with the Poisson equation in (2.14) as the basis. Notice that the latter equation contains the average cost $J(\theta)$, which has to be estimated from sample data, and then plugged into the TD-learning update rule for estimating V. The TD(0) variant in this case, with policy μ^θ, would update the estimate V_{n+1} as follows:

$$V_{n+1}(x) = V_n(x) + \zeta(\nu(x,n))\mathbb{I}\{x_n = x\}\Big(k(x_n,a_n) - \hat{J}_n + V_n(x_{n+1}) - V_n(x_n)\Big), \tag{4.15}$$

$$\hat{J}_{n+1} = (1-\alpha_n)\hat{J}_n + \alpha_n k(x_n,a_n), \tag{4.16}$$

where $\hat{J}_n$ is the average of the sample single-stage costs seen up to n, $a_n \sim \mu^\theta(\cdot|x_n)$, and $\alpha_n \in (0,1)$.

The parameter update rule in (4.15) can be understood using arguments analogous to those used in the discounted-cost setting for (4.12), modulo the additional need to estimate the average cost of the given policy, a quantity denoted by $\hat{J}_n$, which is also updated iteratively using (4.16), necessitated by the fact that the differential value function satisfies a fixed-point relationship that involves the true average cost (refer to Proposition 2.5 and (2.14).)

The function-approximation variant of TD in the average-cost setting would involve the following update iterations:

$$\delta_n = k(x_n,a_n) - \hat{J}_{n+1} + v_n^\mathsf{T}\phi(x_{n+1}) - v_n^\mathsf{T}\phi(x_n),$$
$$v_{n+1} = v_n + \alpha_n\delta_n\phi(x_n),$$

where $\hat{J}_n$ is updated using (4.16), as in the tabular TD(0) case.

4.4 Simultaneous perturbation stochastic approximation (SPSA)

Suppose we want to solve the optimization problem

$$\min_\theta h(\theta) \triangleq \mathbb{E}\left[\hat{h}(\theta,\xi)\right],$$

where $\hat{h}$ denotes a noisy unbiased estimator for h, which itself is not directly available, and ξ denotes the underlying randomness (noise). For this stochastic optimization problem, the Kiefer-Wolfowitz (KW) algorithm performs gradient descent using a finite-differences estimate for ∇h as follows:

$$\theta_{n+1} = \theta_n - \zeta(n)\widehat{\nabla} h(\theta_n), \tag{4.17}$$

$$\widehat{\nabla}_i h(\theta_n) = \left(\frac{\hat{h}(\theta_n + \delta_n e_i, \xi_{n,i}^+) - \hat{h}(\theta_n - \delta_n e_i, \xi_{n,i}^-)}{2\delta_n} \right), i = 1, \ldots, d, \tag{4.18}$$

where $\{\zeta(n)\}$ is a step-size sequence satisfying standard SA conditions, $\widehat{\nabla}_i h$ denotes the ith element of the gradient estimator, $\{\delta_n\}$ is a sequence of positive perturbation constants, $\xi_{n,i}^+$ and $\xi_{n,i}^-$ are the noise components, and e_i is the unit vector in the ith direction. It can be shown that $\widehat{\nabla}_i h(\theta)$ approaches $\nabla_i h(\theta)$ if $\delta_n \to 0$ as $n \to \infty$. However, this symmetric finite-differences gradient estimator (4.18) requires $2d$ samples of $\hat{h}$ for each iteration of (4.17), whereas the SPSA algorithm requires *only two samples in each iteration, regardless of the parameter dimension* d, estimating the gradient as follows:

$$\widehat{\nabla}_i h(\theta_n) = \left(\frac{\hat{h}(\theta_n + \delta_n \Delta(n), \xi_n^+) - \hat{h}(\theta_n - \delta_n \Delta(n), \xi_n^-)}{2\delta_n \Delta_i(n)} \right), \tag{4.19}$$

where $\Delta(n) = (\Delta_1(n), \ldots, \Delta_d(n))^\intercal$ is a random perturbation vector, with each $\Delta_i(n)$ chosen to be symmetric ± 1-valued Bernoulli r.v.s, and ξ_n^+ and ξ_n^- are the analogous noise components as before in (4.17). Note that in (4.19), the numerator is the same for each component of the gradient estimate vector, and only the denominator is changed, so that in each iteration, there are only two distinct values among the gradient estimate components. One can also rescale the gradient estimate if needed by generalizing the scalar δ_n to a direction-dependent vector.

Like all gradient estimators based on finite differences, the SPSA gradient estimator is generally biased, but the bias can be controlled using the perturbation constant δ_n. In the following, we present a result that establishes that the bias is only of order $O(\delta_n^2)$, hence asymptotically unbiased. For this result, we make the following assumptions:

A4.14. Let $\eta_n^\pm = \hat{h}(\theta_n \pm \delta_n \Delta(n), \xi_n^\pm) - h(\theta_n)$. Let $\mathcal{F}_n = \sigma(\theta_m, m < n)$ denote the underlying σ-field. For all $n \geq 1$, the $\eta_n^\pm$ satisfy

$$\mathbb{E}[\eta_n^+ - \eta_n^- \,|\, \mathcal{F}_n] = 0, \quad \text{and} \quad \mathbb{E}[(\eta_n^+ - \eta_n^-)^2 \,|\, \mathcal{F}_n] \leq \sigma^2 < \infty\,.$$

A4.15. The function h is three times continuously differentiable, with $\left|\nabla^3_{i_1 i_2 i_3} h(\theta)\right| < \tilde{B} < \infty$, for $i_1, i_2, i_3 = 1, \ldots, d$, $\theta \in \mathbb{R}^d$, for some positive constant $\tilde{B}$. Furthermore, the function estimator $\hat{h}$ satisfies $\mathbb{E}[\hat{h}(\theta_n \pm \delta_n \Delta(n))^2] \leq B < \infty$ $n \geq 1$, for some positive constant B.

Proposition 4.1. Assume A4.14–A4.15. Then the SPSA gradient estimator defined by (4.19) satisfies

$$\left|\mathbb{E}\left[\widehat{\nabla}_i h(\theta_n)\right] - \nabla_i h(\theta_n)\right| \leq C_1 \delta_n^2, \text{ for } i = 1, \ldots, d, \text{ and}$$
$$\mathbb{E}\left[\left\|\widehat{\nabla} h(\theta_n) - \mathbb{E}\left[\widehat{\nabla} h(\theta_n) \mid \mathcal{F}_n\right]\right\|^2\right] \leq \frac{C_2}{\delta_n^2},$$

for some positive constants C_1 and C_2.

Proof. Using a Taylor series expansion of h around θ_n, we obtain

$$h(\theta_n \pm \delta_n \Delta(n)) = h(\theta_n) \pm \delta_n \, \Delta(n)^{\mathsf{T}} \nabla h(\theta_n) + \frac{\delta_n^2}{2} \Delta(n)^{\mathsf{T}} \nabla^2 h(\theta_n) \Delta(n)$$
$$\pm \frac{\delta_n^3}{6} \nabla^3 h(\tilde{\theta}_n^{\pm})(\Delta(n) \otimes \Delta(n) \otimes \Delta(n)),$$

where $\otimes$ denotes the Kronecker product, and $\tilde{\theta}_n^+$ (resp. $\tilde{\theta}_n^-$) is on the line segment between θ_n and $(\theta_n + \delta_n \Delta(n))$ (resp. $(\theta_n - \delta_n \Delta(n))$).

Thus, we have

$$\mathbb{E}\left[\frac{h(\theta_n + \delta_n \Delta(n)) - h(\theta_n - \delta_n \Delta(n))}{2\delta_n \Delta_i(n)} \middle| \mathcal{F}_n\right]$$
$$= \mathbb{E}\Bigg(\frac{\Delta(n)^{\mathsf{T}} \nabla h(\theta_n)}{\Delta_i(n)}$$
$$+ \frac{\delta_n^2}{12\Delta_i(n)} (\nabla^3 h(\tilde{\theta}_n^+) + \nabla^3 h(\tilde{\theta}_n^-)(\Delta(n) \otimes \Delta(n) \otimes \Delta(n)) \big| \mathcal{F}_n\Bigg)$$
$$= \nabla_i h(\theta_n) + \mathbb{E}\left[\frac{\delta_n^2}{12\Delta_i(n)} (\nabla^3 h(\tilde{\theta}_n^+) + \nabla^3 h(\tilde{\theta}_n^-)(\Delta(n) \otimes \Delta(n) \otimes \Delta(n)) \mid \mathcal{F}_n\right].$$

To arrive at the final inequality, we used

$$\mathbb{E}\left[\frac{\Delta(n)^{\mathsf{T}} \nabla h(\theta_n)}{\Delta_i(n)} \middle| \mathcal{F}_n\right] = \nabla_i h(\theta_n) + \mathbb{E}\left[\sum_{j=1, j \neq i}^{d} \frac{\Delta_j(n)}{\Delta_i(n)} \nabla_j h(\theta_n)\right] = \nabla_i h(\theta_n),$$

since $\Delta(n)$ is a vector of i.i.d. symmetric ± 1-valued Bernoulli r.v.s.

Using

(i) $\mathbb{E}[\widehat{\nabla}_i h(\theta_n)] = \mathbb{E}\left[\dfrac{h(\theta_n + \delta_n\Delta(n)) - h(\theta_n - \delta_n\Delta(n))}{2\delta_n\Delta_i(n)}\right]$;

(ii) $|\nabla^3 h(\tilde{\theta}_n^{\pm})| < \tilde{B}$; and

(iii) $\mathbb{E}\left[\frac{1}{\Delta_i(n)}|\nabla^3 h(\bar{\theta}_n)(\Delta(n)\otimes\Delta(n)\otimes\Delta(n))|\right] \leq \tilde{B}d^3$ for any $\bar{\theta}_n$,

we have

$$\left|\mathbb{E}\left[\widehat{\nabla}_i h(\theta_n)\right] - \nabla h(\theta_n)\right| \leq C_1\,\delta_n^2\,, \text{ where } C_1 = \frac{\tilde{B}d^3}{6}.$$

Next, we prove the second claim concerning the variance of $\widehat{\nabla} h(\theta_n)$. Notice that

$$\begin{aligned}
&\mathbb{E}\left|\widehat{\nabla}_i h(\theta_n)\right|^2 \\
&= \mathbb{E}\left[\left(\frac{\eta_n^+ - \eta_n^-}{2\delta_n\Delta_i(n)}\right)^2\right. \\
&\qquad +2\left(\frac{\eta_n^+ - \eta_n^-}{2\delta_n\Delta_i(n)}\right)\left(\frac{h(\theta_n + \delta_n\Delta(n)) - h(\theta_n - \delta_n\Delta(n))}{2\delta_n\Delta_i(n)}\right) \\
&\qquad \left.+\left(\frac{h(\theta_n + \delta_n\Delta(n)) - h(\theta_n - \delta_n\Delta(n))}{2\delta_n\Delta_i(n)}\right)^2\right] \\
&= \mathbb{E}\left(\left(\frac{\eta_n^+ - \eta_n^-}{2\delta_n}\right)^2\right) + \mathbb{E}\left(\left(\frac{h(\theta_n + \delta_n\Delta(n)) - h(\theta_n - \delta_n\Delta(n))}{2\delta_n}\right)^2\right) \\
&\leq \frac{C_2}{\delta_n^2}
\end{aligned}$$

where $C_2 = (\sigma^2 + 2B^2)/4$. The last equality above uses $\Delta_i(n)^2 = 1$ and A4.14, while the final inequality uses A4.14 and A4.15. Then the second claim in the proposition follows by using $\mathbb{E}\left\|\widehat{\nabla} h(\theta_n) - \mathbb{E}\widehat{\nabla} h(\theta_n)\right\|^2 \leq 4\mathbb{E}\left\|\widehat{\nabla} h(\theta_n)\right\|^2$ in conjunction with the inequality above. □

Remark 4.1. A variant of (4.19) is to use a one-sided estimate, i.e., given sample observations at $\theta_n + \delta_n \Delta(n)$ and θ_n, use the following gradient estimator:

$$\widehat{\nabla}_i \hat{h}(\theta_n) = \left(\frac{\hat{h}(\theta_n + \delta_n \Delta(n), \xi_n^+) - \hat{h}(\theta_n, \xi_n)}{\delta_n \Delta_i(n)} \right),$$

where δ_n and $\Delta(n)$ are as defined earlier. For solving constrained optimization problems, one-sided estimates are efficient, since a sample observation at the unperturbed value of the underlying parameter is necessary for performing the dual ascent on the Lagrange multiplier. The overall SPSA-based policy gradient algorithm would estimate the necessary gradient, as well as the risk measure, using two sample observations corresponding to $\theta_n + \delta_n \Delta(n)$ and θ_n. On the other hand, using a balanced estimate, as defined in (4.19) would require an additional observation with the underlying parameter set to $\theta_n - \delta_n \Delta(n)$.

The following result establishes that the SPSA algorithm converges to a zero of the gradient, where the SPSA algorithm is defined by the gradient-based SA iteration in (4.17) driven by biased (but asymptotically unbiased) stochastic gradient estimates given by (4.19).

Theorem 4.8. Assume A4.14–A4.15. Let $\Theta^* \triangleq \{\theta \mid \nabla h(\theta) = 0\}$. Then θ_n governed by (4.17) using step sizes satisfying A4.6, with $\widehat{\nabla} h(\theta) \triangleq [\widehat{\nabla}_i h(\theta)]$ defined by (4.19), converges a.s. to a zero of the gradient of h, i.e.,

$$\theta_n \to \Theta^* \text{ a.s. as } n \to \infty.$$

The proof involves an application of Theorem 4.1, and we omit the details. A policy gradient algorithm with SPSA-based gradient estimates is analyzed in Section 5.3, and the proof of Theorem 4.8 would go through using arguments similar to those employed in the proof of Theorem 5.1 in Section 5.3.

4.5 Direct single-run gradient estimation using the likelihood ratio (LR) method

When the system is a complete black box, SPSA is an effective way to carry out gradient-based policy optimization. However, in many settings, more is known about the system, and more efficient *direct* gradient estimation techniques may be applicable, where "direct" means that the gradient estimator is unbiased (as opposed to asymptotically unbiased when finite difference methods such as SPSA are used). The main approaches are perturbation analysis, the likelihood ratio method (also known as the score function method), and weak derivatives (also known as measure-valued differentiation). Here, we consider only the likelihood ratio (LR) method, which is a single-run method for gradients where θ parameterizes the input distribution(s) of the system. "Single-run" means that a gradient estimate can be obtained using a single sample path (or simulation) of the system, in contrast to SPSA, which requires multiple sample paths to estimate the gradient. To motivate the more general case, we first illustrate the idea using a single (discrete-valued) random variable example:

$$\mathbb{E}[X] = \sum_x x \mathbb{P}_\theta(X = x) = \sum_x x p_\theta(x),$$

where p_θ denotes the probability mass function of X. Differentiating with respect to θ (assuming the differentiation operator can be brought inside the summation),

$$\frac{d\mathbb{E}[X]}{d\theta} = \sum_x x \frac{d\mathbb{P}_\theta(X = x)}{d\theta} = \sum_x x \frac{d \ln p_\theta(x)}{d\theta} p_\theta(x) = \mathbb{E}\left[X \frac{d \ln p_\theta(X)}{d\theta}\right],$$

and thus the LR derivative estimator for this simple example is given by

$$X \frac{d \ln p_\theta(X)}{d\theta}.$$

Now consider a Markov chain $\{X_n\}$ with a single recurrent state 0, transient states $1, \ldots, r$, and (one-step) transition probability matrix $P(\theta) := [p_{ij}(\theta)]_{i,j=0}^r$, where $p_{ij}(\theta)$ denotes the (one-step transition) probability of going from state $X_n = i$ to $X_{n+1} = j$ and is parameterized by θ. Let τ denote the first passage time to the recurrent state 0 and

$X := (X_0, \ldots, X_{\tau-1})^\intercal$ denote the corresponding sequence of states (sample path). Assuming θ occurs only in the transition probabilities, an unbiased single-run sample path LR gradient estimator for $\nabla h(\theta)$ is given by

$$\widehat{\nabla} h(\theta) = \hat{h}(X) \nabla \ln p_{X_0 X_1 \cdots X_\tau}(\theta) = \hat{h}(X) \sum_{m=0}^{\tau-1} \frac{\nabla p_{X_m X_{m+1}}(\theta)}{p_{X_m X_{m+1}}(\theta)},$$

where the single random variable is replaced by a function of the Markov chain states visited and the single probability mass function is replaced by a joint distribution $p_{X_0 X_1 \cdots X_\tau}$ that is the product of the individual one-step transition probabilities p_{ij}.

It can be shown under mild conditions on the transition probabilities that the LR gradient estimator is unbiased, i.e.,

$$\mathbb{E}[\widehat{\nabla} h(\theta)] = \nabla h(\theta), \tag{4.20}$$

which follows by invoking the dominated convergence theorem to interchange expectation and differentiation operators. Unbiasedness is a desirable property, because it generally leads to a faster convergence rate for gradient-based algorithms, e.g., an asymptotic rate of $1/\sqrt{n}$ in (4.1) rather than $1/n^{1/3}$ for the typical finite-difference gradient estimate (see, e.g., Theorem 5.2). However, it should also be noted that if θ is a common parameter that appears in all of the probabilities, then this LR estimator will have the undesirable property of its variance increasing linearly with the sample path length.

Remark 4.2. Using the LR gradient estimator in the stochastic gradient algorithm (4.17) would ensure that the resulting algorithm converges to the set of stationary points of the objective, i.e., Theorem 4.8 holds for the LR case, as well.

4.6 Bibliographic remarks

Stochastic approximation has a long history, starting with the seminal paper of Robbins and Monro (1951), where the iterative algorithm (4.1) is used to solve a stochastic root-finding problem. For a proof of the

asymptotic convergence claim in Theorem 4.1 for a Robbins-Monro stochastic approximation scheme, the reader is referred to Theorem 2.3.1 of Kushner and Clark (1978), which is what is referred to in the SA literature as the Kushner-Clark Lemma. For a rigorous introduction to the ODE approach for analyzing stochastic approximation algorithms, the reader is referred to Borkar (2008). The claim in Theorem 4.5 is a special case of the result in Theorem 4.1 for a general stochastic approximation scheme, and the interested reader is referred to Chapters 4 and 5 of Bertsekas and Tsitsiklis (1996) for the proof as well as RL applications. The convergence result for the projected stochastic approximation scheme in Theorem 4.3 is the projected form of the Kushner-Clark Lemma, which is Theorem 5.3.1 of Kushner and Clark (1978). The noise conditions referred to in Section 4.1 for avoiding traps (e.g., local maxima, saddle points) are given in Pemantle (1990); see also Section 4.3 of Borkar (2008). The stability result presented in Theorem 4.4 is the Borkar-Meyn theorem; see Theorems 2.1-2.2(i) of Borkar and Meyn (2000) and also Chapter 3 of Borkar (2008). A popular idea that improves the convergence guarantees for general stochastic approximation algorithms is iterate averaging, proposed independently by Polyak (Polyak and Juditsky, 1992) and Ruppert (Ruppert, 1991). The idea behind this scheme is to use larger step-sizes of $\Theta(1/n^{\varsigma})$ for some $\varsigma \in (1/2, 1)$ to perform the update iteration (4.2), and then use the averaged iterate $\bar{\theta}_{n+1} = \frac{1}{n}\sum_{k=1}^{n} \theta_k$ instead of the last iterate θ_n for providing the convergence guarantees. For the special class of stochastic gradient algorithms that solve a minimization problem, popular approaches for improving the convergence rate is to incorporate second-order information and/or variance reduction, see Bottou *et al.* (2018) and Gower *et al.* (2020) for recent surveys on these topics.

The method of temporal differences for policy evaluation was proposed by Sutton (1988). For an analysis of TD learning, the reader is referred to either Chapters 5 and 6 of Bertsekas and Tsitsiklis (1996) or Chapter 6 of Sutton and Barto (2018). An analysis of the extension of TD to incorporate linear function function approximation can be found in Tsitsiklis and Van Roy (1997). The stability of the TD iterate can also be inferred using the Borkar-Meyn theorem, referred above. Non-asymptotic analysis of TD with linear function approximation has

received a lot of attention recently, and a few representative works are Prashanth *et al.* (2021), Dalal *et al.* (2018), Bhandari *et al.* (2018), and Srikant and Ying (2019). Average-cost TD learning algorithm with full state representations and its convergence analysis can be found in Konda and Borkar (1999), in particular, the faster timescale recursion in Algorithm 4 there. Also related is the average-cost Q-learning algorithm with full state representations, which has been proposed/analyzed by Abounadi *et al.* (2001). For general conditions to infer stability of TD/Q-learning algorithms in an average-cost MDP, see Abounadi *et al.* (2002). Finally, for the extension of average-cost TD algorithm to incorporate feature-based representations and linear function approximation, see Tsitsiklis and Van Roy (1999).

Kiefer and Wolfowitz (1952) extended the Robbins-Monro algorithm for root finding to optimizing a function through gradient search. The Kiefer-Wolfowitz algorithm requires $2d$ function evaluations/estimates per iteration to estimate the gradient, whereas the SPSA algorithm proposed by Spall (1992) estimates the gradient using only two function evaluations/estimates per iteration, regardless of the problem dimension d. For a closely related random directions SA algorithm, see Kushner and Clark (1978, pp. 58-60) and Prashanth *et al.* (2018), and for a detailed introduction to such gradient estimation methods, see Bhatnagar *et al.* (2013). The convergence analysis of the SPSA algorithm presented here, in particular, the main result in Theorem 4.8 can be inferred by an application of the Kushner-Clark Lemma (Kushner and Clark, 1978).

The likelihood ratio (LR) method for gradient estimation, also known as the score function method, has its roots in a 1968 Russian paper (Aleksandrov *et al.*, 1968), but the technique appeared to be unknown to the rest of the world until it was "rediscovered" by several different researchers in parallel (Rubinstein, 1989; Glynn, 1987; Reiman and Weiss, 1989); see Fu (2006) and Fu (2015) for an introduction and overview. For the mild conditions justifying the unbiasedness of (4.20), the reader is referred to Section VII.3 of Asmussen and Glynn (2007). The weak derivatives (WD) method, also known as measure-valued differentiation (Pflug, 1989; Pflug, 1996; Heidergott and Vázquez-Abad, 2000), could also be applied in the Markov chain context, but it would generally require multiple sample paths of the chain or appropriate ran-

domization; however, unlike finite-difference-based methods, including the gradient estimators used in SPSA, LR and WD gradient estimators are generally unbiased (as opposed to asymptotically unbiased).

5

Policy Gradient Templates for Risk-sensitive RL

In this section, we present and analyze template algorithms for two risk-sensitive RL settings. The first setting, described in Section 5.1, incorporates the risk measure directly in the objective function, whereas the second setting, described in Section 5.2, considers the risk measure in a constrained formulation, with the usual (risk-neutral) cost function as the objective. Sections 5.3 and 5.4 analyze the convergence of the template algorithms for the risk-as-objective and risk-as-constraint settings, respectively.

Recall that $\left\{\mu^\theta(\cdot|x), x \in \mathcal{X}, \theta \in \Theta \subseteq \mathbb{R}^d\right\}$ is a parameterized set of randomized policies, and the goal is to find a policy that optimizes a risk measure as an objective, or optimizes the usual risk-neutral objective (cost/reward) function while satisfying a risk constraint. The policy parameterization is assumed to be smooth (cf. Sections 5.3 and 5.4), and a commonly used class of distributions that ensures a smooth parameterization is the 'Boltzmann family' taking the form

$$\mu^\theta(a|x) = \frac{\exp(\theta^\mathsf{T}\phi_{x,a})}{\sum_{a' \in \mathcal{A}(x)} \exp(\theta^\mathsf{T}\phi_{x,a'})}, \ \forall x \in \mathcal{X} \ , \ \forall a \in \mathcal{A}(x),$$

where θ is constrained to be in a convex and compact set $\Theta \subset \mathbb{R}^d$ and $\{\phi_{x,a}\}$ is a set of state-action features.

The proposed policy gradient algorithms attempt to find a 'good enough' policy that optimizes a risk-sensitive objective by performing gradient descent in the policy space, where the concept of a 'good enough' policy will be made precise in the convergence analysis of these algorithms.

For notational convenience, we define

$$J(\theta) \triangleq J_{\mu^\theta}, \quad D(\theta) \triangleq D_{\mu^\theta}, \quad G(\theta) \triangleq G_{\mu^\theta}$$

in MDP contexts where the objective is independent of the initial state, e.g., average-cost MDPs. Analogously, for the discounted-cost MDPs and SSP settings, which depend on the initial state, we define

$$J(\theta, x) \triangleq J_{\mu^\theta}(x), D(\theta, x) \triangleq D_{\mu^\theta}(x), G(\theta, x) \triangleq G_{\mu^\theta}(x), \quad \forall x \in \mathcal{X}.$$

5.1 Template for the setting with risk as objective

This setting considers the following optimization problem:

$$\min_{\theta \in \Theta} G(\theta),$$

where G involves one of the risk measures presented in Section 3. Solving the problem (5.1) via a policy gradient algorithm invokes the following stochastic approximation (SA) iterative update:

$$\theta_{n+1} = \Gamma\left[\theta_n - \zeta(n)\widehat{\nabla} G(\theta_n)\right], \tag{5.1}$$

where $\{\zeta(n)\}$ is a step-size sequence, $\widehat{\nabla} G(\theta_n)$ is an estimate of $\nabla G(\theta_n)$, and Γ is a projection operator that keeps the iterate θ_n bounded within a convex and compact set Θ. A simple choice for the projected region is $\Theta := \prod_{i=1}^{d}[\theta_{\min}^{(i)}, \theta_{\max}^{(i)}]$, which leads to the following simple implementation for the projection operator: For any $\theta \in \mathbb{R}^d$, $\Gamma(\theta) = (\Gamma^{(1)}(\theta^{(1)}), \ldots, \Gamma^{(d)}(\theta^{(d)}))^T$, with $\Gamma^{(i)}(\theta^{(i)}) := \min(\max(\theta_{\min}^{(i)}, \theta^{(i)}), \theta_{\max}^{(i)})$.

The convergence analysis of the policy update algorithm in (5.1) is presented in Section 5.3. In particular, we provide both asymptotic and non-asymptotic convergence guarantees there.

5.2 Template for the setting with risk as constraint

Recall that the setting incorporating risk as constraint is given by the following problem:

$$\min_{\theta \in \Theta} J(\theta) \qquad \text{subject to} \qquad G(\theta) \leq \kappa, \tag{5.2}$$

where J is the usual risk-neutral MDP objective, while the constraint G involves one of the risk measures presented in Section 3. In an average-cost formulation, the objective/constraint do not depend on the initial state, whereas they do in total-cost formulations such as SSP and discounted problems. In either case, the template for solving the problem remains the same, and to keep the presentation simple, we have chosen to have the policy parameter only in J and G. In the special cases of Section 6, we shall include the initial state as necessary.

If there is a policy in Θ that satisfies the constraint in (5.2), then it can be inferred that there exists an optimal policy.

Using the Lagrangian approach, we consider the following relaxed MDP problem:

$$\max_{\lambda} \min_{\theta} \Big(L(\theta, \lambda) \triangleq J(\theta) + \lambda \big(G(\theta) - \kappa \big) \Big),$$

where λ is the Lagrange multiplier. The goal here is to find the saddle point of $L(\theta, \lambda)$, i.e., a point (θ^*, λ^*) that satisfies

$$L(\theta, \lambda^*) \geq L(\theta^*, \lambda^*) \geq L(\theta^*, \lambda), \forall \theta \in \Theta, \forall \lambda > 0.$$

For a standard convex optimization problem where the objective $L(\theta, \lambda)$ is convex in θ and concave in λ, one can ensure the existence of a unique saddle point under mild regularity conditions. Further, convergence to this point can be achieved by descending in θ and ascending in λ using $\nabla_\theta L(\theta, \lambda)$ and $\nabla_\lambda L(\theta, \lambda)$, respectively.

However, in the risk-sensitive RL setting, the Lagrangian $L(\theta, \lambda)$ is not necessarily convex in θ, which implies there may not be an unique saddle point. Hence, performing primal descent and dual ascent, one can only get to a local saddle point, i.e., a point (θ^*, λ^*) that is a local minima w.r.t. θ, and local maxima w.r.t λ of the Lagrangian. The problem is further complicated by the fact in a *typical RL* setting,

closed-form evaluation of the Lagrangian for any given policy parameter θ and Lagrange multiplier λ is not feasible. Instead, one can run sample trajectories after fixing the parameters θ and λ, and obtain estimates of the Lagrangian corresponding to the policy parameter.

For the purpose of finding an optimal risk-sensitive policy, a standard procedure would update the policy parameter θ and Lagrange multiplier λ in two nested loops: an inner loop that descends in θ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. θ, and an outer loop that ascends in λ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. λ.

We operate in a setting where we only observe simulated costs of the underlying MDP. Thus, it is required to estimate both J and G for a given θ and then use these estimates to compute an estimate of the gradient of the Lagrangian w.r.t. θ and λ. The gradient $\nabla_\lambda L(\theta, \lambda)$ has a particularly simple form of $(G(\theta) - \kappa)$, suggesting that a sample of the risk measure can be used to perform the dual ascent for Lagrange multiplier λ. On the other hand, the policy gradient $\nabla_\theta L(\theta, \lambda) = \nabla J(\theta) + \lambda \nabla G(\theta)$ is usually complicated and does not lend itself to stochastic programming techniques in a straightforward fashion. We shall address the topic of gradient estimation in the next section, but for presenting the template of the risk-sensitive policy gradient algorithm, we assume the availability of estimators $\widehat{\nabla} J(\theta)$, $\widehat{\nabla} G(\theta)$ and $\widehat{G}(\theta)$ of $\nabla J(\theta)$, $\nabla G(\theta)$ and $G(\theta)$, respectively. Then, using two-timescale stochastic approximation, the inner and outer loops mentioned above can run in parallel, as follows (please refer to Figure 5.1):

$$\lambda_{n+1} = \left[\lambda_n + \zeta_1(n) \left(\widehat{G}(\theta_n) - \kappa\right)\right]^+, \tag{5.3}$$

$$\theta_{n+1} = \Gamma\left[\theta_n - \zeta_2(n) \left(\widehat{\nabla} J(\theta_n) + \lambda_n \widehat{\nabla} G(\theta_n)\right)\right], \tag{5.4}$$

where $[x]^+ = \max(0, x)$ for any real x, Γ is a projection operator that keeps the iterate θ_n stable by projecting onto a compact and convex set Θ, as in the setting considered in Section 5.1; and $\{\zeta_1(n), \zeta_2(n)\}$ are step-size sequences selected such that the θ update is on the faster timescale, and the λ update is on the slower timescale.

The template for a risk-sensitive policy gradient algorithm would involve the following components:

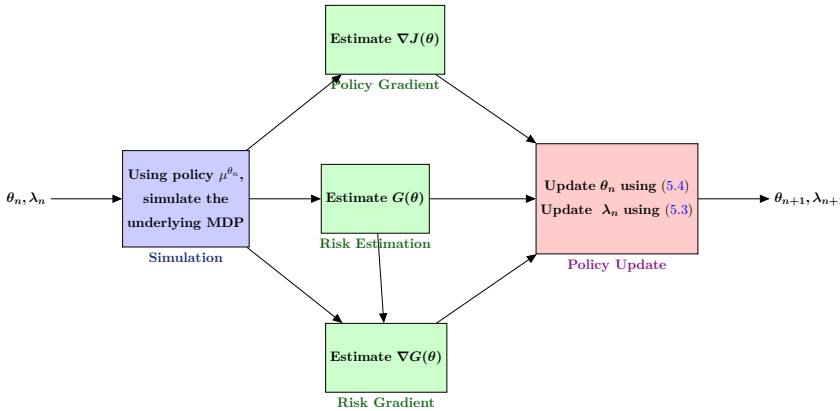


Figure 5.1: Overall flow of risk-sensitive policy gradient algorithm.

1. A two-timescale update rule for the policy parameter θ and the Lagrange multiplier λ;

2. Estimates of the objective $J(\cdot)$ and the risk measure $G(\cdot)$, which can be obtained by sampling from the underlying MDP with the current policy parameter, and then using a suitable estimation scheme, usually based on stochastic approximation (in RL, this would be equivalent to some form of TD-learning), or based on Monte Carlo averaging; the estimate of the objective would feed into estimating the policy gradient (see step below), while the estimate of the risk measure is necessary for the gradient of the Lagrangian, as well as for the dual ascent procedure;

3. Estimates of the gradients $\nabla J(\cdot)$ and $\nabla G(\cdot)$ for primal descent, which may be challenging to obtain if the underlying risk measure has no structure that can be exploited in an MDP framework.

The convergence analysis of the two-timescale SA algorithm given by (5.3) – (5.4) is presented in Section 5.4.

5.3 Convergence analysis in the setting with risk as objective

The following assumptions are used for the convergence analysis of the SA recursive parameter update given by (5.1) in Section 5.1. First, let

$\mathcal{F}_n = \sigma(\theta_m, m < n)$ denote the underlying σ-field.

A5.1. The risk measure G is a continuously differentiable function of the policy parameter θ.

A5.2. For all n, the gradient estimator $\widehat{\nabla} G(\theta_n)$ satisfies

$$\mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right) = \nabla G(\theta_n), \text{ and } \mathbb{E}\left\|\widehat{\nabla} G(\theta_n) - \nabla G(\theta_n)\right\|^2 \leq \sigma^2 < \infty.$$

A5.3. For all n, the gradient estimator $\widehat{\nabla} G(\theta_n)$ depends on a parameter $\delta_n > 0$ and satisfies

$$\left\|\mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right) - \nabla G(\theta_n)\right\| = C_1 \delta_n^2, \text{ and } \mathbb{E}\left\|\widehat{\nabla} G(\theta_n)\right\|^2 < \frac{C_2}{\delta_n^2},$$

where C_1 and C_2 are dimension-dependent constants.

A5.4. The step-size sequence $\{\zeta(n)\}$ satisfies

$$\sum_n \zeta(n) = \infty, \text{ and } \sum_n \zeta(n)^2 < \infty.$$

A5.5. $\delta_n \to 0$ as $n \to \infty$, and the step-size sequence $\{\zeta(n)\}$ satisfies

$$\sum_n \zeta(n) = \infty, \sum_n \left[\frac{\zeta(n)}{\delta_n}\right]^2 < \infty.$$

We now discuss the assumptions. Assumption A5.1 imposes a smoothness requirement on the risk measure G, when viewed as a function of the policy parameter θ. A similar requirement, i.e., the value function J is smooth in θ, holds when the underlying policy parameterization is smooth, i.e., μ^θ is a continuously differentiable function of θ. While a smooth policy parameterization ensures J is smooth, one cannot infer the same for an abstract risk measure G. As an example, one could consider the VaR risk measure. Hence, the second part of A5.1 explicitly imposes the smoothness condition on G, and this condition together with the fact that J is smooth is necessary for the ordinary differential equation (ODE) underlying the θ-recursion to be well-posed. Assumption A5.2 requires that the risk gradient estimator be unbiased with finite variance, which is a standard assumption in the analysis of stochastic gradient schemes, and is generally satisfied

using likelihood ratio-based gradient estimators. Assumption A5.3 is a relaxed variant of A5.2, where the gradient estimators may not be unbiased, as in the case of finite-difference estimators that depend on a difference (perturbation) parameter δ_n, which can be used to control the bias-variance tradeoff — lower δ_n results in lower gradient estimation bias but higher variance, and vice versa. The SPSA approach presented in Section 4.4, as well as the general class of simultaneous perturbation schemes, meet the conditions in A5.3. For convergence, δ_n has to vanish asymptotically, but not too fast, as outlined in the second part of A5.5. The conditions on the step-size sequence $\{\zeta_1(n)\}$ in A5.4 are standard stochastic approximation requirements (see A4.6). Assumption A5.5 is a variant of A5.4, and has to be coupled with A5.3, in the sense that the gradient estimates have a $O(\delta_n^2)$ bias, but $\sum_n \left(\frac{\zeta_2(n)}{\delta_n}\right)^2 < \infty$. The latter requirement ensures that the effect of noise in gradient estimation can be (eventually) ignored, while ensuring asymptotic convergence. Such a requirement that couples the step-size sequence and finite-difference gradient estimator perturbation sequence arises when a biased gradient estimation scheme such as SPSA is employed.

Consider the following ODE underlying the policy update in (5.1):

$$\dot{\theta}(t) = \check{\Gamma}\left(-\nabla G(\theta(t))\right), \tag{5.5}$$

where $\check{\Gamma}(\cdot)$ is a projection operator that ensures the evolution of θ via the ODE (5.5) stays within the set Θ and is defined as follows: For any bounded continuous function $f(\cdot)$,

$$\check{\Gamma}\big(f(\theta)\big) = \lim_{\tau \to 0} \frac{\Gamma\big(\theta + \tau f(\theta)\big) - \theta}{\tau}. \tag{5.6}$$

The limit defined above may not exist and in that case, one can define $\check{\Gamma}(f(\theta))$ to be the set of all possible limit points. Further, for θ in the interior of Θ, $\check{\Gamma}(f(\theta)) = f(\theta)$, while for θ on the boundary of Θ, $\check{\Gamma}(f(\theta))$ is the projection of $f(\theta)$ onto the tangent space of the boundary of Θ at θ.

The main result establishing asymptotic convergence of the policy gradient algorithm (5.1) is given below.

Theorem 5.1. Assume that A5.1, (A5.2 + A5.4) or (A5.3 + A5.5) hold. For θ_n governed by (5.1),

$$\theta_n \to \mathcal{Z} \text{ a.s. as } n \to \infty,$$

where $\mathcal{Z} = \{\theta \in \Theta : \check{\Gamma}(\nabla G(\theta(t))) = 0\}$ is the set of limit points of the ODE (5.5).

Proof. The proof involves an application of the Kushner-Clark lemma for projected stochastic approximation, provided in Section 4.1.3 as Theorem 4.3.

We first rewrite the recursion (5.4) as follows:

$$\theta_{n+1} = \Gamma\Big(\theta_n + \zeta(n)\Big(\nabla G(\theta_n) + \xi_n\Big)\Big), \tag{5.7}$$
$$\text{where } \xi_n = \widehat{\nabla} G(\theta_n) - \nabla G(\theta_n).$$

Application of Theorem 4.3 requires the conditions A4.1–A4.4 to hold, and we verify these conditions for the recursion in (5.7) in the case where we assume A5.1, A5.2, and A5.4. We shall later provide the deviations necessary to handle the case when A5.3, and A5.5 are used in place of A5.2, and A5.4, respectively.

- The smoothness condition in A5.1 implies A4.1.
- Since A5.2 implies an unbiased gradient estimator, the assumption A4.2 trivially holds.
- A5.4 implies A4.3.
- The verification of A4.4 requires the application of a martingale inequality attributed to Doob, which is given in (4.4). We apply this inequality in our setting to the martingale sequence $\{\sum_{n=k}^{l} \zeta(n)\xi_n\}_{l \geq k}$ to obtain

$$\begin{aligned}\lim_{k\to\infty} \mathbb{P}\left(\sup_{l\geq k} \left\| \sum_{n=k}^{l} \zeta(n)\xi_n \right\| \geq \epsilon\right) &\leq \frac{1}{\epsilon^2} \lim_{k\to\infty} \sum_{n=k}^{\infty} \zeta(n)^2 \mathbb{E}\left\|\xi_n\right\|^2 \\ &\leq \frac{C_3}{\epsilon^2} \lim_{k\to\infty} \sum_{n=k}^{\infty} \zeta(n)^2 = 0,\end{aligned}$$

where the final inequality uses $\mathbb{E}\left\|\xi_n\right\|^2 \leq C_3 < \infty$, which can be inferred from the second condition in A5.2, while the final equality follows from the square summability of the step-size $\zeta(n)$, which is assumed in A5.4. Thus, A4.4 is satisfied.

- $\mathcal{Z}_\lambda$ is an asymptotically stable attractor for the ODE (5.5), with $G(\theta)$ itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{dG(\theta)}{dt} = \nabla G(\theta)\dot{\theta} = \nabla G(\theta)\check{\Gamma}(-\nabla G(\theta)) < 0, \ \forall \theta \notin \mathcal{Z}_\lambda.$$

The claim now follows by an application of Theorem 4.3.

For the case when A5.3 and A5.5 are used in place of A5.2, and A5.4, the proof again follows by verifying conditions A4.1–A4.4 of Theorem 4.3. Of these, A4.1 and A4.3 hold as shown above. The conditions A4.2 and A4.4 require a few deviations from the proof above, and we provide the details below. We rewrite the θ-recursion as follows:

$$\theta_{n+1} = \Gamma\Big(\theta_n + \zeta(n)\Big(\nabla G(\theta_n, \lambda) + \xi_{1,n} + \xi_{2,n}\Big)\Big),$$

where

$$\begin{aligned}\xi_{1,n} &= \mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right) - \nabla G(\theta_n),\\ \xi_{2,n} &= \widehat{\nabla} G(\theta_n) - \mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right).\end{aligned}$$

- From A5.3, we have that $\xi_{1,n} = O(\delta_n^2)$. Using A5.5, we have $\xi_{1,n} \to 0$ as $n \to \infty$. This verifies A4.2.

- For verifying A4.4, first note that $\mathbb{E}\left\|\xi_{2,n}\right\|^2 \leq C_4 < \infty$ using the last condition in A5.10. As before, applying Doob's martingale inequality and using square-summability of step-sizes $\{\zeta(n)\}$, we obtain

$$\begin{aligned}\lim_{k\to\infty} \mathbb{P}\left(\sup_{l\geq k}\left\|\sum_{n=k}^{l} \zeta(n)\xi_{2,n}\right\| \geq \epsilon\right) &\leq \frac{1}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\zeta(n)^2\mathbb{E}\left\|\xi_{2,n}\right\|^2\\ &\leq \frac{C_4}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\frac{\zeta(n)^2}{\delta_n^2} = 0.\end{aligned}$$

Thus A4.4 holds.

Hence, the claim follows for the case with assumptions A5.1, A5.3, and A5.5. □

Interestingly, one can derive a non-asymptotic bound for the policy gradient algorithm in a setting where optimizing a risk measure is the objective. The asymptotic result in the theorem above guarantees convergence to a point θ^* where the gradient of the objective vanishes. The non-asymptotic bound that we present next establishes a non-asymptotic bound on the norm of the gradient $\mathbb{E}\|\nabla G(\theta_R)\|^2$, where θ_R is a point picked uniformly at random from the set $\{\theta_1, \ldots, \theta_n\}$. Notice the resemblance of this scheme of picking a random iterate uniformly to the well-known Polyak-Ruppert averaging scheme for stochastic approximation. In the latter, the averaging is explicit, while the former achieves the same in expectation. However, we note that the non-asymptotic bounds presented below are for a constant step-size, while the Polyak-Ruppert averaging scheme uses diminishing step-sizes of the form $\frac{1}{k^\alpha}$, for some $\alpha \in (1/2, 1)$.

Recall that the update rule of the algorithm in the unconstrained involved a projection operator, which is not needed for the non-asymptotic bound that we present below. In other words, we shall analyze the following projection-free variant of the risk-sensitive policy gradient algorithm from a non-asymptotic viewpoint:

$$\theta_{n+1} = \theta_n - \zeta(n)\widehat{\nabla} G(\theta_n). \tag{5.8}$$

We require the following assumption for deriving the non-asymptotic bound:

A5.6. There exists a constant $L_1 > 0$ such that

$$\|\nabla G(\theta) - \nabla G(\theta')\| \leq L_1 \|\theta - \theta'\|, \quad \forall \theta, \theta' \in \mathbb{R}^d.$$

The smoothness requirement on the risk objective is not stringent, considering one would require such an assumption for obtaining the gradient estimates in A5.2/A5.3.

The main result providing the non-asymptotic bound of the policy gradient algorithm in (5.1) is given below.

Theorem 5.2.
(i) Assume A5.2 and A5.6 hold. Set $\zeta(k) = \min\left\{\frac{1}{L_1}, \frac{1}{\sqrt{n}}\right\}$ for $k = 1, \ldots, n$. Let θ_R be chosen uniformly at random from $\{\theta_1, \ldots, \theta_n\}$, and let θ^* be a global optima of G. Then, for any $n \geq 1$,

$$\mathbb{E}\left\|\nabla G\left(\theta_R\right)\right\|^2 \leq \frac{2L_1\left(G(\theta_1) - G(\theta^*)\right)}{n} + \frac{\left[2\left(G(\theta_1) - G(\theta^*)\right) + L_1\sigma^2\right]}{\sqrt{n}}. \tag{5.9}$$

(ii) If instead A5.3 and A5.6 hold and $\|\nabla G\left(\theta\right)\|_1 \leq B$ for any $\theta \in \mathbb{R}^d$, then setting $\zeta(k) = \min\left\{\frac{1}{L_1}, \frac{1}{n^{2/3}}\right\}$ and $\delta_k = \frac{1}{n^{1/6}}$, for $k = 1, \ldots, n$,

$$\mathbb{E}\left\|\nabla G\left(\theta_R\right)\right\|^2 \leq \frac{2\left(G(\theta_1) - G(\theta^*)\right)}{n} \max\left\{L_1, n^{2/3}\right\} + \frac{4BC_1 + L_1C_2}{n^{1/3}} + \frac{L_1 d C_1^2}{n^{4/3}}, \tag{5.10}$$

where the constants C_1 and C_2 are given in assumption A5.3.

The non-asymptotic bound in the result above of the order $O\left(\frac{1}{\sqrt{n}}\right)$ for the case when unbiased gradient estimates are available. On the other hand, the corresponding bound for the case of biased gradient estimates is $O\left(\frac{1}{n^{1/3}}\right)$. The weaker rate is a result of the fact that the gradient estimates exhibit a bias variance tradeoff via the perturbation constant δ_n, i.e., decreasing δ_n leads to a gradient estimate that has lower bias, though this is at the cost of increasing variance.

Proof. We first prove part (i), where (5.9) provides a finite-time bound for the SA iterative update (5.8) using unbiased gradient estimates with bounded variance, i.e., gradient estimators satisfying A5.2.

Using A5.6, we have

$$\begin{aligned} G\left(\theta_{k+1}\right) &\leq G\left(\theta_k\right) + \left\langle \nabla G\left(\theta_k\right), \theta_{k+1} - \theta_k \right\rangle + \frac{L_1}{2}\left\|\theta_{k+1} - \theta_k\right\|^2 \\ &= G\left(\theta_k\right) - \zeta(k)\left\langle \nabla G\left(\theta_k\right), \widehat{\nabla} G(\theta_k) \right\rangle + \frac{L_1}{2}\zeta(k)^2\left\|\widehat{\nabla} G(\theta_k)\right\|^2. \end{aligned}$$

Note that the search point θ_k is a function of the history $\{\theta_m, m < k\}$, and is random. Let $\mathbb{E}_k$ denote the expectation w.r.t. the sigma field $\mathcal{F}_k = \sigma(\theta_m, m < k)$. Taking expectations with respect to $\mathbb{E}_k$ on both sides of above equation, and noting that from A5.2, we have $\mathbb{E}_k\left[\widehat{\nabla} G(\theta_k)\right] = \nabla G\left(\theta_k\right)$, and $\mathbb{E}_k\left[\left\|\widehat{\nabla} G(\theta_k)\right\|^2\right] \leq \|\nabla G(\theta_n)\|^2 + \sigma^2$, we obtain

$$\begin{aligned}\mathbb{E}_k[G\left(\theta_{k+1}\right)] &\leq G\left(\theta_k\right) - \zeta(k)\left\|\nabla G\left(\theta_k\right)\right\|^2 + \frac{L_1}{2}\zeta(k)^2\left[\left\|\nabla G\left(\theta_k\right)\right\|^2 + \sigma^2\right] \\ &= G\left(\theta_k\right) - \left(\zeta(k) - \frac{L_1}{2}\zeta(k)^2\right)\left\|\nabla G\left(\theta_k\right)\right\|^2 + \frac{L_1}{2}\zeta(k)^2\sigma^2.\end{aligned}$$

Rearranging the terms, we obtain

$$\begin{aligned}&\left(\zeta(k) - \frac{L_1}{2}\zeta(k)^2\right)\left\|\nabla G\left(\theta_k\right)\right\|^2 \leq G\left(\theta_k\right) - \mathbb{E}_k[G\left(\theta_{k+1}\right)] + \frac{L_1}{2}\zeta(k)^2\sigma^2 \\ &\implies \zeta(k)\left\|\nabla G\left(\theta_k\right)\right\|^2 \leq \frac{2\left[G\left(\theta_k\right) - \mathbb{E}_k[G\left(\theta_{k+1}\right)]\right]}{\left(2 - L_1\zeta(k)\right)} + \frac{L_1\zeta(k)^2\sigma^2}{\left(2 - L_1\zeta(k)\right)}.\end{aligned}$$

Now summing up the above inequality from $k = 1$ to n, we obtain

$$\begin{aligned}&\sum_{k=1}^{n}\zeta(k)\left\|\nabla G\left(\theta_k\right)\right\|^2 \\ &\leq 2\sum_{k=1}^{n}\frac{\left[G\left(\theta_k\right) - \mathbb{E}_k[G\left(\theta_{k+1}\right)]\right]}{\left(2 - L_1\zeta(k)\right)} + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2 - L_1\zeta(k)\right)}.\end{aligned}$$

Taking expectations on both sides of the equation above, we obtain

$$\begin{aligned}&\sum_{k=1}^{n}\zeta(k)\mathbb{E}_n\left\|\nabla G\left(\theta_k\right)\right\|^2 \\ &\leq 2\sum_{k=1}^{n}\frac{\left[\mathbb{E}_n\left[G\left(\theta_k\right)\right] - \mathbb{E}_n\left[G\left(\theta_{k+1}\right)\right]\right]}{\left(2 - L_1\zeta(k)\right)} + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2 - L_1\zeta(k)\right)} \\ &= 2\left[\frac{G\left(\theta_1\right)}{\left(2 - L_1\zeta_1\right)} - \sum_{k=2}^{n}\left(\frac{1}{\left(2 - L_1\zeta_{k-1}\right)} - \frac{1}{\left(2 - L_1\zeta(k)\right)}\right)\mathbb{E}_n\left[G\left(\theta_k\right)\right]\right. \\ &\quad \left. - \frac{\mathbb{E}_n\left[G\left(\theta_{n+1}\right)\right]}{\left(2 - L_1\zeta_n\right)}\right] + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2 - L_1\zeta(k)\right)}.\end{aligned}$$

Notice that since step sizes $\{\zeta(k)\}_{k\geq 1}$ are non-increasing, we have $\left(\frac{1}{(2-L_1\zeta_{k-1})} - \frac{1}{(2-L_1\zeta(k))}\right) \geq 0$ and using the fact that $\mathbb{E}_n\left[G\left(\theta_k\right)\right] \geq G(\theta^*)$, where θ^* is a local optima, we obtain

$$\begin{aligned}
&\sum_{k=1}^{n} \zeta(k)\mathbb{E}_n \left\|\nabla G\left(\theta_k\right)\right\|^2 \\
&\leq 2\left[\frac{G(\theta_1)}{(2-L_1\zeta_1)} - G(\theta^*)\sum_{k=2}^{n}\left(\frac{1}{(2-L_1\zeta_{k-1})} - \frac{1}{(2-L_1\zeta(k))}\right) - \frac{G(\theta^*)}{(2-L_1\zeta_n)}\right] \\
&\quad + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{(2-L_1\zeta(k))} \\
&= \frac{2\left(G(\theta_1)-G(\theta^*)\right)}{(2-L_1\zeta_1)} + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{(2-L_1\zeta(k))}.
\end{aligned}$$

Let R be a r.v. with the following mass function:

$$P_R(k) := \mathbb{P}\left(R=k\right) = \frac{\zeta(k)}{\sum_{k=1}^{n}\zeta(k)}, \quad k=1,\ldots,n.$$

It follows from the definition of P_R above that,

$$\mathbb{E}\left[\left\|\nabla G\left(\theta_R\right)\right\|^2\right] = \frac{\sum_{k=1}^{n}\zeta(k)\mathbb{E}_n\left\|\nabla G\left(\theta_k\right)\right\|^2}{\sum_{k=1}^{n}\zeta(k)}.$$

Thus, we conclude

$$\mathbb{E}\left[\left\|\nabla G\left(\theta_R\right)\right\|^2\right] \leq \frac{1}{\sum_{k=1}^{n}\zeta(k)}\left[\frac{2\left(G(\theta_1)-G(\theta^*)\right)}{(2-L_1\zeta_1)} + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{(2-L_1\zeta(k))}\right]. \tag{5.11}$$

The bound in the equation above holds for a general step-size choice. Specializing the bound in (5.11) to the case of a constant step size $\zeta(k) = \left\{\zeta = \min\left\{\frac{1}{L_1}, \frac{1}{\sqrt{n}}\right\}\right\}, \forall k \geq 1$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla G\left(\theta_R\right)\right\|^2\right] &\leq \frac{1}{\sum_{k=1}^{n}\zeta(k)}\left[\frac{2\left(G(\theta_1)-G(\theta^*)\right)}{(2-L_1\zeta_1)} + L_1\sigma^2\sum_{k=1}^{n}\frac{\zeta(k)^2}{(2-L_1\zeta_k)}\right] \\
&= \frac{1}{n\zeta}\left[\frac{2\left(G(\theta_1)-G(\theta^*)\right)}{(2-L_1\zeta)} + L_1\sigma^2 n\frac{\zeta^2}{(2-L_1\zeta)}\right] \\
&\leq \frac{1}{n\zeta}\left[2\left(G(\theta_1)-G(\theta^*)\right) + L_1\sigma^2 n\zeta^2\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2\left(G(\theta_1) - G(\theta^*)\right)}{n\zeta} + L_1\sigma^2\zeta \\
&\leq \frac{2\left(G(\theta_1) - G(\theta^*)\right)}{n} \max\left\{L_1, \sqrt{n}\right\} + L_1\sigma^2\frac{1}{\sqrt{n}} \\
&\leq \frac{2L_1\left(G(\theta_1) - G(\theta^*)\right)}{n} + \frac{2\left(G(\theta_1) - G(\theta^*)\right)}{\sqrt{n}} + L_1\sigma^2\frac{1}{\sqrt{n}} \\
&= \frac{2L_1\left(G(\theta_1) - G(\theta^*)\right)}{n} + \frac{1}{\sqrt{n}}\left[2\left(G(\theta_1) - G(\theta^*)\right) + L_1\sigma^2\right],
\end{aligned}
$$

establishing (5.9) to complete the proof of part (i).

Proof of part (ii). Now we prove the second part of the theorem, specifically the finite-time bound (5.10) for the SA iterative update (5.8) using biased gradient estimators satisfying A5.3. As before, using A5.6, we have

$$
G\left(\theta_{k+1}\right) \leq G\left(\theta_k\right) - \zeta(k)\left\langle \nabla G\left(\theta_k\right), \widehat{\nabla} G(\theta_k)\right\rangle + \frac{L_1}{2}\zeta(k)^2\left\|\widehat{\nabla} G(\theta_k)\right\|^2. \tag{5.12}
$$

Taking expectations w.r.t. $\mathbb{E}_k$ on both sides of (5.12), followed by an application of the inequalities in A5.3, we obtain

$$
\begin{aligned}
&\mathbb{E}_k\left[G\left(\theta_{k+1}\right)\right] \\
&\leq \mathbb{E}_k\left[G\left(\theta_k\right)\right] - \zeta(k)\left\langle \nabla G\left(\theta_k\right), \nabla G\left(\theta_k\right) + C_1\delta_k^2\mathbf{1}_{d\times 1}\right\rangle \\
&\quad + \frac{L_1}{2}\zeta(k)^2\left[\left\|\mathbb{E}_k\left[\widehat{\nabla} G(\theta_k)\right]\right\|^2 + \frac{C_2}{\delta_k^2}\right] \\
&\leq G\left(\theta_k\right) - \zeta(k)\left\|\nabla G\left(\theta_k\right)\right\|^2 + C_1\delta_k^2\zeta(k)\mathbb{E}_k\|\nabla G\left(\theta_k\right)\|_1 \\
&\quad + \frac{L_1}{2}\zeta(k)^2\left[\left\|\nabla G\left(\theta_k\right)\right\|^2 + 2C_1\delta_k^2\mathbb{E}_k\|\nabla G\left(\theta_k\right)\|_1 + dC_1^2\delta_k^4 + \frac{C_2}{\delta_k^2}\right] \qquad (5.13) \\
&\leq G\left(\theta_k\right) - \left(\zeta(k) - \frac{L_1}{2}\zeta(k)^2\right)\left\|\nabla G\left(\theta_k\right)\right\|^2 + C_1\delta_k^2 B\left(\zeta(k) + L_1\zeta(k)^2\right) \\
&\quad + \frac{L_1}{2}\zeta(k)^2\left[dC_1^2\delta_k^4 + \frac{C_2}{\delta_k^2}\right], \qquad (5.14)
\end{aligned}
$$

where the inequality in (5.13) uses $-\|\theta\|_1 \leq \sum_{i=1}^d \theta_i$ for any vector θ, while the final inequality uses $\|\nabla G\left(\theta_k\right)\|_1 \leq B$, which holds by an

assumption in the theorem statement. A straightforward rearrangement of the terms in (5.14) leads to

$$\zeta(k)\left\|\nabla G\left(\theta_k\right)\right\|^2 \leq \frac{2}{\left(2-L_1\zeta(k)\right)}\Big[G\left(\theta_k\right)-\mathbb{E}_k G\left(\theta_{k+1}\right) + C_1\delta_k^2\left(\zeta(k)+L_1\zeta(k)^2\right)B\Big] + \frac{L_1\zeta(k)^2}{\left(2-L_1\zeta(k)\right)}\left[dC_1^2\delta_k^4+\frac{C_2}{\delta_k^2}\right].$$

Now, summing up the inequality above for $k = 1$ to n, and taking expectations, we obtain

$$\begin{aligned}
&\sum_{k=1}^{n}\zeta(k)\mathbb{E}_n\left\|\nabla G\left(\theta_k\right)\right\|^2\\
&\leq 2\sum_{k=1}^{n}\frac{\left(\mathbb{E}_n G\left(\theta_k\right)-\mathbb{E}_n G\left(\theta_{k+1}\right)\right)}{\left(2-L_1\zeta(k)\right)}+2\sum_{k=1}^{n}C_1\delta_k^2 B\left(\frac{\zeta(k)+L_1\zeta(k)^2}{2-L_1\zeta(k)}\right)\\
&\quad+L_1\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2-L_1\zeta(k)\right)}\left[dC_1^2\delta_k^4+\frac{C_2}{\delta_k^2}\right]\\
&=2\left[\frac{G\left(\theta_1\right)}{\left(2-L_1\zeta(1)\right)}-\sum_{k=2}^{n}\left(\frac{\mathbb{E}_n G\left(\theta_k\right)}{\left(2-L_1\zeta(k-1)\right)}-\frac{\mathbb{E}_n G\left(\theta_k\right)}{\left(2-L_1\zeta(k)\right)}\right)-\frac{\mathbb{E}_n\left[G\left(\theta_{n+1}\right)\right]}{\left(2-L_1\zeta(n)\right)}\right]\\
&\quad+2\sum_{k=1}^{n}C_1\delta_k^2 B\left(\frac{\zeta(k)+L_1\zeta(k)^2}{2-L_1\zeta(k)}\right)+L_1\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2-L_1\zeta(k)\right)}\left[dC_1^2\delta_k^4+\frac{C_2}{\delta_k^2}\right].
\end{aligned}$$

Using $\mathbb{E}_n\left[G\left(\theta_k\right)\right] \geq G(\theta^*)$, and $\left(\frac{1}{(2-L_1\zeta(k-1))}-\frac{1}{(2-L_1\zeta(k))}\right) \geq 0$, we obtain

$$\begin{aligned}
&\sum_{k=1}^{n}\zeta(k)\mathbb{E}_n\left\|\nabla G\left(\theta_k\right)\right\|^2 \leq \frac{2\left(\left(G(\theta_1)-G(\theta^*)\right)\right)}{\left(2-L_1\zeta(1)\right)}\\
&+2\sum_{k=1}^{n}C_1\delta_k^2 B\left(\frac{\zeta(k)+L_1\zeta(k)^2}{2-L_1\zeta(k)}\right)+L_1\sum_{k=1}^{n}\frac{\zeta(k)^2}{\left(2-L_1\zeta(k)\right)}\left[dC_1^2\delta_k^4+\frac{C_2}{\delta_k^2}\right].
\end{aligned}$$

Using the fact that $\mathbb{P}\left(R=i\right)=1/n, i=1,\ldots,n$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla G\left(\theta_R\right)\right\|^2\right] \leq \frac{1}{\sum_{k=1}^{n}\zeta(k)}\left[\frac{2\left(G(\theta_1)-G(\theta^*)\right)}{\left(2-L_1\zeta(1)\right)}+\right.\\
&\left.2B\sum_{k=1}^{n}C_1\delta_k^2\left[\frac{\zeta(k)+L_1\zeta(k)^2}{2-L_1\zeta(k)}\right]+\sum_{k=1}^{n}\frac{L_1\zeta(k)^2}{\left(2-L_1\zeta(k)\right)}\left[dC_1^2\delta_k^4+\frac{C_2}{\delta_k^2}\right]\right].
\end{aligned} \tag{5.15}$$

We now specialize the result obtained in the equation above, to derive the bound in (5.10). Using $\zeta(k) \triangleq \left\{\zeta = \min\left\{\frac{1}{L_1}, \frac{1}{n^{2/3}}\right\}\right\}$, and $\delta_k \triangleq \left\{\delta = \frac{1}{n^{1/6}}\right\}$ in (5.15), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\|\nabla G(\theta_R)\|^2\right] \\
&\leq \frac{1}{n\zeta}\left[2\left(G(\theta_1) - G(\theta^*)\right) + 4n\zeta BC_1\delta^2 + L_1 n\zeta^2\left[dC_1^2\delta^4 + \frac{C_2}{\delta^2}\right]\right] \\
&\leq \frac{2\left(G(\theta_1) - G(\theta^*)\right)}{n}\max\left\{L_1, n^{2/3}\right\} + \frac{4BC_1}{n^{1/3}} + \frac{L_1}{n^{2/3}}\left[\frac{dC_1^2}{n^{2/3}} + \frac{C_2}{n^{-1/3}}\right],
\end{aligned}$$

where the first inequality above follows by using the fact that $\zeta \leq 1/L_1$, while the final inequality follows by using the definition of ζ and δ. The final bound in (5.10) then follows by rearranging terms. □

5.4 Convergence analysis in the setting with risk as constraint

In this section, we analyze the convergence properties of the two-timescale SA algorithm given by (5.3)–(5.4) in Section 5.2. The convergence analysis using the ODE approach on the two timescales in (5.3) and (5.4) is based on the following intuition, to be made rigorous: the faster timescale recursion in (5.4) sees the iterate λ_n on the slower timescale as quasi-static, while the slower timescale recursion in (5.3) sees the iterate θ_n on the faster timescale as equilibrated. In essence, this viewpoint is equivalent to assuming that slower timescale iterate as constant while analyzing the faster timescale recursion, and using converged values of the faster timescale iterate for analysis of the slower timescale recursion.

We make the following assumptions for the convergence analysis of the two-timescale SA recursions given by (5.3) and (5.4). Again, let $\mathcal{F}_n = \sigma(\theta_m, m \leq n)$ denote the underlying σ-field.

A5.7. The policy $\mu^\theta(\cdot|x)$ is a continuously differentiable function of θ, for any $x \in \mathcal{X}$ and $a \in \mathcal{A}$. Furthermore, the risk measure G is a continuously differentiable function of the policy parameter θ.

A5.8. The risk-measure estimator $\widehat{G}(\cdot)$ satisfies $\mathbb{E}\left(\widehat{G}(\theta_n) \mid \mathcal{F}_n\right) = G(\theta_n)$

A5.9. The gradient estimators $\widehat{\nabla} J(\theta_n)$ and $\widehat{\nabla} G(\theta_n)$ satisfy

$$\mathbb{E}\left(\widehat{\nabla} J(\theta_n) \mid \mathcal{F}_n\right) = \nabla J(\theta_n), \mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right) = \nabla G(\theta_n),$$
$$\text{and } \mathbb{E}\left\|\widehat{\nabla} J(\theta_n)\right\|^2 + \mathbb{E}\left\|\widehat{\nabla} G(\theta_n)\right\|^2 < \infty.$$

A5.10. The gradient estimators $\widehat{\nabla} J(\theta_n)$ and $\widehat{\nabla} G(\theta_n)$ with perturbation $\delta_n > 0$ satisfy

$$\left\|\mathbb{E}\left(\widehat{\nabla} J(\theta_n) \mid \mathcal{F}_n\right) - \nabla J(\theta_n)\right\| = C_5\ \delta_n^2,$$
$$\left\|\mathbb{E}\left(\widehat{\nabla} G(\theta_n) \mid \mathcal{F}_n\right) - \nabla G(\theta_n)\right\| = C_6\ \delta_n^2,$$
$$\text{and } \left(\mathbb{E}\left\|\widehat{\nabla} J(\theta_n)\right\|^2 + \mathbb{E}\left\|\widehat{\nabla} G(\theta_n)\right\|^2\right) < \frac{C_7}{\delta_n^2},$$

where C_5, C_6, and C_7 are dimension-dependent constants.

A5.11. The step-size sequences $\{\zeta_1(n), \zeta_2(n)\}$ satisfy

$$\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty,\ \sum_n (\zeta_1(n)^2 + \zeta_2(n)^2)\ < \infty,$$
$$\zeta_1(n) = o(\zeta_2(n)).$$

A5.12. The gradient estimator perturbation $\delta_n \to 0$ as $n \to \infty$, and the step-size sequences $\{\zeta_1(n), \zeta_2(n)\}$ satisfy

$$\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty,\ \sum_n \left[\zeta_1(n)^2 + \left[\frac{\zeta_2(n)}{\delta_n}\right]^2\right] < \infty,$$
$$\zeta_1(n) = o(\zeta_2(n)).$$

We now discuss the assumptions. The first part in assumption A5.7 is a standard requirement in the analysis of policy gradient-type RL algorithms. The second part imposes a smoothness requirement on the risk measure G, when viewed as a function of the policy parameter θ. As mentioned earlier, for an abstract risk measure G, one cannot infer smoothness of G based only on the policy parameterization being smooth. This smoothness condition together with the fact that J is

smooth ensures that the ODE underlying the θ-recursion is well-posed. Assumption A5.8 is an unbiasedness requirement on the risk measure estimate that is used for dual ascent in (5.3). Assumptions A5.9 and A5.10 are unbiasedness/asymptotic unbiasedness requirements on the estimators of the gradient of the objective and risk measure, and are necessary to ensure that the θ-recursion in (5.4) is descending in the Lagrangian objective. Assumption A5.9 requires that the estimators of the gradient of J and G be unbiased, and also that the variance of the these gradient estimates is bounded. Such requirements are common in the analysis of stochastic gradient schemes, and estimators formed using the likelihood ratio method (described in the next section) usually satisfy A5.9. Assumption A5.10 is a relaxed version of A5.9, where the gradient estimators are asymptotically unbiased, i.e., the gradient estimators have a perturbation parameter δ_n, which can be used to control the bias-variance tradeoff. The SPSA technique for gradient estimation, presented in Section 4.4, as well as the general class of simultaneous perturbation schemes, satisfy the conditions in A5.10. For convergence, δ_n has to vanish asymptotically, but not too fast, as outlined in the second part of A5.12.

The first two conditions on the step-sizes $\{\zeta_1(n), \zeta_2(n)\}$ in A5.11 are standard stochastic approximation requirements. The condition $\zeta_1(n) = o(\zeta_2(n))$ is required for the standard two-timescale view, i.e., the policy recursion in (5.4) views the Lagrange multiplier as quasi-static, while the Lagrange multiplier recursion views the policy parameter as almost equilibrated. This view is made precise in the convergence analysis presented in Section 5.4 below. Two-timescale updates are convenient because both policy and Lagrange multiplier can be updated in parallel, albeit with varying step-sizes. The latter are chosen carefully so that one is able to mimic a two-loop behavior, with policy updates in the inner loop and Lagrange multiplier updates in the outer loop. Assumption A5.12 is a variant of A5.11, which when coupled with (A3') ensures that the bias $O(\delta_n^2)$ of gradient estimates vanish asymptotically. Further, the noise in gradient estimates can be ignored in the asymptotic analysis owing to the condition $\sum_n \left(\frac{\zeta_2(n)}{\delta_n}\right)^2 < \infty$.

We adopt the ODE approach for analyzing the template algorithm in (5.3)–(5.4). In particular, under the assumptions listed above, the ODE governing the policy update, for any given Lagrange multiplier λ, is given by

$$\dot{\theta}(t) = \check{\Gamma}\left(-\nabla J(\theta(t)) - \lambda \nabla G(\theta(t))\right), \tag{5.16}$$

where $\check{\Gamma}(\cdot)$ is a projection operator that ensures the evolution of θ via the ODE (5.16) stays within the set Θ and is defined as in (5.6).

Remark 5.1. (Two-timescale view) In describing the ODE governing the policy recursion, we have assumed that the Lagrange multiplier is constant, and this view can be justified as follows: First rewrite the λ-recursion as

$$\lambda_{n+1} = \left[\lambda_n + \zeta_2(n)\left(\frac{\zeta_1(n)}{\zeta_2(n)}\left(G(\theta_n) - \kappa + \varsigma_{1,n}\right)\right)\right]^+,$$

where $\varsigma_{1,n}$ is a martingale difference sequence (a consequence of (A1)). Considering that we have a finite-dimensional MDP setting, together with the fact that $\frac{\zeta_1(n)}{\zeta_2(n)} = o(1)$ (see (A4)), it is clear that the λ-recursion above tracks the ODE $\dot{\lambda}(t) = 0$.

The claim that the λ-recursion views the policy parameter as almost equilibrated requires a more sophisticated argument, and we provide a proof sketch below. We first rewrite the λ update iteration as follows:

$$\lambda_{n+1} = \left[\lambda_n + \zeta_1(n)\left(G(\theta_{\lambda_n}) - \kappa + \varsigma_{2,n}\right)\right]^+,$$

where $\varsigma_{2,n} = G(\theta_n) - G(\theta_{\lambda_n})$. The noise factor $\varsigma_{2,n}$ is defined using the fast timescale parameter θ_{λ_n} with the slow timescale iterate λ_n. The parameter θ_{λ_n} is a limiting point of the θ-recursion, with the Lagrange multiplier λ_n. Owing to the convergence of θ-recursion, one can infer that $\varsigma_{2,n} = o(1)$, i.e., $\varsigma_{2,n}$ adds an asymptotically vanishing bias term to the λ-recursion above. Thus, it is apparent that the λ-recursion views the policy parameter as almost equilibrated, and the technical proof proceeds by showing that the λ-recursion tracks the following ODE:

$$\dot{\lambda}(t) \;=\; \check{\Gamma}_\lambda\left[G(\theta_{\lambda(t)}) - \kappa\right],$$

where $\check{\Gamma}_\lambda$ is a projection operator that is defined as follows: For any bounded continuous function $f(\cdot)$,

$$\check{\Gamma}_\lambda(f(\lambda)) = \lim_{\tau \to 0} \frac{(\lambda + \tau f(\lambda))^+ - \lambda}{\tau}. \tag{5.17}$$

The projection operator $\check{\Gamma}$ ensures that the λ-recursion stays within $[0, \infty)$. The tools used in establishing such a claim are classic for stochastic approximation schemes, e.g., Theorem 4.1 in the previous section.

5.4.1 Convergence of policy parameter

Theorem 5.3. Assume A5.7, A5.8, (A5.9 + A5.11) or (A5.10 + A5.12). If $\lambda_n = \lambda \ \forall n$, then for θ_n governed by (5.4),

$$\theta_n \to \mathcal{Z}_\lambda \text{ a.s. as } n \to \infty,$$

where $\mathcal{Z}_\lambda = \{\theta \in \Theta : \check{\Gamma}(-\nabla J(\theta(t)) - \lambda \nabla G(\theta(t))) = 0\}$ is the set of limit points of the ODE (5.16).

Proof. The proof again involves an application of the Kushner-Clark lemma for projected stochastic approximation, i.e., Theorem 4.3 in Section 4.1.3.

We first rewrite the recursion (5.4) as follows (with $\lambda_n = \lambda$):

$$\theta_{n+1} = \Gamma\Big(\theta_n - \zeta_2(n)\Big(\nabla L(\theta_n, \lambda) + \xi_n\Big)\Big),$$

where

$$\begin{aligned} L(\theta, \lambda) &= J(\theta) + \lambda(G(\theta) - \kappa), \\ \xi_n &= \widehat{\nabla} J(\theta_n) + \lambda \widehat{\nabla} G(\theta_n) - \nabla L(\theta_n, \lambda). \end{aligned}$$

We now verify conditions A4.1–A4.4 of Theorem 4.3 for (5.4.1) in the case where we assume A5.7, A5.8, A5.9, and A5.11. Later we provide the deviations necessary to handle the case when A5.10 and A5.12 are used in place of A5.9 and A5.11, respectively.

- From A5.7, we have that the policy μ^θ is a continuously differentiable function of θ, which implies the value function J is

continuously differentiable in θ, as well. This fact combined with the second part of A5.7, which imposed a smoothness requirement on the risk measure G, imply that the condition A4.1 follows for $\nabla L(\theta_n, \lambda)$.

- Since A5.9 implies unbiased gradient estimators, the assumption A4.2 trivially holds.
- A5.11 implies A4.3.
- Using the third condition in A5.9, it is easy to infer that $\mathbb{E}\left\|\xi_n\right\|^2 \leq C_8 < \infty$. Using this fact in conjunction with Doob's martingale inequality stated earlier, we obtain

$$\begin{aligned}\lim_{k\to\infty} \mathbb{P}\left(\sup_{l\geq k}\left\|\sum_{n=k}^{l}\zeta_2(n)\xi_n\right\| \geq \epsilon\right) \leq & \frac{1}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\zeta_2(n)^2\mathbb{E}\left\|\xi_n\right\|^2 \\ \leq & \frac{C_8}{\epsilon^2}\lim_{k\to\infty}\sum_{n=k}^{\infty}\zeta_2(n)^2 \to 0,\end{aligned}$$

  where the last limit follows from the square summability of the step-size $\zeta_2(n)$, which is assumed in A5.11. Thus, A4.4 is satisfied.
- $\mathcal{Z}_\lambda$ is an asymptotically stable attractor for the ODE (5.16), with $L(\theta, \lambda)$ itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{dL(\theta,\lambda)}{dt} = \nabla L(\theta,\lambda)\dot{\theta} = \nabla L(\theta,\lambda)\check{\Gamma}\big(-\nabla L(\theta,\lambda)\big) < 0,\ \forall \theta \notin \mathcal{Z}_\lambda.$$

The claim now follows by an application of Theorem 4.3. □

For the case when A5.10 and A5.12 are used in place of A5.9 and A5.11, the proof again follows by verifying conditions A4.1–A4.4 of Theorem 4.3. Of these, A4.1 and A4.3 hold as shown above. The conditions A4.2 and A4.4 require a few deviations from the proof above, and we provide the details below.

First, we rewrite the θ-recursion as follows:

$$\theta_{n+1} = \Gamma\Big(\theta_n - \zeta_2(n)\Big(\nabla L(\theta_n,\lambda) + \xi_{1,n} + \xi_{2,n}\Big)\Big),$$

where

$$\begin{aligned}\xi_{1,n} =& \mathbb{E}\left(\widehat{\nabla} J(\theta_n) + \lambda \widehat{\nabla} G(\theta_n) \mid \theta_n\right) - \nabla L(\theta_n, \lambda),\\ \xi_{2,n} =& \widehat{\nabla} J(\theta_n) + \lambda \widehat{\nabla} G(\theta_n) - \mathbb{E}\left(\widehat{\nabla} J(\theta_n) + \lambda \widehat{\nabla} G(\theta_n) \mid \theta_n\right).\end{aligned}$$

- From A5.10, we have $\xi_{1,n} = O(\delta_n^2)$. Using A5.12, we have $\xi_{1,n} \to 0$ as $n \to \infty$. This verifies A4.2.
- For verifying A4.4, first note that $\mathbb{E}\left\|\xi_{2,n}\right\|^2 \leq C_9 < \infty$ using the last condition in A5.10. As before, applying Doob's martingale inequality and using square-summability of step-sizes $\{\zeta_2(n)\}$,

$$\begin{aligned}\lim_{k\to\infty} \mathbb{P}\left(\sup_{l\geq k} \left\|\sum_{n=k}^{l} \zeta_2(n)\xi_{2,n}\right\| \geq \epsilon\right) \leq& \frac{1}{\epsilon^2} \lim_{k\to\infty} \sum_{n=k}^{\infty} \zeta_2(n)^2 \mathbb{E}\left\|\xi_{2,n}\right\|^2\\ \leq& \frac{C_9}{\epsilon^2} \lim_{k\to\infty} \sum_{n=k}^{\infty} \frac{\zeta_2(n)^2}{\delta_n^2} \to 0.\end{aligned}$$

  Thus A4.4 holds.

Hence, the claim follows for the case with assumptions A5.7, A5.8, A5.10, and A5.12. □

Remark 5.2. The policy update (5.4) might not converge to a local minimum, and instead get trapped in undesirable equilibria. As discussed earlier in Section 4.1, stochastic (i.e., inherently noisy) gradient estimators that drive the policy update may ensure avoidance of traps. An alternative is to add extraneous noise, as in (4.6).

5.4.2 Convergence of Lagrange multiplier

We now turn to the analysis of λ-recursion in (5.3). The ODE underlying the Lagrange multiplier is given below:

$$\dot{\lambda}(t) \;=\; \check{\Gamma}_\lambda[G(\theta_{\lambda(t)}) - \kappa], \tag{5.18}$$

where $\theta_{\lambda(t)}$ is the converged value of the θ-recursion, when the Lagrange multiplier is set to $\lambda(t)$, and the operator $\check{\Gamma}_\lambda$ is defined in (5.17).

Theorem 5.4. Assume A5.7, A5.8, (A5.9 + A5.11) or (A5.10 + A5.12). Then, for λ_n governed by (5.3),

$$\lambda_n \to \mathcal{Z} \text{ a.s. as } n \to \infty,$$

where $\mathcal{Z} = \left\{\lambda \in [0,\infty) : \check{\Gamma}_\lambda\left(G(\theta_\lambda) - \kappa\right) = 0, \theta_\lambda \in \mathcal{Z}_\lambda\right\}$ is the set of limit points of the ODE (5.18) and $\mathcal{Z}_\lambda$ is defined in Theorem 5.3.

Proof. Let $H(\lambda) = \min_{\theta\in\Theta} L(\theta,\lambda)$. Note that the function H is a pointwise infimum of a family of affine functions of λ, hence concave. Thus it is differentiable except at at most countably many points, where its has right and left derivatives. Furthermore, the right derivative at a point does not exceed the left derivative and both right and let derivatives are monotone decreasing. For simplicity of exposition, we assume below that H is differentiable everywhere, the argument can easily be adapted to the more general case by using super-gradients where needed. Also, H attains its maximum at a unique point, viz., the Lagrange multiplier λ^*. It follows that $H(\lambda) \downarrow -\infty$ as $\lambda \to \pm\infty$. In fact, we have some $C, c > 0$ such that

$$H'(\lambda) < -c\lambda, \forall \lambda \geq C, \tag{5.19}$$

$$H'(\lambda) > c\lambda, \forall \lambda \leq -C. \tag{5.20}$$

The Lagrange multiplier iterate λ_n tracks the following ODE:

$$\dot{\lambda}(t) = H'(\lambda(t)).$$

In view of (5.19)–(5.20), a straightforward adaptation of the arguments of Theorem 4.4 ensure the a.s. stability of the iterates.
Given that the λ iteration is stable, the rest of the proof follows by using arguments similar to those employed in proving convergence of standard stochastic approximation algorithms, and we omit the details. □

So far, we have shown that (θ_n, λ_n) converges to $(\theta_{\lambda^*}, \lambda^*)$, for some λ^* satisfying $\check{\Gamma}_\lambda\left(G(\theta_\lambda) - \kappa\right) = 0$, and $\theta_{\lambda^*} \in \mathcal{Z}_{\lambda^*}$. For a given λ, the condition $\check{\Gamma}_\lambda(G(\theta_\lambda) - \kappa) = 0$ is the same as $\check{\Gamma}_\lambda(\nabla_\lambda L(\theta_\lambda, \lambda)) = 0$.

We now state a result that helps us understand if the limit $(\theta_{\lambda^*}, \lambda^*)$ of the tuple (θ_n, λ_n) is a local saddle point of the Lagrangian, and if θ_{λ^*} satisfies the risk constraint.

Theorem 5.5. Let $H(\lambda) = \min_{\theta\in\Theta} L(\theta, \lambda)$. The ODE underlying the λ-recursion in (5.18) is the same as

$$\dot{\lambda}(t) \quad = \quad \check{\Gamma}_\lambda[\nabla_\lambda H(\lambda(t))], \tag{5.21}$$

where the latter ODE is to be interpreted in the 'Caratheodory' sense, i.e.,

$$\lambda(t) = \lambda(0) + \int_0^t \check{\Gamma}_\lambda[\nabla_\lambda H(\lambda(s))]ds, t \geq 0. \tag{5.22}$$

Thus, the iterate λ_n governed by (5.3) converges to a local minima of H.

Proof. (Sketch) For inferring the main claim in the result above, one invokes the envelope theorem of mathematical economics. In particular, using this theorem, it follows that at every point $\lambda(t)$ where the function $\tilde{H} = \check{\Gamma}_\lambda H(\cdot)$ is differentiable, the RHS of the ODE (5.21) coincides with that in the integral equation (5.22). At points where the function $\tilde{H}$ is not differentiable, it can be argued that the ODE spends zero time, provided they are not the global minima of $\tilde{H}$. □

Remark 5.3. Notice that the limiting policy θ_{λ^*} corresponding to $\lambda^* \in \mathcal{Z}$ satisfies the risk constraint $G(\theta_{\lambda^*}) \leq \kappa$, since λ^* corresponds to the equilibrium of the ODE $\dot{\lambda}(t) = \nabla_\lambda H(\lambda(t))$ that is constrained to remain in $[0, \infty)$, implying $\nabla_\lambda H(\lambda^*) = 0$.

To summarize, the two-timescale risk-sensitive policy gradient algorithm with iterate (θ_n, λ_n) converges to a local saddle point of the Lagrangian $L(\cdot, \cdot)$, i.e., to a point that is a local minimum w.r.t. θ and a local maximum w.r.t. λ. Moreover, the limit is a policy that satisfies the risk constraint.

5.4.3 Projection of Lagrange multiplier onto a finite interval

In practice, one may want to project λ iterate in (5.3) onto a finite interval, say $[0, \lambda_{\max}]$, i.e., the following update iteration:

$$\lambda_{n+1} = \Gamma_\lambda \left[\lambda_n + \zeta_1(n) \left(\widehat{G}(\theta_n) - \kappa\right)\right],$$

where Γ_λ denotes the projection operator onto $[0, \lambda_{\max}]$.

The analysis of this projected λ iteration would be similar to the case analyzed before. In particular, a variant of Theorem 5.4 can be claimed easily, without a detailed argument for stability of iterates owing to the projection onto a finite interval. However, the observation in Remark 5.3 regarding the risk constraint is not true in the case when there is projection. In other words, the λ-recursion in (5.3) involves projection on to the interval $[0, \lambda_{\max}]$, and this recursion tracks the ODE given in (5.21).

However, it is possible to establish the following claims concerning the limit $(\theta_{\lambda^*}, \lambda^*)$.

Theorem 5.6. Consider the limit set $\mathcal{Z}$ defined in Theorem 5.4.

(i) For the following "truncated" version of $\mathcal{Z}$,

$$\widehat{\mathcal{Z}} = \left\{ \lambda \in [0, \lambda_{\max}) : \check{\Gamma}_\lambda \left(G(\theta_\lambda) - \kappa \right) = 0, \theta_\lambda \in \mathcal{Z}_\lambda \right\},$$

the policy $\theta_{\hat{\lambda}}$ corresponding to a $\hat{\lambda} \in \widehat{\mathcal{Z}}$ satisfies the risk constraint $G(\theta_{\hat{\lambda}}) \leq \kappa$.

(ii) For $\lambda^* \in \mathcal{Z}$, if $G(\theta_{\lambda^*}) < \kappa$, then $\lambda^* = 0$, and $L(\theta_{\lambda^*}, \lambda^*) = J(\theta_{\lambda^*})$. Thus, the risk-sensitive policy gradient algorithm converges to a local minimum of J while satisfying the risk constraint.

(iii) Call $\lambda^* \in \mathcal{Z}$ a spurious stationary point of (5.18) if λ^* is not a stationary point of the ODE $\dot{\lambda}(t) = G(\theta_{\lambda(t)}) - \kappa$. For $\lambda^* \in \mathcal{Z}$, if $G(\theta_{\lambda^*}) > \kappa$, then $\lambda^* = \lambda_{\max}$, and such a λ^* corresponds to a spurious stationary point.

Proof. We prove the claim in (i) by contradiction. Assume that $G(\theta_{\hat{\lambda}}) > \kappa$ for $\hat{\lambda} \in \widehat{\mathcal{Z}}$. Then, we have

$$\check{\Gamma}_\lambda(G(\theta_{\hat{\lambda}}) - \kappa) = \lim_{\eta \to 0} \frac{\Gamma_\lambda(\hat{\lambda} + \eta(G(\theta_{\hat{\lambda}}) - \kappa)) - \hat{\lambda}}{\eta} = G(\theta_{\hat{\lambda}}) - \kappa > 0, \tag{5.23}$$

which leads to a contradiction since $\check{\Gamma}_\lambda(G(\theta_{\hat{\lambda}}) - \kappa) = 0$ as $\hat{\lambda} \in \widehat{\mathcal{Z}}$. The equality in (5.23) follows by using the facts that $\hat{\lambda} \geq 0$, and $G(\theta_{\hat{\lambda}}) > \kappa$ to infer that for small enough $\eta > 0$,

$$\Gamma_\lambda(\hat{\lambda} + \eta(G(\theta_{\hat{\lambda}}) - \kappa)) = \hat{\lambda} + \eta(G(\theta_{\hat{\lambda}}) - \kappa).$$

We now proceed to prove the claim in (ii). In the case where $\lambda^* = 0$, we have $\Gamma_\lambda(\lambda^* + \eta(G(\theta_{\lambda^*}) - \kappa)) = 0$ for any $\eta > 0$, since $G(\theta_{\lambda^*}) < \kappa$.

Next, the case of $\lambda^* > 0$ is not possible, and this can be argued as follows: Suppose that $\lambda^* > 0$. Then, for small enough $\eta > 0$

$$\Gamma_\lambda(\lambda^* + \eta(G(\theta_{\lambda^*}) - \kappa)) = \lambda^* + \eta(G(\theta_{\lambda^*}) - \kappa) > 0, \text{ implying}$$
$$\check{\Gamma}_\lambda(G(\theta_{\lambda^*}) - \kappa) < 0,$$

which leads to a contradiction since $\lambda^* \in \mathcal{Z}$.

For the final claim in (iii), observe that $\Gamma_\lambda(\lambda^* + \eta(G(\theta_{\lambda^*}) - \kappa)) = \lambda_{\max}$ leading to $\check{\Gamma}_\lambda(G(\theta_{\hat{\lambda}}) - \kappa) = 0$, since $\lambda^* + \eta(G(\theta_{\lambda^*}) - \kappa) > \lambda_{\max}$ for any $\eta > 0$. □

Thus, when one projects the Lagrange multiplier onto a finite interval, the convergence guarantee is to a local saddle point of the Lagrangian, as before. However, the limiting policy may not necessarily satisfy the risk constraint due to finite projection for the Lagrange multiplier. The latter can be avoided in practice by choosing a large enough value for $\lambda_{\max}$. However, the only way to theoretically guarantee avoidance of spurious limiting policies (i.e., ones that do not satisfy the risk constraint) is to eliminate the projection, i.e., to allow λ to be any positive real number.

5.5 Bibliographic remarks

For constrained MDPs, a textbook reference is Altman (1999). In Section 5.2, we invoke Theorem 3.8 from there to infer that there exists an optimal policy for the risk-constrained MDP in (5.2), whenever there is a policy that satisfies the risk constraint for this problem. A classic reference for the regularity conditions for ensuring the existence of a unique saddle point of the Lagrangian of the problem (5.2) is Sion (1958). Mean-variance optimization of MDPs has been shown to be NP-hard in Mannor and Tsitsiklis (2013).

Our template algorithm for the 'risk as constraint' setting incorporates two-timescale stochastic approximation (Borkar, 1997; Borkar, 2008). Two-timescale algorithms are popular for solving the problem of control in the context of reinforcement learning, where they are

usually referred to as actor-critic algorithms, cf. Konda and Borkar (1999), Borkar (2005), Bhatnagar *et al.* (2009), Bhatnagar (2010), and Prashanth and Ghavamzadeh (2016). Several simulation-based optimization algorithms also involve multiple timescales, see the textbook by Bhatnagar *et al.* (2013) for several examples. From the field of simulation optimization, the simultaneous perturbation method for gradient estimation is particularly relevant for solving risk-sensitive MDPs, when the underlying risk measure does not possess the structure to enable direct gradient estimation schemes such as the likelihood ratio method; see the case studies involving variance and CPT risk measures in Sections 6 and 7 for concrete examples.

Borkar (2008, Theorem 2 in Chapter 6) provides a justification of the standard two-timescale viewpoint, i.e., the λ-recursion on the slower timescale sees the policy parameter as almost equilibrated, while the θ-recursion on the faster timescale sees the Lagrange multiple λ as quasi-static. The Kushner-Clark Lemma (Kushner and Clark, 1978) is a classic result that can be invoked to establish asymptotic convergence of stochastic approximation schemes. The proof of Theorem 5.4 follows by using arguments similar to those employed in the proof of Theorem 2 in Chapter 2 of Borkar (2008). For the claim that the two-timescale algorithm in the constrained setting converges to a policy that satisfies the constraint in (5.2), one invokes the envelope theorem of mathematical economics (Mas-Colell *et al.*, 1995). The reader is referred to Borkar (2005) or Bhatnagar (2010) for further details.

The convergence guarantees in Section 5.4 for the template algorithm with risk as constraint are asymptotic in nature, and we have not provided any convergence rate results for this algorithm. Deriving such a rate result is challenging, as there are no rate rate results for general two-timescale stochastic approximation schemes, barring a few exceptions that we note next. In Konda and Tsitsiklis (2004), the authors handle the case of linear recursions, and provide an asymptotic rate result through a central limit theorem (CLT)-type result. In Mokkadem and Pelletier (2006), the authors extend this result to handle nonlinear recursions, and it is not clear if the assumptions for invoking this result are satisfied by the template algorithm that uses the update iterations (5.3)–(5.4). More recently, in Dalal *et al.* (2018) and Dalal *et al.* (2020),

the authors derive non-asymptotic bounds for two-timescale stochastic approximation, albeit with linear recursions.

The non-asymptotic analysis in the 'risk as objective' setting is based on the randomized stochastic gradient algorithm proposed in Ghadimi and Lan (2013). In particular, the proof of the non-asymptotic bound in Theorem 5.2 follows by a completely parallel argument to the proof of Theorem 2.1 in Ghadimi and Lan (2013).

The theoretical guarantees in this section establish convergence of the risk-sensitive policy gradient algorithms to a stationary point, a standard notion often employed in the analysis of policy gradient algorithms. As noted in Section 4.1, one can avoid saddle points/local maxima (minima for maximization problems) and ensure convergence to a local minimum (resp. maximum) if the gradient estimator has sufficient noise in all directions; see Pemantle (1990) and Brandiere and Duflo (1996) for a precise set of conditions, and a textbook reference for the topic of avoidance of such undesirable stationary point "traps" is Section 4.3 of Borkar (2008). A recent result in this direction is Barakat *et al.* (2021). If the gradient estimation noise is lacking, extraneous noise can be added in the policy gradient update, as in (4.6). Such an approach has been explored in a non-RL context in Ge *et al.* (2015) and Jin *et al.* (2017). Recent work in the policy gradient literature has tried to go beyond stationary convergence, e.g., using second-order information and/or incorporating variance reduction, e.g., Papini *et al.* (2018), Shen *et al.* (2019), and Zhang *et al.* (2020); however, providing guarantees beyond stationary convergence and characterizing the optima is beyond the scope of our monograph, as such considerations are not specific to a risk-sensitive context but apply to any policy gradient algorithm.

6

MDPs with Risk as the Constraint

Recall that the main ingredients in each iteration n of the risk-sensitive RL algorithm in the constrained setting are as follows:

(i) Simulation of the underlying MDP. For the case of non-perturbation-based approaches, such as the likelihood ratio method, one simulation with policy θ_n would suffice. On the other hand, for SPSA-based approaches, an additional simulation using a perturbed policy parameter would be necessary (see Section 4.4).

(ii) Estimation of $\nabla J(\theta_n)$ and $\nabla G(\theta_n)$. These estimates are fed into the primal descent update for θ_n.

(iii) Estimation of $G(\theta_n)$ using sample data. This estimate is used for dual ascent.

(iv) Estimation of $J(\theta_n)$ using sample data. This estimate is used for primal descent. Note that in the case of SPSA, we would require estimate of J for the perturbed policy as well, while additional function estimates are not necessary using the likelihood ratio method.

In the four special cases that we discuss in detail below, we shall address the items above in a variety of MDP contexts, under mean-variance, CVaR, and chance constraints. In particular, Table 6.1 presents the combinations for the objective J and risk measure G in (1.1).

Table 6.1: Risk-sensitive MDPs considered in this monograph.

	MDP type	objective J	constraint G
Case 1	discounted cost	cumulative cost (see (2.2))	overall variance (see (3.1))
Case 2	average cost	average cost (see (2.11))	per-period variance (see (3.3))
Case 3	SSP	total cost (see (2.6))	CVaR (see (3.4))
Case 4	discounted cost or SSP	total/discounted cost	chance (see (3.5))

Section 4 presented the necessary background material on TD algorithms and two widely used approaches for gradient estimation, which serve as building blocks for the four special cases in Table 6.1 described in the rest of this section. The presentation presumes the reader is familiar with the theory of risk-neutral RL. The references in the bibliographic remarks provide more details for the interested reader.

6.1 Case 1: Discounted-cost MDP + variance as risk

We consider the following constrained problem: For a given $\kappa > 0$ and initial state x_0 of the discounted-cost MDP,

$$\min_{\theta} J(\theta, x_0) \qquad \text{subject to} \qquad G(\theta, x_0) \leq \kappa,$$

where $J(\theta, x) = \mathbb{E}[D(\theta, x)]$ and $G(\theta, x) = U(\theta, x) - J(\theta, x)^2$ are the expectation and variance of the cumulative cost r.v., respectively, with $D(\theta, x)$ denoting the discounted total cost, which was defined in (2.1).

Gradient of the Lagrangian

Defining $L(\theta, \lambda) \triangleq J(\theta, x_0) + \lambda\big(G(\theta, x_0) - \kappa\big)$, the necessary gradients of the Lagrangian are given by

$$\begin{aligned}\nabla_\theta L(\theta, \lambda) &= \nabla J(\theta, x_0) + \lambda \nabla G(\theta, x_0) \\ &= \nabla J(\theta, x_0) + \lambda \left(\nabla U(\theta, x_0) - 2J(\theta, x_0)\nabla J(\theta, x_0)\right), \\ \nabla_\lambda L(\theta, \lambda) &= G(\theta, x_0) - \kappa.\end{aligned}$$

The expressions for $\nabla J(\theta, x_0)$ and $\nabla U(\theta, x_0)$ require a counterpart of $U(\cdot)$ with initial state-action pair (x, a) under policy θ, defined by

$$W(\theta, x, a) \stackrel{\triangle}{=} \mathbb{E}\left[\left(\sum_{n=0}^{\infty} \gamma^n k(x_n, a_n)\right)^2 \middle| x_0 = x,\ a_0 = a,\ \theta\right].$$

Similar to U, the function W also satisfies a fixed point equation:

$$\begin{aligned}W(\theta, x, a) = k(x, a)^2 &+ \gamma^2 \sum_y P(y|x, a) U(\theta, y) \\ &+ 2\gamma k(x, a) \sum_y P(y|x, a) J(\theta, y).\end{aligned}$$

We now provide expressions for the gradients $\nabla J(\theta, x_0)$ and $\nabla U(\theta, x_0)$.

Lemma 6.1.

$$\nabla J(\theta, x_0) = \sum_{x,a} \pi_\gamma^\theta(x, a|x_0) \nabla \log \mu^\theta(a|x) Q(\theta, x, a), \tag{6.1}$$

$$\begin{aligned}\nabla U(\theta, x_0) = &\sum_{x,a} \widetilde{\pi}_\gamma^\theta(x, a|x_0) \nabla \log \mu^\theta(a|x) W(\theta, x, a) \\ &+ 2\gamma \sum_{x,a,y} \widetilde{\pi}_\gamma^\theta(x, a|x_0) P(y|x, a) k(x, a) \nabla J(\theta, y),\end{aligned} \tag{6.2}$$

$$\text{where } Q(\theta, x, a) = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n k(x_n, a_n) \,\middle|\, x_0 = x,\ a_0 = a,\ \theta\right],$$

$$\pi_\gamma^\theta(x, a|x_0) = (1 - \gamma) \sum_{n=0}^{\infty} \gamma^n \mathbb{P}\left(x_n = x|x_0; \theta\right) \mu^\theta(a|x),$$

$$\widetilde{\pi}_\gamma^\theta(x, a|x_0) = (1 - \gamma^2) \sum_{n=0}^{\infty} \gamma^{2n} \mathbb{P}\left(x_n = x|x_0; \theta\right) \mu^\theta(a|x).$$

Note that $Q(\theta,\cdot,\cdot)$ is the Q-value function associated with policy μ^θ, while π_γ^θ and $\widetilde{\pi}_\gamma^\theta$ are the respective γ and γ^2-discounted visiting distributions of the state-action pair (x,a) under policy μ^θ.

Proof. We derive the expression for $\nabla U(\theta,x_0)$, as the proof for the case of $\nabla J(\theta,x_0)$ is standard. Using $U(\theta,x)=\sum_a \mu^\theta(x|a)W(\theta,x,a)$, and differentiating w.r.t. θ, we have

$$\begin{aligned}
\nabla U(\theta,x_0) &= \sum_a \nabla\mu^\theta(a|x_0)W(\theta,x_0,a) + \sum_a \mu^\theta(a|x_0)\nabla W(\theta,x_0,a)\\
&= \sum_a \nabla\mu^\theta(a|x_0)W(\theta,x_0,a) + \sum_a \mu^\theta(a|x_0)\\
&\quad\times \nabla\Big[k(x_0,a)^2+\gamma^2\sum_y P(y|x_0,a)U(\theta,y)\\
&\quad + 2\gamma k(x_0,a)\sum_y P(y|x_0,a)J(\theta,y)\Big]\\
&= \underbrace{\sum_a \nabla\mu^\theta(a|x_0)W(\theta,x_0,a) + 2\gamma\sum_{a,y}\mu^\theta(a|x_0)k(x_0,a)P(y|x_0,a)\nabla J(\theta,y)}_{h(\theta,x_0)}\\
&\quad + \gamma^2\sum_{a,y}\mu^\theta(a|x_0)P(y|x_0,a)\nabla U(\theta,y)\\
&= h(\theta,x_0)+\gamma^2\sum_x \mathbb{P}\left(x_1=x|x_0;\theta\right)\nabla U(\theta,x)\\
&= h(\theta,x_0)+\gamma^2\sum_x \mathbb{P}\left(x_1=x|x_0;\theta\right)\Big[h(\theta,x)+\gamma^2\sum_x\cdots\Big]
\end{aligned}$$

Repeated application of the above relationship yields

$$\begin{aligned}
\nabla U(\theta,x_0) &= \sum_{n=0}^\infty \gamma^{2n}\sum_x \mathbb{P}(x_n=x|x_0;\theta)h(\theta,x)\\
&= \sum_{n=0}^\infty \gamma^{2n}\Big[\sum_{x,a}\mathbb{P}\left(x_n=x|x_0;\theta\right)\mu^\theta(a|x)\nabla\log\mu^\theta(a|x)W(\theta,x,a)\\
&\quad + 2\gamma\sum_{x,a,y}\mathbb{P}(x_n=x|x_0;\theta)\mu^\theta(a|x)k(x,a)P(y|x,a)\nabla J(\theta,y)\Big]\\
&= \frac{1}{1-\gamma^2}\Big[\sum_{x,a}\widetilde{\pi}_\gamma(x,a|x_0)\nabla\log\mu^\theta(a|x)W(\theta,x,a)\\
&\quad +2\gamma\textstyle\sum_{x,a,y}\widetilde{\pi}_\gamma(x,a|x_0)k(x,a)P(y|x,a)\nabla J(\theta,y)\Big]. \qquad \square
\end{aligned}$$

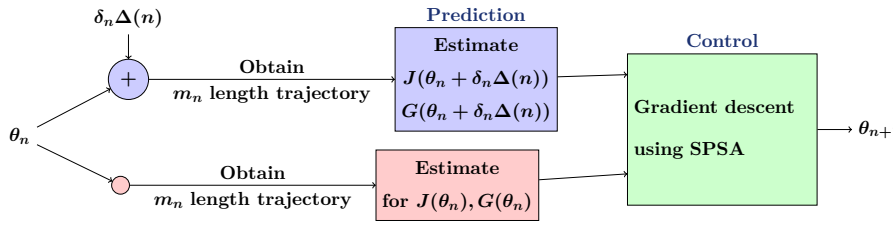


Figure 6.1: Overall flow of SPSA-based risk-sensitive policy gradient algorithm in a discounted cost MDP setting.

Estimating $\nabla J(\theta, x_0)$ and $\nabla U(\theta, x_0)$ is challenging due to the following reasons:

i) Two different sampling distributions are used for $\nabla J(\theta, x_0)$ and $\nabla U(\theta, x_0)$. In particular, the distributions π_γ^θ and $\widetilde{\pi}_\gamma^\theta$ involve factors γ and γ^2, respectively.

ii) $\nabla J(\theta, y)$ appears in the second summation on the RHS of (6.2), and this makes the estimation task hard in practice, as one needs an estimate of the gradient of the value function $J(\theta, y)$ at every state y of the MDP, and not just at the initial state x_0.

To overcome these issues, we use SPSA to estimate $\nabla J(\theta, x_0)$ and $\nabla U(\theta, x_0)$. As illustrated in Figure 6.1, such an estimation scheme requires running two trajectories corresponding to policy parameters $\theta_n + \delta_n\Delta(n)$ (where δ_n and $\Delta(n)$ are described in Section 4.4) and θ_n. The samples from the trajectories would be used to estimate $J(\theta_n + \delta_n\Delta(n))$, $J(\theta_n)$, $U(\theta_n+\delta_n\Delta(n))$ and $U(\theta_n)$, which in turn help in forming the estimates of the gradient of $J(\theta, x_0)$ and $U(\theta, x_0)$ as follows: For $i = 1, \ldots, \|\mathcal{X}\|$,

$$\widehat{\nabla}_i J(\theta_n, x_0) = \frac{\widehat{J}(\theta_n + \delta_n\Delta(n), x_0) - \widehat{J}(\theta_n, x_0)}{\delta_n\Delta_i(n)}, \text{ and}$$
$$\widehat{\nabla}_i U(\theta_n, x_0) = \frac{\widehat{U}(\theta_n + \delta_n\Delta(n), x_0) - \widehat{U}(\theta_n, x_0)}{\delta_n\Delta_i(n)}. \tag{6.3}$$

In the above, $\widehat{J}(\theta, x_0)$ (resp. $\widehat{U}(\theta, x_0)$) denotes an estimate of $J(\theta, x_0)$ (resp. $U(\theta, x_0)$), for any $\theta \in \Theta$.

Policy evaluation using TD

The task of estimating J is straightforward, and the regular TD algorithm described earlier in Section 4.3 can be employed. Notice that the policy parameter update would use an estimate of $\nabla_\theta L(\theta, \lambda)$ to perform an incremental update. In this case, this gradient features a factor of the form in the template algorithm (5.4) that is the product $J(\theta, x_0)\nabla J(\theta, x_0)$, and to ensure independence, we use double sampling to form estimates $\widehat{J}$ and $\hat{\widehat{J}}$ of $J(\theta, x_0)$. The first estimate would be employed in (6.3), while the second one would be used in the first term of the aforementioned product.

Recall that the value function J satisfies a fixed-point equation. One could possibly combine both J and G, or view the fixed-point equation for J in (2.5) together with the equation (3.2) over $2|\mathcal{X}|$ variables. However, relying on the equation (3.2) for policy optimization is challenging owing to the fact that variance lacks the monotonicity property. Monotonicity is required in classic policy improvement algorithms, and one cannot derive meaningful convergence guarantees in the absence of monotone operators. Alternatively, one can derive a fixed-point equation for the squared value function U, where the operator underlying this equation is a contraction mapping, and the variance can then be estimated using U and J, where estimation of the latter quantities is facilitated through TD-type learning algorithms. The following proposition presents the fixed-point equation for the squared value function.

Proposition 6.1. The squared value function $U(\theta, x)$ satisfies

$$U(\theta, x) = \sum_a \mu^\theta(a|x)k(x,a)^2 + \gamma^2 \sum_{a,y} \mu^\theta(a|x)P^\theta(y|x,a)U(\theta, y) + 2\gamma \sum_{a,y} \mu^\theta(a|x)P^\theta(y|x,a)k(x,a)J(\theta, y). \tag{6.4}$$

Proof.

$$
\begin{aligned}
U(\theta,x) &= \mathbb{E}\left[(D(\theta,x))^2 \mid x_0 = x\right] = \mathbb{E}\left[\left(k(x,a_0) + \sum_{m=1}^{\infty} \gamma^m k(x_m,a_m)\right)^2\right] \\
&= \mathbb{E}\left[k(x,a_0)^2\right] + \mathbb{E}\left[\left(\sum_{m=1}^{\infty} \gamma^m k(x_m,a_m)\right)^2 \mid x_0 = x\right] \\
&\quad + 2\gamma\mathbb{E}\left[k(x,a_0) \times \left(\sum_{m=0}^{\infty} \gamma^m k(x_m,a_m)\right) \mid x_0 = x\right] \\
&= \sum_a \mu^\theta(a|x)k(x,a)^2 + \gamma^2 \sum_{a,y} \mu^\theta(a|x)P^\theta(y|x,a)U(\theta,y) \\
&\quad + 2\gamma \sum_{a,y} \mu^\theta(a|x)P^\theta(y|x,a)k(x,a)J(\theta,y).
\end{aligned}
$$

□

Let $\boldsymbol{k}^\theta$ be a $|\mathcal{X}|$ vector of single-stage costs for each state, and $\boldsymbol{K}^\theta$ be a $|\mathcal{X}| \times |\mathcal{X}|$ matrix with the entries $\sum_a \mu^\theta(a|x)k(x,a)$ for each state x along the diagonal and zeroes elsewhere. For any $(\boldsymbol{J},\boldsymbol{U}) \in \mathbb{R}^{2|\mathcal{X}|}$, where $\boldsymbol{J}$ and $\boldsymbol{U}$ denote the first and last $|\mathcal{X}|$ entries, respectively, define

$$
\begin{aligned}
T^\theta(\boldsymbol{J},\boldsymbol{U}) &= \begin{bmatrix} T_1^\theta(\boldsymbol{J}) \\ T_2^\theta(\boldsymbol{J},\boldsymbol{U}) \end{bmatrix}, \text{ where} \\
T_1^\theta(\boldsymbol{J}) &= \boldsymbol{k}^\theta + \gamma P^\theta \boldsymbol{J}, \text{ and} \\
T_2^\theta(\boldsymbol{J},\boldsymbol{U}) &= \boldsymbol{K}^\theta \boldsymbol{k}^\theta + 2\gamma \boldsymbol{K}^\theta P^\theta \boldsymbol{J} + \gamma^2 P^\theta \boldsymbol{U}.
\end{aligned}
$$

Using the notation above, the fixed-point equations for J in (2.5) and U in (6.4) can be combined together as

$$
(\boldsymbol{J}^{\boldsymbol{\theta}},\boldsymbol{U}^{\boldsymbol{\theta}}) = T^\theta(\boldsymbol{J}^{\boldsymbol{\theta}},\boldsymbol{U}^{\boldsymbol{\theta}}),
$$

where $\boldsymbol{J}^{\boldsymbol{\theta}} = [J(\theta,x)]_{x\in\mathcal{X}}$ and $\boldsymbol{U}^{\boldsymbol{\theta}} = [U(\theta,x)]_{x\in\mathcal{X}}$.

Let d^θ denote the stationary distribution of the Markov chain underlying policy θ. We shall assume that such a distribution exists — an assumption that is easily satisfied for unichain policies (i.e., the underlying Markov chain is irreducible and positive recurrent). We now focus on establishing that the operator T^θ is a contraction mapping

w.r.t. a weighted norm, for any policy θ. The weighted norm, denoted by $\|\cdot,\cdot\|_\nu$, is defined as follows: for any $(J,U) \in \mathbb{R}^{2|\mathcal{X}|}$,

$$\|J,U\|_\nu = \nu\|J\|_{d^\theta} + (1-\nu)\|U\|_{d^\theta},$$

where $\|x\|_{d^\theta} = \sqrt{\sum_{i=1}^{|\mathcal{X}|} d^\theta(i)x_i^2}$ for any $x \in \mathbb{R}^{|\mathcal{X}|}$.

Proposition 6.2. There exists a $\nu \in (0,1)$ and $\bar{\gamma} < 1$ such that

$$\left\|T^\theta(J,U) - T^\theta(\bar{J},\bar{U})\right\|_\nu \leq \bar{\gamma}\left\|(J,U) - (\bar{J},\bar{U})\right\|_\nu, \forall J,\bar{J},U,\bar{U} \in \mathbb{R}^{|\mathcal{X}|}.$$

Proof. First, we show that T_1^θ is a contraction mapping. This can be inferred as follows: For any $y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}$,

$$\begin{aligned}||P^\theta J||_{d^\theta}^2 &= \sum_{i=1}^{|\mathcal{X}|} d^\theta(i)(\sum_{j=1}^{|\mathcal{X}|} P^\theta(j|i)J(j))^2 \leq \sum_{i=1}^{|\mathcal{X}|} d^\theta(i) \sum_{j=1}^{|\mathcal{X}|}(P^\theta(j|i)J(j)^2) \\ &= \sum_{j=1}^{|\mathcal{X}|}(\sum_{i=1}^{|\mathcal{X}|} d^\theta(i)P^\theta(j|i))(J(j))^2 = \sum_{j=1}^{|\mathcal{X}|} d^\theta(j)(J(j))^2 = ||J||_{d^\theta}^2.\end{aligned}$$

Using $||P^\theta J||_{d^\theta} \leq ||J||_{d^\theta}$, we have

$$\|T_1^\theta(J) - T_1^\theta(\bar{J})\|_{d^\theta} = \gamma\|P^\theta J - P^\theta \bar{J}\|_{d^\theta} \leq \gamma\|J - \bar{J}\|_{d^\theta}.$$

Now, for any $J,\bar{J},U,\bar{U} \in \mathbb{R}^{|\mathcal{X}|}$, we have

$$\begin{aligned}&\|T_2^\theta(J,U) - T_2^\theta(\bar{J},\bar{U})\|_{d^\theta} \\ =&\|2\gamma \boldsymbol{K}^\theta P^\theta J - 2\gamma \boldsymbol{K}^\theta P^\theta \bar{J} + \gamma^2 P^\theta U - \gamma^2 P^\theta \bar{U}\|_{d^\theta} \\ \leq&2\gamma\|\boldsymbol{K}^\theta P^\theta J - \boldsymbol{K}^\theta P^\theta \bar{J}\|_{d^\theta} + \gamma^2\|U - \bar{U}\|_{d^\theta} \\ \leq&\gamma C_1\|J - \bar{J}\|_{d^\theta} + \gamma^2\|U - \bar{U}\|_{d^\theta},\end{aligned}$$

for some $C_1 < \infty$. The first inequality above follows by using $||P^\theta J||_{d^\theta} \leq ||J||_{d^\theta}$, while the second inequality follows by using the equivalence of norms.

Now, setting $\nu = \dfrac{\gamma C_1}{\epsilon + \gamma C_1}$, with ϵ satisfying $\gamma + \epsilon < 1$, we have

$$\begin{aligned}&\|T^\theta(J,U) - T^\theta(\bar{J},\bar{U})\|_\nu \\ =&\nu\|T_1^\theta J - T_1^\theta \bar{J}\|_{d^\theta} + (1-\nu)\|T_2^\theta U - T_2^\theta \bar{U}\|_{d^\theta}\end{aligned}$$

$$\begin{aligned}
&\leq \nu\gamma\|J-\bar{J}\|_{d^\theta} + (1-\nu)\gamma C_1\|J-\bar{J}\|_{d^\theta} + (1-\nu)\gamma^2\|U-\bar{U}\|_{d^\theta} \\
&\leq \nu(\gamma+\epsilon)\|J-\bar{J}\|_{d^\theta} + (1-\nu)\gamma\|U-\bar{U}\|_{d^\theta} \\
&\leq (\gamma+\epsilon)\|(J,U)-(\bar{J},\bar{U})\|_\nu.
\end{aligned}$$

The claim follows by setting $\bar{\gamma} = \gamma + \epsilon$. □

We now have what we need to estimate the squared value function U using a TD-type algorithm, as U satisfies the fixed-point equation (6.4), and the T^θ operator underlying this equation is well behaved, in the sense that Proposition 6.2 establishes that it leads to a contraction mapping that is amenable for stochastic approximation, so the results of Sections 4.2 and 4.3 are applicable. On the other hand, note that estimating variance directly would not help, because the corresponding underlying operator is not monotone.

From the foregoing, we have the following TD-type update for estimating U:

$$\begin{aligned}
U_{n+1}(x) = U_n(x) + \zeta(\nu(x,n))\mathbb{I}\{x_n = x\}\Big(&k(x_n,a_n)^2 \\
&+2\gamma k(x_n,a_n)J_n(x_{n+1}) + \gamma^2 U_n(x_{n+1}) - U_n(x_n)\Big),
\end{aligned} \tag{6.7}$$

where $\nu(x,n) = \sum_{m=0}^{n} \mathbb{I}\{x_m = x\}$ and x_{n+1} is a r.v. sampled from $P(\cdot \mid x_n, a_n)$. Notice that the factor J goes into the fixed-point equation for U, and hence, the TD algorithm for U has to employ the TD-based estimate of J for estimating U.

Algorithm 1 presents the pseudocode for the risk-sensitive policy-gradient algorithm for the discounted-cost setting. In a nutshell, this algorithm uses multi-timescale stochastic approximation to perform the following tasks: (i) run the TD algorithm on the fastest timescale to estimate both J and U; (ii) use an SPSA-based gradient descent scheme on the intermediate timescale for solving the primal problem in (1.1); and (iii) perform dual ascent on the Lagrange multiplier using the sample variance constraint (using the estimate of U) on the slowest timescale. The latter two-timescale updates follow the template provided in Section 5.

Algorithm 1: Policy gradient algorithm under variance as a risk measure in a discounted-cost MDP setting

Input : initial parameter $\theta_0 \in \Theta$, perturbation constants $\{\delta_n\}$, trajectory lengths $\{m_n\}$, step sizes $\{\zeta_1(n)\}$, $\{\zeta_2(n)\}$, projection operators Γ and Γ_λ, # iterations M.

for $n \leftarrow 0$ **to** $M-1$ **do**

  Set $\Delta(n)$ using symmetric ± 1-valued Bernoulli distribution;

  for $m \leftarrow 0$ **to** $m_n - 1$ **do**

    `/* Unperturbed policy simulation */`

    Use the policy μ^{θ_n} to draw action $a_m \sim \mu^{\theta_n}(\cdot \mid x_m)$;

    Observe next state x_{m+1} and cost $k(x_m, a_m)$;

    Use (4.12) and (6.7) to form estimates $\widehat{J}(\theta_n, x_0)$ and $\widehat{G}(\theta_n, x_0)$ of $J(\theta_n)$ and $G(\theta_n)$, respectively;

    Use another independent sample trajectory to form the estimate $\hat{\hat{J}}(\theta_n, x_0)$ of $J(\theta_n)$;

    `/* Perturbed policy simulation */`

    Use the policy $\mu^{\theta_n + \delta_n \Delta(n)}$ to generate the state x_m^+, draw action $a_m^+ \sim \mu^{\theta_n + \delta_n \Delta(n)}\left(\cdot \mid x_m^+\right)$;

    Observe next state x_{m+1}^+ and cost $k(x_m^+, a_m^+)$;

    Use (4.12) and (6.7) to form estimates $\widehat{J}(\theta_n + \delta_n \Delta(n), x_0)$ and $\widehat{G}(\theta_n + \delta_n \Delta(n), x_0)$ of $J(\theta_n + \delta_n \Delta(n))$ and $G(\theta_n + \delta_n \Delta(n))$, respectively;

  end

  Gradient estimate for the objective:

$$\widehat{\nabla}_i J(\theta_n, x_0) = \frac{\widehat{J}(\theta_n + \delta_n \Delta(n), x_0) - \widehat{J}(\theta_n, x_0)}{\delta_n \Delta_i(n)};$$

  Gradient estimate for the constraint:

$$\widehat{\nabla}_i U(\theta_n, x_0) = \frac{\widehat{U}(\theta_n + \delta_n \Delta(n), x_0) - \widehat{U}(\theta_n, x_0)}{\delta_n \Delta_i(n)};$$

  `/* Policy update: Gradient descent using SPSA */`

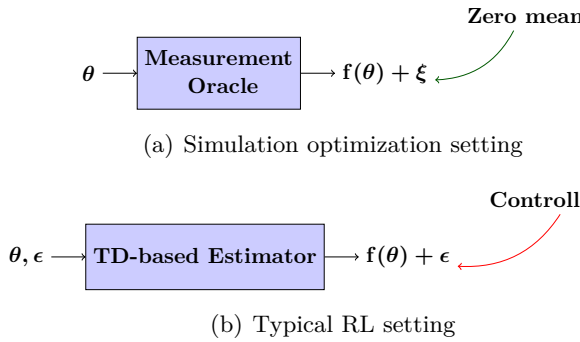
$$\theta_{n+1} = \Gamma\Bigg[\theta_n - \zeta_2(n)\Bigg(\widehat{\nabla} J(\theta_n, x_0) + \lambda_n\left(\widehat{\nabla} U(\theta_n, x_0) - 2\hat{\hat{J}}(\theta_n, x_0)\widehat{\nabla} J(\theta_n, x_0)\right)\Bigg)\Bigg];$$

  `/* Lagrange multiplier update: Dual ascent */`

$$\lambda_{n+1} = \Gamma_\lambda\left[\lambda_n + \zeta_1(n)\left(\widehat{U}(\theta_n, x_0) - 2\hat{\hat{J}}(\theta_n, x_0) - \kappa\right)\right];$$

end

Output : Policy θ_M



(a) Simulation optimization setting

(b) Typical RL setting

Figure 6.2: Illustration of the difference between classic simulation optimization and optimization of the variance risk measure in an RL setting. In the latter setting, the error ϵ in function estimates can be *controlled* and made very low at the cost of additional simulations.

On the batch size m_n

To understand the challenge in choosing an appropriate batch size m_n for policy evaluation in Step 2 of Algorithm 1, so that the overall algorithm converges, consider a simpler setting of optimizing a smooth function f, i.e.,

$$\text{find} \quad \theta^* = \arg\min_{\theta \in \Theta} f(\theta), \tag{6.8}$$

where Θ is a convex and compact subset of $\mathbb{R}^d$. In a classic stochastic optimization setting, one has an oracle that supplies noisy function measurements, but the noise is usually zero mean. On the other hand, in a typical RL setting, the function f that has to be estimated from sample trajectories is the value $J(\theta)$ associated with a given policy θ, as illustrated in Figure 6.2. For the sake of simplicity, we drop the dependence on the parameter θ, and instead, study the value estimation problem. Subsequently, when we analyze the policy gradient scheme in Algorithm 1 for CPT-value optimization, we shall make the dependence on the policy parameter explicit.

For a given policy with a fixed start state x_0, let m denote the length of the sample trajectory used to form an estimate J_m, using (4.12), of the value $J(x_0)$. One can derive a bound on the estimation

error $|J_m - J(x_0)|$, and such a bound would aid the proof of asymptotic convergence of the risk-sensitive policy gradient in Algorithm 1. An informal statement of such a finite-time bound for TD is as follows: *Using a step size $\zeta(n) = c/n$ with a suitable choice for the constant c,*

$$\mathbb{E}\left\|\widehat{J}_m - J(x_0)\right\| = O\left(\frac{1}{\sqrt{m}}\right). \tag{6.9}$$

We avoid a detailed discussion of the derivation of such a bound, as it is quite technical, and deviates from the focus of optimizing risk in an RL setting. However, in passing, we note that the step size c in the bound above would require information about the underlying transition dynamics, and such a problematic dependence can be avoided by using Polyak-Ruppert iterate averaging, where one employs a bigger step size c/n^α, with $\alpha \in (1/2, 1)$ together with averaging of the iterates. Such an approach would result in a bound of the order $O\left(\frac{1}{m^{\alpha/2}}\right)$.

In the following discussion, we shall use f to denote the smooth objective function that we want to minimize. From the foregoing, it is apparent that we have a setting where f is not perfectly observable, and instead, one can obtain biased measurements of f at any input parameter θ. Choosing larger values of the batch size m leads to an increase in the accuracy of the function measurement. In particular, from (6.9), the estimation bias is of order $O(\frac{1}{\sqrt{m}})$. Figure 6.2 illustrates this difference in estimation between a classic optimization setting, and a typical RL setting, where the policy evaluation is performed for estimation of the value of a given policy.

A stochastic gradient-descent scheme to solve the problem defined in (6.8) would update as follows:

$$\theta_{n+1} = \Gamma\left(\theta_n - \gamma_n \widehat{\nabla} f(\theta_n)\right), \tag{6.10}$$

where $\{\gamma_n\}$ is a step-size sequence that satisfies standard stochastic approximation conditions, $\Gamma = (\Gamma_1, \ldots, \Gamma_d)$ is an operator that ensures that the update (6.10) stays bounded within the compact and convex set Θ, and $\widehat{\nabla} f(\theta_n)$ is an estimate of the gradient of f at θ_n.

Suppose that the gradient estimate $\widehat{\nabla} f(\theta_n)$ in (6.10) is formed using SPSA, as described in Section 4.4, i.e.,

$$\widehat{\nabla}_i f(\theta_n) = \frac{\widehat{f}(\theta_n + \delta_n \Delta(n)) - \widehat{f}(\theta_n)}{\delta_n \Delta_i(n)},$$

where $\widehat{f}(\theta)$ denotes the estimate of $f(\theta)$, when the underlying parameter is θ. Suppose that the estimation scheme returns $\widehat{f}(\theta_n) = f(\theta_n) + \varphi_n^{\theta}$, where φ_n^{θ} denotes the error in estimating the objective f using m_n function measurements. For the sake of this discussion, suppose that the estimation error vanishes at the rate $\frac{1}{m_n^{1/2}}$. We first rewrite the update rule in (6.10) as follows:

$$\theta_{n+1}^i = \Gamma_i\Big(\theta_n^i - \gamma_n\Big(\frac{f(\theta_n + \delta_n\Delta(n)) - f(\theta_n)}{\delta_n\Delta_i(n)} + \kappa_n\Big)\Big),$$

where $\kappa_n = \frac{(\varphi_n^{\theta_n+\delta_n\Delta(n)} - \varphi_n^{\theta_n})}{\delta_n\Delta_i(n)}$. Let $\zeta_n = \sum_{l=0}^{n}\gamma_l\kappa_l$. Then, a critical requirement that allows us to ignore the estimation error term ζ_n is the following condition:

$$\sup_{l\geq 0}(\zeta_{n+l} - \zeta_n) \to 0 \text{ as } n \to \infty. \tag{6.11}$$

Notice that the estimation error in ζ_n is a function of number of samples m_n used for estimating the objective value, and it is obviously necessary to increase the number of samples m_n so that the bias vanishes asymptotically. In addition to the usual conditions on the step-size sequence and perturbation constant δ_n, one possible choice for m_n that ensures that the bias in the gradient estimate vanishes and the overall algorithm converges is the following: $\frac{1}{\sqrt{m_n}\delta_n} \to 0$.

Remark 6.1. A similar observation holds even for the case where the function f is the squared value function J. In this case, the estimation scheme is TD-learning, and order $\frac{1}{\sqrt{m}}$ bound on the root mean-squared error of TD-learning can be derived. In this case, the condition on m_n for ensuring convergence of the overall stochastic gradient scheme would be $\frac{1}{m_n^{1/2}\delta_n} \to 0$. Finally, similar considerations on the trajectory lengths hold for the purpose of CVaR estimation, as well as estimation of CPT-value. In both cases, the estimation procedure (see Algorithms 1 and 3) is asymptotically unbiased, but one does not have the luxury of having a very long run of the policy evaluation procedure, considering that the outer loop of incremental policy update needs to perform policy evaluation often.

Remark 6.2. (Extension to incorporate function approximation) One could parameterize both J and U using linear function approximation

and then employ TD-type schemes for policy evaluation. Notice that both $J(\cdot)$ and $U(\cdot)$ need to be evaluated for the perturbed policies. Let $J(x) \approx v^\mathsf{T}\phi_v(x)$ and $U(x) \approx u^\mathsf{T}\phi_u(x)$ be the linear approximations to J and U, respectively, with features $\phi_v(\cdot)$ and $\phi_u(\cdot)$ from low-dimensional spaces. It can be shown that an appropriate operator can be defined for U using the above equation and an operator that projects orthogonally onto the linear space $\{\Phi_u u \mid u \in \mathbb{R}^d\}$. Such a projected Bellman operator turns out to be a contraction mapping, and hence, a TD-type scheme can be arrived at, along the lines of that for the regular cost J.

6.2 Case 2: Average-cost MDP + variance as risk

We consider the following constrained optimization problem for average cost MDPs:

$$\min_\theta J(\theta) \qquad \text{subject to} \qquad G(\theta) \leq \kappa,$$

where $J(\theta)$ is the long-run average cost and $G(\theta)$ is the variance, as defined in Sections 2.3 and 3.3.

Gradient of the Lagrangian

Letting $L(\theta, \lambda) \triangleq J(\theta) + \lambda\big(G(\theta) - \kappa\big)$, and noting that $\nabla G(\theta) = \nabla\eta(\theta) - 2J(\theta)\nabla J(\theta)$, it is apparent that $\nabla J(\theta)$ and $\nabla\eta(\theta)$ are enough to calculate the necessary gradients of the Lagrangian. Let U^θ and W^θ denote the differential value and action-value functions associated with the squared cost under policy μ^θ, respectively. These two quantities satisfy the following Poisson equations:

$$\eta(\theta) + U(\theta, x) = \sum_a \mu^\theta(a|x)\big[k(x,a)^2 + \sum_y P(y|x,a)U(\theta,y)\big],$$
$$\eta(\theta) + W(\theta, x, a) = k(x,a)^2 + \sum_y P(y|x,a)U(\theta,y).$$

As mentioned earlier, we consider finite state-action space MDPs, which together with an irreducibility assumption implies the existence of a stationary distribution for the Markov chain underlying any policy θ. Denote by $d^\theta(x)$ and $\pi^\theta(x,a) = d^\theta(x)\mu^\theta(a|x)$, the stationary distribution

of state x and state-action pair (x, a) under policy μ^θ, respectively. We now present the gradients of $J(\theta)$ and $\eta(\theta)$ below.

$$\nabla J(\theta) = \sum_{x,a} \pi^\theta(x,a) \nabla \log \mu^\theta(a|x) Q(\theta, x, a), \tag{6.12}$$

$$\nabla \eta(\theta) = \sum_{x,a} \pi^\theta(x,a) \nabla \log \mu^\theta(a|x) W(\theta, x, a), \tag{6.13}$$

where $Q(\theta, x, a) = \sum_{n=0}^{\infty} \mathbb{E}\big[k(x_n, a_n) - J(\theta) \mid x_0 = x, a_0 = a, \mu\big]$, with actions $a_n \sim \mu^\theta(\cdot \mid x_n)$.

The above relationships follow from parameterizing the policies, and hence, the gradient of the transition probabilities can be estimated from the policy alone. This is the well-known policy gradient technique that makes it amenable for estimating the gradient of a performance measure in MDPs, since the values of the transition probabilities are not required and one can work with policies and simulated transitions from the MDP.

An important observation concerning $\nabla J(\theta)$ is that any function $b : \mathcal{X} \to \mathbb{R}$ can be added or subtracted to $Q(\theta, x, a)$ on the RHS of (6.12), and the resulting summation stays as $\nabla J(\theta)$. In a risk-neutral setting, a popular choice is to replace $Q(\theta, x, a)$ with the advantage function $A(\theta, x, a) = Q(\theta, x, a) - V(\theta, x)$.

Policy evaluation using TD

In a typical RL setting, $\nabla J(\theta)$ has to be estimated, and from the discussion before, this implies estimation of the advantage function using samples – TD-learning is a straightforward choice for this task. Using the expression on the RHS of (6.12), one can arrive at a decrement factor for the policy update as follows: substitute a TD-based empirical approximation to the advantage function, calculate the likelihood ratio $\nabla \log \mu(\cdot)$, and perform a gradient descent using the product of the advantage estimate with the likelihood ratio, and arrive at an empirical approximation to the RHS of (6.12) with the advantage function A instead of Q there. The TD algorithm-based estimate for the value function is given below.

$$\delta_n = k(x_n, a_n) - J_{n+1} + V_n(x_{n+1}) - V_n(x_n),$$
$$V_{n+1} = V_n + \zeta_3(n)\delta_n,$$

The idea described above, i.e., to use the advantage function in place of Q, can be used for the case of $\nabla\eta(\theta)$ as well, with the advantage function variant $B(\theta, x, a) = W(\theta, x, a) - U(x; \theta)$ on the RHS of (6.13). The TD algorithm-based estimate for the squared value function is given below.

$$\epsilon_n = k(x_n, a_n)^2 - \eta_{n+1} + U_n(x_{n+1}) - U_n(x_n),$$
$$U_{n+1} = U_n + \zeta_3(n)\epsilon_n.$$

The pseudocode of the overall algorithm in the average reward setting is given in Algorithm 2. As in the discounted case discussed in the previous section, an extra sample trajectory is needed to form an independent estimate $\widehat{J}_n$, so that the product $\widehat{J}_n\delta_n\psi_n$ that we have in the policy parameter update can be separated after taking expectations to obtain $J(\theta_n)\nabla J(\theta_n)$ in the convergence analysis.

In addition to the step-size requirements in (A4), we require that $\zeta_2(n) = o(\zeta_3(n))$ and $\zeta_4(n)$ is a constant multiple of $\zeta_3(n)$. Such choices ensure that the TD-critic and average cost updates are on the fastest timescale, the policy update is on an intermediate timescale, and the Lagrange multiplier update is on the slowest timescale.

Remark 6.3. The variance notion employed in this section involved measuring the deviations of the single-stage cost from its average. As we demonstrated in Algorithm 2, the per-period variance as a risk measure lends itself to policy gradient techniques well, since the likelihood ratio method can be employed to solve the risk-constrained problem. In contrast, the variance notion in the discounted cost setting involved the variance of the cumulative discounted cost, i.e., the (overall) variance of the underlying r.v. and not the per-period one. Such a measure is hard to optimize (see discussion below (6.2)), though SPSA could be employed. The flip side to the latter approach is that we do not exploit the structure of the underlying problem in forming the gradient estimates, e.g., using the likelihood ratio method. More importantly, SPSA requires simulation of two independent trajectories (corresponding

Algorithm 2: Policy gradient algorithm under variance as a risk measure in an average cost MDP setting

Input : initial parameter $\theta_0 \in \Theta$, where Θ is a compact and convex subset of $\mathbb{R}^d$, step sizes $\{\zeta_1(n)\}$, $\{\zeta_2(n)\}$, $\{\zeta_3(n)\}$, $\{\zeta_4(n)\}$, projection operators Γ and Γ_λ, # iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**

Draw action $a_m \sim \mu^{\theta_n}(\cdot|x_m)$, observe next state x_{m+1} and cost $k(x_m, a_m)$;

`/* Estimate for average cost */`

$J_{n+1} = (1-\zeta_4(n))J_n + \zeta_4(n)k(x_n, a_n)$;

`/* Another estimate for average cost */`

Draw action $\hat{a}_m \sim \mu^{\theta_n}(\cdot|\hat{x}_m)$, observe next state $\hat{x}_{m+1}$ and cost $k(\hat{x}_m, \hat{a}_m)$;

$\widehat{J}_{n+1} = (1-\zeta_4(n))\widehat{J}_n + \zeta_4(n)k(\hat{x}_n, \hat{a}_n)$;

`/* Estimate for average squared cost */`

$\eta_{n+1} = (1-\zeta_4(n))\eta_n + \zeta_4(n)k(x_n, a_n)^2$;

`/* TD estimate for the value function */`

$\delta_n = k(x_n, a_n) - J_{n+1} + V_n(x_{n+1}) - V_n(x_n)$;

$V_{n+1} = V_n + \zeta_3(n)\delta_n$;

`/* TD estimate for the squared value function */`

$\epsilon_n = k(x_n, a_n)^2 - \eta_{n+1} + U_n(x_{n+1}) - U_n(x_n)$;

$U_{n+1} = U_n + \zeta_3(n)\epsilon_n$;

Set $\psi_n = \nabla \log \mu^{\theta_n}(a_n|x_n)$; `// Likelihood ratio`

`/* Policy update */`

$\theta_{n+1} = \Gamma\Big(\theta_n - \zeta_2(n)\big(-\delta_n\psi_n + \lambda_n(\epsilon_n\psi_n - 2\widehat{J}_{n+1}\delta_n\psi_n)\big)\Big)$;

`/* Lagrange multiplier update */`

$\lambda_{n+1} = \Gamma_\lambda\Big(\lambda_n + \zeta_1(n)(\eta_{n+1} - J_{n+1}^2 - \kappa)\Big)$;

end

Output : Policy θ_M

to unperturbed and perturbed policy parameters), and this may not be feasible in many practical applications.

One could consider swapping the risk measures of discounted and average cost settings, i.e., employ per-period variance in a discounted cost MDP, and overall variance in the average cost MDP. Leaving the question of which is the best risk measure for a given MDP aside, we believe that such a swap of risk measures would make solving the average cost problem difficult, and discounted cost problem easy in comparison.

Remark 6.4. As in the discounted setting, incorporating function approximation for the functions J and U is straightforward, and we omit the details.

6.3 Case 3: Stochastic shortest path + CVaR as risk

We again consider the following constrained optimization problem: For a given $\kappa > 0$ and initial state x_0 of the SSP MDP,

$$\min_{\theta} J(\theta, x_0) \qquad \text{subject to} \qquad G(\theta, x_0) \leq \kappa,$$

where $J(\theta, x_0)$ and $G(\theta, x_0)$ are the expectation and $\text{CVaR}_\beta, \beta \in (0, 1)$, of the total cost r.v. $D(\theta, x_0)$, respectively (see Sections 2.2 and 3.4).

Gradient of the Lagrangian

With the Lagrangian $L(\theta, \lambda) \triangleq J(\theta, x_0) + \lambda\big(G(\theta, x_0) - \kappa\big)$, the necessary gradients for solving the constrained problem above are $\nabla J(\theta, x_0)$ and $\nabla \text{CVaR}_\beta(D(\theta, x_0))$. Using the likelihood ratio method, the first gradient is obtained as follows:

$$\nabla J(\theta, x_0) = \mathbb{E}\left[\left[\sum_{n=0}^{\tau-1} k(x_n, a_n)\right] \sum_{m=0}^{\tau-1} \nabla \log \mu^\theta(a_m \,|x_m)\,\middle|\, x_0\right].$$

To estimate the gradient of the CVaR of $D(\theta, x_0)$ for a given policy parameter θ, we use the following variation of the policy gradient theorem for CVaR:

$$\nabla \mathrm{CVaR}_\beta(D(\theta, x_0)) = \mathbb{E}\left[[D(\theta, x_0) - \mathrm{VaR}_\beta(D(\theta, x_0))] \times \sum_{m=0}^{\tau-1} \nabla \log \mu^\theta(a_m \,|x_m) \middle| D(\theta, x_0) \geq \mathrm{VaR}_\beta(D(\theta, x_0))\right].$$

We shall provide a derivation of the expression above in the next section. In particular, we shall specialize the gradient expression for an abstract coherent risk measure to handle the case of CVaR, and the reader is referred to Section 7.3 for the details.

VaR and CVaR estimation

What remains to be specified is the technique employed for estimating VaR and CVaR for a given policy θ. Notice that CVaR estimation is required for dual ascent, since $\nabla_\lambda L(\theta, \lambda) = \mathrm{CVaR}_\beta(D(\theta, x_0)) - \kappa$. VaR is required for estimating CVaR and the CVaR gradient. A well-known result is that both VaR and CVaR can be obtained from the solution of a certain convex optimization problem. More precisely, for any r.v. X, let

$$v(\xi, X) := \xi + \frac{1}{1-\beta}(X - \xi)_+ \text{ and } V(\xi) = \mathbb{E}\left[v(\xi, X)\right].$$

Then, $\mathrm{VaR}_\beta(X)$ is the minimizer of V, i.e., a point ξ^*_β that satisfies $V'(\xi^*_\beta) = 0$ and $\mathrm{CVaR}_\beta(X) = V(\xi^*_\beta)$.

Since $v(\xi, \cdot)$ is continuous w.r.t. ξ, $V'(\xi) = \mathbb{E}\left(1 - \frac{1}{1-\beta}\mathbb{I}\{X \geq \xi\}\right)$. The minimizer ξ^* would be a VaR, and $V(\xi^*)$ would be the CVaR of the r.v. X. Observing that V is convex, a stochastic approximation-based procedure can be derived for estimating VaR and CVaR. In an SSP context, the r.v. is $D(\theta, x_0)$. Suppose that we can obtain i.i.d. samples from the distribution of $D(\theta, x_0)$, i.e., we can simulate the underlying SSP using the policy θ. Let $D_k, k = 1, \ldots$ denote these samples. Then, VaR and CVaR can be estimated as follows:

$$\text{VaR: } \xi_m = \xi_{m-1} - \zeta_3(m)\left(1 - \frac{1}{1-\beta}\mathbb{I}\{D_m \geq \xi_m\}\right), \quad (6.14)$$

$$\text{CVaR: } C_m = C_{m-1} - \frac{1}{m}\left(C_{m-1} - v(\xi_{m-1}, C_{m-1})\right). \quad (6.15)$$

In the above, (6.14) can be seen as a gradient descent rule, while (6.15) can be seen as a plain averaging update. The step-size sequence $\{\zeta_3(m)\}$

is required to satisfy standard stochastic approximation conditions, i.e., $\sum_m \zeta_3(m) = \infty$, and $\sum_m \zeta_3(m)^2 < \infty$.

The complete algorithm, along with the update rules for various parameters, is presented in Algorithm 3.

6.4 Case 4: Stochastic shortest path + chance constraint as risk

The last case studied in this section employs a chance constraint in the optimization problem (1.1), i.e.,

$$\min J(\theta, x_0) \qquad \text{subject to} \qquad G(\theta, x_0) \leq \kappa,$$

where $J(\theta, x_0)$ is the expectation the total cost r.v. $D(\theta, x_0)$, while $G(\theta, x_0) = \mathbb{P}(D(\theta, x_0) \geq \beta)$ is the probability that feeds into the chance constraint (see Sections 2.2 and 3.5).

From the discussion in the previous sections, it is apparent that the main technical challenges in handling any risk measure are as follows: (i) estimation of the risk measure from samples; and (ii) gradient estimation for the policy update iteration. For the sake of brevity, we provide the necessary details for handling (i) and (ii), and the rest of the pieces of the resulting actor-critic scheme follows in a manner similar to that for variance or CVaR.

To handle (i), suppose that we are given n i.i.d. samples, say $\{X_1, \ldots, X_n\}$, from the distribution of X, and the goal is to estimate the probability involved in the chance constraint, i.e., $\mathbb{P}(X \geq \beta)$. The sample average estimator for the latter probability is given by

$$\overline{\mathcal{C}}_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{X_i \geq \beta\}.$$

For handling point (ii) concerning the policy gradient for the chance probability, we use the following likelihood ratio gradient expression for the chance probability:

$$\nabla G(D(\theta, x_0)) = \mathbb{E}\left[\sum_{m=0}^{\tau-1} \nabla \log \mu^{\theta}(a_m \,|x_m) \mathbb{I}\{D(\theta, x_0) \geq \beta\}\right].$$

The complete algorithm with chance constraint as the risk measure and the usual value function as the objective is presented in Algorithm 4.

Algorithm 3: Policy gradient algorithm under CVaR as a risk measure in an SSP setting

Input : initial parameter $\theta_0 \in \Theta$, where Θ is a compact and convex subset of $\mathbb{R}^d$, $\beta \in (0,1)$, trajectory lengths $\{m_n\}$, step sizes $\{\zeta_1(n)\}$, $\{\zeta_2(n)\}$, $\{\zeta_3(n)\}$, projection operators Γ and Γ_λ, # iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**

  for $m \leftarrow 0$ **to** $m_n - 1$ **do**

    Simulate the SSP for an episode to generate the state sequence $\{x_{n,j}\}$ using actions $\{a_{n,j} \sim \mu^{\theta_n}(\cdot \mid x_{n,j})\}$. Let τ_m denote the time instant when state 0 was visited in this episode;

    Observe total cost $D_{n,m} = \sum_{j=0}^{\tau_m - 1} k(x_{n,j}, a_{n,j})$;

    Calculate likelihood ratio: $\psi_{n,m} = \sum_{j=0}^{\tau_m - 1} \nabla \log \mu^{\theta_n}(a_{n,j} | x_{n,j})$;

  end

  `/* Policy evaluation */`

  Use the scheme in (6.14)–(6.15) to obtain the VaR estimate ξ_n and CVaR_β-estimate C_n;

  Total cost estimate: $\bar{D}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} D_{n,j}$;

  Likelihood ratio: $\bar{\psi}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} \psi_{n,j}$;

  `/* Gradient of the objective */`

  $\widehat{\nabla} J(\theta_n) = \bar{D}_n \bar{\psi}_n$;

  `/* Gradient of the risk measure */`

  $\widehat{\nabla} \text{CVaR}_\beta(\theta_n) = (C_n - \xi_n)\bar{\psi}_n \mathbb{I}\{C_n \geq \xi_n\}$;

  `/* Policy and Lagrange Multiplier Update */`

  $\theta_{n+1} = \Gamma\left(\theta_n - \zeta_2(n)\left(\widehat{\nabla} J(\theta_n) + \lambda_n \widehat{\nabla} \text{CVaR}_\beta(\theta_n)\right)\right)$;

  $\lambda_{n+1} = \Gamma_\lambda\Big(\lambda_n + \zeta_1(n)(C_n - \kappa)\Big)$;

end

Output : Policy θ_M

Algorithm 4: Policy gradient algorithm under the chance constraint in an SSP setting

Input : initial parameter $\theta_0 \in \Theta$, where Θ is a compact and convex subset of $\mathbb{R}^d$, $\beta \in (0,1)$, trajectory lengths $\{m_n\}$, step sizes $\{\zeta_1(n)\}$, $\{\zeta_2(n)\}$, $\{\zeta_3(n)\}$, projection operators Γ and Γ_λ, # iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**

  for $m \leftarrow 0$ **to** $m_n - 1$ **do**

    Simulate the SSP for an episode to generate the state sequence $\{x_{n,j}\}$ using actions $\{a_{n,j} \sim \mu^{\theta_n}(\cdot \mid x_{n,j})\}$. Let τ_m denote the time instant when state 0 was visited in this episode;

    Observe total cost $D_{n,m} = \sum_{j=0}^{\tau_m - 1} k(x_{n,j}, a_{n,j})$;

    Observe sample chance constraint $C_{n,m} = \mathbb{I}\{D_{n,m} \geq \beta\}$;

    Calculate likelihood ratio:

    $\psi_{n,m} = \sum_{j=0}^{\tau_m - 1} \nabla \log \mu^{\theta_n}(a_{n,j} | x_{n,j})$;

  end

  `/* Policy evaluation */`

  Total cost estimate: $\bar{D}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} D_{n,j}$;

  Likelihood ratio: $\bar{\psi}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} \psi_{n,j}$;

  Chance constraint estimate: $\bar{C}_n = \frac{1}{m_n} \sum_{j=1}^{m_n} C_{n,j}$;

  `/* Gradient of the objective */`

  $\widehat{\nabla} J(\theta_n) = \bar{D}_n \bar{\psi}_n$;

  `/* Gradient of the risk measure */`

  $\widehat{\nabla} G(\theta_n) = \bar{C}_n \bar{\psi}_n$;

  `/* Policy and Lagrange Multiplier Update */`

  $\theta_{n+1} = \Gamma\left(\theta_n - \zeta_2(n)\left(\widehat{\nabla} J(\theta_n) + \lambda_n \widehat{\nabla} G(\theta_n)\right)\right)$;

  $\lambda_{n+1} = \Gamma_\lambda\Big(\lambda_n + \zeta_1(n)(C_n - \kappa)\Big)$;

end

Output : Policy θ_M

6.5 Bibliographic remarks

The presentation of the risk-sensitive RL algorithm with variance as the underlying risk measure in discounted and average cost MDPs is based on Prashanth and Ghavamzadeh (2016), while the descriptions for the cases of CVaR and chance constraints are based on Prashanth (2014) and Chow *et al.* (2017), respectively. In the following, we provide additional bibliographic remarks for each case studied.

6.1 For a justification of the requirement in (6.11), see Borkar (2008, Lemma 1 in Chapter 2). In Prashanth and Ghavamzadeh (2016), the authors parameterize both J and U using linear function approximation, and show that the underlying projected Bellman operators are contractive; see Prashanth and Ghavamzadeh (2016, Lemma 2) for a proof. The fact that linear parameterization for J leads to a contraction mapping is well known, and a similar approach was shown to work for the squared cost U in Tamar *et al.* (2013) for an SSP setting. In Prashanth and Ghavamzadeh (2016), the authors extended this idea to include discounted problems. Notice that linear parameterization for J and U implies the underlying variance is also parameterized; however, a direct parameterization of variance is not feasible, as the underlying operator is not monotone, see Sobel (1982). For the finer details of the linear function approximation case in the discounted setting, see Prashanth and Ghavamzadeh (2016).

6.2 The expression in (6.12) for the gradient of the average cost was derived independently in Marbach and Tsitsiklis (2001) and Sutton *et al.* (1999). This expression leads naturally to policy gradient algorithms, cf. Bartlett and Baxter (2011). There is a corresponding discounted variant of this expression in Sutton *et al.* (1999), and the policy gradient technique in Bartlett and Baxter (2011). As in the discounted setting, incorporating function approximation for the functions J and U is straightforward, and we refer the reader to Prashanth and Ghavamzadeh (2016) for the case where a linear function approximation architecture is used.

6.3 Rockafellar and Uryasev (2000) first showed that both VaR and CVaR can be obtained from the solution of a certain convex optimization problem, so that a stochastic approximation-based procedure can be derived for estimating VaR and CVaR, as in Bardou *et al.* (2009), in turn leading to a specialization for MDPs in Prashanth (2014). For such a scheme, a non-asymptotic analysis is not available. However, stochastic gradient schemes have received a lot of attention from a non-asymptotic analysis viewpoint, see (Bottou *et al.*, 2018) for a survey. Since VaR estimation through (6.14) falls under the realm of stochastic gradient schemes, one could use the bounds in the aforementioned reference. As an alternative, one could consider a direct sample average approximation (SAA) of CVaR, see (Serfling, 2009). Concentration bounds for such an SAA-approximation of CVaR has received a lot of attention in past decade or so, cf. (Brown, 2007; Wang and Gao, 2010; Kolla *et al.*, 2019; Bhat and Prashanth, 2019; Thomas and Learned-Miller, 2019; Prashanth *et al.*, 2020). Such results would be useful for the analysis of Algorithm 3, considering that one needs to decide on the number of episodes m_n in each policy gradient iteration; see the discussion in Section 7.2.3.

The likelihood ratio-based gradient estimate for CVaR was derived by Tamar *et al.* (2014b) for the case of continuous distributions. An expression for the gradient of an abstract coherent risk measure, for both discrete and continuous distributions and also specialized to handle the case of CVaR, is derived in Tamar *et al.* (2015a). We shall present this expression as well as the specialization in Section 7.3.

7

MDPs with Risk as the Objective

In this section, we discuss policy gradient algorithms for solving risk-sensitive MDPs where the risk measure is explicitly the objective, i.e., the following optimization problem:

$$\min_{\theta \in \Theta} G(\theta),$$

where G is one of the risk measures presented in Section 3 that consider the entire distribution. Specifically, we consider exponential cost, CPT, and coherent risk measures in Sections 7.1, 7.2, and 7.3, respectively. This complements the incorporation of risk measures such as variance or CVaR that are based on the tail of the underlying distribution, which were considered in the constrained optimization formulations in the previous section.

Following the template in Section 5, the main ingredients in each iteration n of a policy gradient algorithm for optimizing a risk objective are as follows:

(i) Simulation of the underlying MDP to obtain one or more sample trajectories.

(ii) Estimation of $\nabla G(\theta_n)$ from the sample data.

(iii) Incremental update of the policy parameter in the descent direction using the gradient estimate from the step above, i.e.,

$$\theta_{n+1} = \Gamma\big[\theta_n - \zeta(n)\widehat{\nabla} G(\theta_n)\big],$$

where $\zeta(n)$ is the step size and Γ is a projection operator that keeps the iterate bounded.

7.1 Case 1: Average-cost MDP + Exponential cost as risk

In many cases studied earlier, the recipe for a risk-sensitive policy gradient algorithm is to first derive an expression for the gradient of the risk measure, and then use this expression to form an estimator using sample trajectories of the underlying MDP.

Recall from Section 3.1, under A2.3, the exponential cost associated with a policy μ^θ is given by

$$G(\theta) = \lim_{T\to\infty} \frac{1}{T}\frac{1}{\beta} \log \mathbb{E}\left[\exp\left(\beta \sum_{n=0}^{T-1} k(x_n, a_n)\right)\right], \tag{7.1}$$

where β is the risk-sensitivity parameter.

For the analysis, we also require aperiodicity in addition to irreducibility and positive recurrence, which we specify in the following variant of A2.3.

A7.1. For each policy μ^θ, the underlying Markov chain is irreducible, positive recurrent, and aperiodic.

Define the $|\mathcal{X}| \times |\mathcal{X}|$ matrix A_θ as

$$A_\theta \triangleq \frac{1}{\beta}\left[\sum_a \mu^\theta(a|x) \exp\left(\beta k(x,a)\right) P(y|x,a)\right]_{x,y\in\mathcal{X}}. \tag{7.2}$$

Since the Markov chain underlying μ^θ is assumed to be irreducible, and each entry of A_θ is non-negative, we can apply Perron-Frobenius theorem to infer that there exists a unique eigenvalue-eigenvector pair $(\lambda_\theta, V(\theta))$ satisfying

$\lambda_\theta > 0,\ V(\theta, i) > 0, \forall i$ and $|\lambda'| \le \lambda_\theta$ for any other eigenvalue λ' of A_θ.

Proposition 7.1. Assume A7.1 and that the state-action spaces of the underlying MDP are both finite. Then, for any policy μ^θ,

$$G(\theta) = \log \lambda_\theta,$$

where $G(\theta)$ is the exponential cost associated with the policy μ^θ given by (7.1) and λ_θ is the Perron-Frobenius eigenvalue of the matrix A_θ defined by (7.2).

Proof. Since $(\lambda_\theta, V(\theta))$ is an eigenvalue-eigenvector pair associated with the matrix A_θ, we have $A_\theta V(\theta) = \lambda_\theta V(\theta)$, or equivalently,

$$\lambda_\theta V(\theta, x) = \frac{1}{\beta} \sum_y \sum_a \mu^\theta(a|x) \exp\left(\beta k(x,a)\right) P(y|x,a) V(\theta, y).$$

Let

$$\tilde{P}_\theta(y|x) = \frac{\sum_a \mu^\theta(a|x) \exp\left(\beta k(x,a)\right) P(y|x,a) V(\theta, y)}{\beta \lambda_\theta V(\theta, x)}, \ \forall x, y \in \mathcal{X}. \tag{7.3}$$

Notice that $\tilde{P}_\theta(y|x) \geq 0$ and $\sum_y \tilde{P}_\theta(y|x) = 1$, implying $\tilde{P}_\theta(y|x)$ is a valid transition probability function.

Let $\{\tilde{x}_n\}$ be a Markov chain governed by the transition probability function $\tilde{P}_\theta$. Then, under A7.1, this Markov chain is irreducible, positive recurrent and aperiodic, which in turn implies the existence of a stationary distribution, say $\tilde{\psi}$. Thus, by ergodicity,

$$\mathbb{E}\left[h(\tilde{x}_n)\right] \to \sum_x \tilde{\psi}(x) h(x) \text{ a.s. as } n \to \infty.$$

Notice that

$$\frac{1}{T}\frac{1}{\beta} \log \mathbb{E}\left[\exp\left(\beta \sum_{n=0}^{T-1} k(x_n, a_n)\right) \middle| x_0\right]$$

$$= \frac{1}{T}\frac{1}{\beta} \log \left[\sum_{\substack{x_1,\ldots,x_T,\\ a_0,\ldots,a_{T-1}}} \prod_{n=0}^{T-1} \exp\left(\beta k(x_n, a_n)\right) \mu^\theta(x_n, a_n) P(x_{n+1}|x_n, a_n)\right]$$

$$= \frac{1}{\beta T} \times$$

$$\log\left[\sum_{\substack{x_1,\ldots,x_T,\\ a_0,\ldots,a_{T-1}}}\prod_{n=0}^{T-1}\frac{\exp\left(\beta k(x_n,a_n)\right)\mu^\theta(a_n|x_n)V(\theta,x_{n+1})P(x_{n+1}|x_n,a_n)}{\lambda_\theta V(\theta,x_n)}\right.$$
$$\left.\times\frac{\lambda_\theta V(\theta,x_0)}{V(\theta,x_T)}\right]$$
$$=\log\lambda_\theta+\frac{1}{T}\frac{1}{\beta}\left(\log V(\theta,x_0)-\log\mathbb{E}\left[V(\theta,\tilde{x}_T)\right]\right)\to\log\lambda_\theta\text{ as }T\to\infty.$$

The claim follows. □

Since the state and action spaces are assumed to be finite, we have $\lambda^*=\min_{\mu^\theta}\lambda_\theta$, where the minimum is taken over all randomized policies. Let V^* denote the corresponding eigenvector. Then it can be shown that

$$\lambda^*V^*(x)=\min_\mu\left(\frac{1}{\beta}\sum_a\mu(a|x)\exp(\beta k(x,a))\sum_y P(y|x,a)V^*(y)\right).\quad(7.4)$$

As in the case of risk-neutral average-cost MDPs, we can define Q-values that assist in solving the problem of control. For the exponential cost case, the optimal Q-value is defined as

$$Q^*(x,a)=\frac{\exp(\beta k(x,a))}{\beta\lambda^*}\sum_y P(y|x,a)V^*(y).$$

(Q^*,λ^*) is a solution, unique up to a scalar multiple, of the following eigenvalue problem:

$$Q^*(x,a)\ \lambda^*=\frac{\exp(\beta k(x,a))}{\beta}\sum_y P(y|x,a)\min_b Q^*(y,b).\qquad(7.5)$$

Notice that $V(x)=\min_a Q^*(x,a)$ satisfies (7.4).

For the special case of a fixed policy μ^θ, we define the Q-value analogue as

$$Q(\theta,x,a)=\frac{\exp(\beta k(x,a))}{\beta\lambda_\theta}\sum_y P(y|x,a)V(\theta,y).$$

$(Q^\theta,\lambda_\theta)$ satisfies the following eigenvalue problem:

$$Q(\theta,x,a)\ \lambda_\theta=\frac{\exp(\beta k(x,a))}{\beta}\sum_y P(y|x,a)\sum_b\mu(b|y)Q(\theta,y,b).\quad(7.6)$$

For the sake of consistent notation, we shall use $Q_\mu(\cdot,\cdot)$ and $V_\mu(\cdot)$ to denote the Q-value and the Perron-Frobenius eigenvector associated with a policy μ that is not necessarily parameterized.

The results in (7.5) and (7.6) can be used to derive value and policy iteration algorithms for finding a policy that optimizes the exponential cost. We present the policy iteration algorithm, as it forms the basis for the risk-sensitive policy gradient algorithm presented later in this section.

Policy iteration for exponential cost

Initialization: Policy μ_0, fixed state x_f.

For all $n = 1, 2, \ldots,$ **repeat**

Policy evaluation: Solve the eigenvalue problem

$$V_n(x) = \sum_a \mu_n(a|x) \frac{\exp(\beta k(x,a))}{\beta \lambda_n} \sum_y P(y|x,a) V_n(y),$$
$$V_n(x_f) = 1. \tag{7.7}$$

Policy improvement: Choose action according to

$$\mu_{n+1}(\cdot|x) \in \arg\min_\mu \left[\sum_a \mu_n(a|x) \frac{\exp(\beta k(x,a))}{\beta} \sum_y P(y|x,a) V_n(y) \right].$$

The policy evaluation step in the algorithm above can be performed in an iterative fashion as follows: Initializing with $V_n^0 = V_{n-1}$, update

$$\tilde{V}_n^{m+1}(x) = \sum_a \mu_n(a|x) \frac{\exp(\beta k(x,a))}{\beta} \sum_y P(y|x,a) V_n^m(y),$$
$$V_n^{m+1}(x) = \frac{\tilde{V}_n^{m+1}(x)}{\tilde{V}_n^{m+1}(x_f)}.$$

The above variation of policy evaluation can be seen as value iteration for a fixed policy. Such a scheme can be shown to converge, and the limit coincides with the solution to the eigenvalue problem in (7.7).

Next, the policy iteration algorithm could be written using Q-values. Such an algorithm is not really necessary in a context where the transition dynamics of the underlying MDP is known. However, the actor-critic

algorithm presented subsequently could be seen as a learning variant of the Q-value based policy iteration, which we present next.

Policy iteration using Q-values

Initialization: Policy μ_0, fixed state x_f and action a_f.

For all $n = 1, 2, \ldots$, **repeat**

Policy evaluation: With $Q_n^0 = Q_{n-1}$, update (until convergence)

$$\tilde{Q}_n^{m+1}(x,a) = \sum_a \frac{\exp(\beta k(x,a))}{\beta} \sum_y P(y|x,a) \sum_b \mu_n(b|y) Q_n^m(y,b),$$

$$Q_n^{m+1}(x,a) = \frac{\tilde{Q}_n^{m+1}(x,a)}{\tilde{Q}_n^{m+1}(x_f,a_f)}.$$

Policy improvement: Choose action according to

$$\mu_{n+1}(\cdot|x) \in \arg\min_{\mu} \left[\sum_a \mu_n(a|x) Q_n^{m+1}(x,a) \right].$$

The result below presents a variant of the policy gradient theorem for the exponential cost risk measure.

Proposition 7.2. Assume A7.1. Then,

$$\nabla \lambda_\theta = \sum_{x,a} \tilde{\psi}_\theta(x) \nabla \mu^\theta(a|x) \tilde{Q}(\theta,x,a) \lambda_\theta, \tag{7.8}$$

$$\text{where } \tilde{Q}(\theta,x,a) = \frac{\exp\left(\beta k(x,a)\right)}{\beta V(\theta,x) \lambda_\theta} \sum_y P(y|x,a) V(\theta,y)$$

is the modified Q-value function, and $\tilde{\psi}_\theta$ is the stationary distribution underlying a Markov chain governed by the transition probability function $\tilde{P}_\theta(\cdot|\cdot)$ defined in (7.3).

Proof. Letting $V(\theta)$ denote the eigenvector corresponding to λ_θ, the eigenvalue equation can be written as

$$\lambda_\theta V(\theta, x) = \frac{1}{\beta} \sum_y \sum_a \mu^\theta(a|x) \exp(\beta k(x, a)) P(y|x, a) V(\theta, y),$$

or equivalently,

$$V(\theta, x) = \sum_a \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta} \sum_y P(y|x, a) V(\theta, y). \tag{7.9}$$

Setting $V(\theta, x_0) = 1$ would ensure that the solution $V(\theta)$ to (7.9) is unique.

Differentiating w.r.t. θ in (7.9), we obtain

$$\begin{aligned}
\nabla V(\theta, x) = & \sum_a \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta} \sum_y P(y|x, a) \nabla V(\theta, y) \\
& + \sum_a \nabla \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta} \sum_y P(y|x, a) V(\theta, y) \\
& - \sum_a \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta^2} \sum_y P(y|x, a) V(\theta, y) \nabla \lambda_\theta.
\end{aligned}$$

Dividing by $V(\theta, x)$ and then summing over the stationary distribution $\tilde{\psi}_\theta$ on both sides of the equation above, we obtain

$$\begin{aligned}
& \sum_x \tilde{\psi}_\theta(x) \frac{\nabla V(\theta, x)}{V(\theta, x)} \\
& = \underbrace{\sum_x \tilde{\psi}_\theta(x) \sum_a \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta V(\theta, x)} \sum_y P(y|x, a) \nabla V(\theta, y)}_{(I)} \\
& + \underbrace{\sum_x \tilde{\psi}_\theta(x) \sum_a \nabla \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta \lambda_\theta V(\theta, x)} \sum_y P(y|x, a) V(\theta, y)}_{(II)} \\
& - \underbrace{\sum_x \tilde{\psi}_\theta(x) \sum_a \mu^\theta(a|x) \frac{\exp(\beta k(x, a))}{\beta V(\theta, x) \lambda_\theta^2} \sum_y P(y|x, a) V(\theta, y) \nabla \lambda_\theta}_{(III)}. \qquad (7.10)
\end{aligned}$$

Using the fact that $\tilde{P}_\theta$, defined in (7.3), is a transition probability function, and also that $\tilde{\psi}_\theta$ is a stationary distribution, we simplify each

of the terms on the RHS of (7.10) as follows:

$$(I) = \sum_x \tilde{\psi}_\theta(x) \sum_y \tilde{P}(y|x) \frac{\nabla V(\theta, y)}{V(\theta, y)} = \sum_x \tilde{\psi}_\theta(x) \frac{\nabla V(\theta, x)}{V(\theta, x)},$$
$$(II) = \sum_x \tilde{\psi}_\theta(x) \sum_a \nabla \mu^\theta(a|x) \tilde{Q}(\theta, x, a),$$
$$(III) = \sum_y \frac{\nabla \lambda_\theta}{\lambda_\theta} \tilde{P}(y|x) = \frac{\nabla \lambda_\theta}{\lambda_\theta}.$$

From the above simplifications, it is apparent that term (I) is the same as the LHS in (7.10). A reordering of the simplified terms (II) and (III) leads to

$$\frac{\nabla \lambda_\theta}{\lambda_\theta} = \sum_{x,a} \tilde{\psi}_\theta(x) \nabla \mu^\theta(a|x) \tilde{Q}(\theta, x, a).$$

The claim follows. □

Policy gradient algorithm for exponential cost

In the case of the regular value function, the policy gradient theorem (6.1) lends itself to an RL algorithm easily, since one can replace the expectation on the RHS of (6.1) with a sample trajectory-based approximation. On the other hand, the formula in (7.8) is complicated from a sampling viewpoint because the distribution on the RHS of (7.8) is different from the transition dynamics underlying the given MDP. To elaborate, the averaging in the policy gradient expression for exponential cost involves the stationary distribution $\tilde{\psi}_\theta$ that underlies the Markov chain governed by transition probabilities $\tilde{P}(\cdot|\cdot)$, and it is not practically feasible to obtain samples from this distribution. However, one can develop a policy gradient-type algorithm without a compact representation, i.e., by treating the policy as a vector of probabilities.

The main idea that leads to the policy update that we describe next is the following: Letting $\Lambda_\theta = \log \lambda_\theta$, we have the following variant of (7.8):

$$\nabla \Lambda_\theta = \sum_{x,a} \tilde{\psi}_\theta(x) \nabla \mu^\theta(a|x) \tilde{Q}(\theta, x, a), \tag{7.11}$$

Since log is monotone, the minimizer of λ_θ coincides with that of Λ_θ. Next, treating the policy as a probability vector over all states and all but one action, i.e., $\mu^\theta = [\mu^\theta(x,a)]_{x\in\mathcal{X},a\in\mathcal{A}\setminus a_f}$, where a_f is a fixed action. Given μ^θ, we can infer the probability of choosing action a_f in state x as follows:

$$\mu^\theta(a_f|x) = 1 - \sum_{a\neq a_f} \mu^\theta(a|x). \tag{7.12}$$

The component of the RHS of (7.11) corresponding to state-action pair (x,a) is

$$\begin{aligned}\frac{\partial \Lambda_\theta}{\partial \mu^\theta(a|x)} &= \tilde{\psi}_\theta(x)\nabla\mu^\theta(a|x)\tilde{Q}(\theta,x,a) + \tilde{\psi}_\theta(x)\nabla\mu^\theta(a_f|x)\tilde{Q}(\theta,x,a_f)\\ &= \tilde{\psi}_\theta(x)\left(\tilde{Q}(\theta,x,a) - \tilde{Q}(\theta,x,a_f)\right).\end{aligned} \tag{7.13}$$

The equalities above follow from (7.12). Thus, it is enough to use the factor $\left(\tilde{Q}(x,a) - \tilde{Q}(x,a_f)\right)$ to perform a gradient descent in the policy, while ignoring the multiplicative factor $\tilde{\psi}_\theta(x)$, which is not available in practice for an RL algorithm.

We next describe an algorithm that performs such a gradient descent update in the policy space. Notice that, for the policy update, one would require $\tilde{Q}(\theta,\cdot,\cdot)$, and the algorithm estimates this modified Q-value on the faster timescale, while performing the policy update on the slower timescale. The two-timescale algorithm requires varying step-size sequences $\{\zeta_1(n),\zeta_2(n)\}$ to satisfy the following conditions:

$$\sum_{n=1}^{\infty}\zeta_1(n) = \sum_{n=1}^{\infty}\zeta_2(n) = \infty, \sum_{n=1}^{\infty}\left(\zeta_1^2(n)+\zeta_2^2(n)\right) < \infty, \frac{\zeta_1(n)}{\zeta_2(n)} \to 0.$$

Letting $d(n)$ denote either of the step sizes $\zeta_1(n)$ and $\zeta_2(n)$, $\forall\ z\in(0,1)$,

$$\sup_n \frac{d(\lceil zn\rceil)}{d(n)} < \infty, \text{ and } \sup_n \frac{A(\lceil z'n\rceil)}{A(n)} \to 1 \text{ uniformly in } z'\in[z,1],$$

$$\text{where } A(n) = \sum_{m=0}^{n} d(m).$$

The first set of conditions on the step-size sequences are standard for two-timescale stochastic approximation, which in particular ensure

that $\zeta_2(n)$ would be on the slower timescale, while $\zeta_1(n)$ on the faster timescale. The additional conditions ensure that the both step sizes eventually decrease.

Since we treat the policy as a vector indexed by state-action pairs, we drop the dependence on the parameter θ, and instead work with a randomized policy iterate, say μ_n. Using step-size sequences satisfying these conditions, the algorithm for optimizing exponential cost would update along two timescales, with fixed $x_f \in \mathcal{X}, a_f \in \mathcal{A}$, as follows:

$$Q_{n+1}(x,a) = Q_n(x,a) + \zeta_1(\nu(x,a,n))\mathbb{I}\{x_n = x, a_n = a\} \times \left(\frac{\exp(\beta k(x,a)) Q_n(x_{n+1}, a_{n+1})}{\beta Q_n(x_f, a_f)} - Q_n(x,a) \right), \quad (7.14)$$

$$\mu_{n+1}(x) = \Gamma\left(\mu_n(x) + \zeta_2(\nu(x,a,n))\mathbb{I}\{x_n = x, a_n = a\} \times (Q_n(x,a) - Q_n(x,a_f))\right), \quad (7.15)$$

where $\nu(x,a,n) = \sum_{m=0}^{n} \mathbb{I}\{x_m = x, a_m = a\}$ denote the number of times the state-action pair (x,a) has been visited up to time n, and Γ is a projection operator that ensures that, for any $x \in \mathcal{X}$, the updated policy $\mu_{n+1}(x)$ stays within the simplex $\{(d_1, \ldots, d_{|\mathcal{A}|-1}) \mid d_i \geq 0, \forall i = 1, \ldots, |\mathcal{A}| - 1, \sum_{j=1}^{|\mathcal{A}|-1} d_j \leq 1\}$. The policy vector is updated for all but one action a_f, and the probability associated with this action can be inferred using (7.12). Furthermore, the actions a_n are picked using an ϵ-greedy randomized policy, i.e., w.p. $(1-\epsilon)$ choose an action according to μ_n, and w.p. ϵ pick a random action. Here $\epsilon \in (0,1)$ is an exploration parameter, that is chosen to be small constant, or could be decayed as the algorithm updates.

The faster timescale recursion in (7.14) can be seen as the stochastic approximation variant of the policy evaluation step in the policy iteration using Q-values. In other words, following the standard two timescale viewpoint to assume the policy μ is quasi-static, the faster timescale iterate Q_n converges to Q_μ. Next, viewing the faster timescale as almost equilibrated, the slower timescale recursion can be seen to perform a gradient step in the policy space with an exponential cost objective. The policy increment in (7.15) is motivated by the discussion around

(7.13), and the remark below uses an ODE argument to show that one can ignore the positive multiplicative factor in the gradient expression, and still converge to a local minimum of the exponential cost objective. This argument is facilitated by the fact that we treat the policy as a vector of probabilities over all states and all but one action.

Remark 7.1. The update iteration for $\mu(i,u)$ in (7.15) is separate for each fixed i, with the following limiting ODE:

$$\dot{\mu}(i,\cdot) = -\left(Q_\mu(i,\cdot) - Q_\mu(i,a_f)\right). \tag{7.16}$$

The gradient descent would have been

$$\dot{\mu}(i,\cdot) = -\tilde{\psi}(i)(\tilde{Q}_\mu(i,\cdot)) - \tilde{Q}_\mu(i,a_f) = \frac{\tilde{\psi}(i)}{V_\mu(i)}(Q_\mu(i,\cdot) - Q_\mu(i,a_f)).$$

Thus, (7.16) is of the form

$$\dot{x}_i(t) = -C_i \nabla f(x(t)), \forall i,$$

with $C_i > 0,\ \forall i$. This still converges to a stationary point for almost all initial conditions, because

$$\frac{d}{dt} f(x(t)) = -\sum_i C_i \left\|\nabla f(x(t))\right\|^2 < 0,$$

except at critical points.

Notes on convergence

Unlike the other cases studied in this section, the analysis of Algorithm 5 does not conform to the template in Section 5.3, because the policy update in (7.15) treats the policy as a vector over all states and actions. If one uses a compact representation, say via a parameterized family of policies μ^θ, then there are two challenges involved in arriving at a policy gradient algorithm. The first relates to sampling. As mentioned before, the policy gradient expression in (7.8) involves a distribution that is different from the transition dynamics underlying the MDP considered. The second relates to projection. With a parameterized policy class, the projection Γ onto the probability simplex should be

Algorithm 5: Policy gradient algorithm under exponential cost as a risk measure in an average-cost MDP setting

Input : initial policy μ_0, step sizes $\{\zeta_1(n)\}$, $\{\zeta_2(n)\}$, exploration parameter ϵ, projection operator Γ, # iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**

Draw action $a_n \sim \mu_n(\cdot|x_n)$ w.p. $(1-\epsilon)$ and pick a random action w.p. ϵ;

Observe next state x_{n+1} and cost $k(x_n, a_n)$;

`/* Q-value estimate */`

$$Q_{n+1}(x,a) = Q_n(x,a) + \zeta_1(\nu(x,a,n))\mathbb{I}\{x_n = x, a_n = a\} \times \left(\frac{\exp(\beta k(x,a))Q_n(x_{n+1},a_{n+1})}{Q_n(x_f,a_f)} - Q_n(x,a)\right);$$

`/* Policy update */`

$$\mu_{n+1}(x) = \Gamma\left(\mu_n(x) + \zeta_2(\nu(x,a,n))\mathbb{I}\{x_n = x, a_n = a\} \times (Q_n(x,a) - Q_n(x_f,a))\right);$$

end

Output : Policy μ_M

replaced by a projection onto the set $\{\mu^\theta(\cdot|\cdot), \theta \in \Theta\}$, where Θ is the set of parameterized policies. However, the aforementioned set of probability vectors used for projection need not be convex.

We now provide a sketch of the convergence analysis for Algorithm 5. Recall that Algorithm 5 employs two-timescale stochastic approximation, i.e., it comprises of iteration sequences that are updated using two different step-size schedules defined via $\{\zeta_1(n)\}$ and $\{\zeta_2(n)\}$, respectively. The analysis follows a standard sequence of steps needed to show convergence of two-timescale stochastic approximation algorithms, as discussed in Section 5.4. In particular, the faster timescale analysis for the modified Q-value estimate sees the policy update as quasi-static, while the slower timescale analysis for the policy μ views the modified Q-value updates to have converged.

For the analysis of the faster timescale $\tilde{Q}$-recursion, consider the following ODE: $\forall x \in \mathcal{S}, i = 1, 2, \ldots, N$,

$$\dot{\varphi}_{xa}(t) = \frac{1}{|\mathcal{X}||\mathcal{A}|+1}\left(\frac{\exp(\beta k(x,a))\sum_y P(y|x,a)\sum_b \mu(b|y)\varphi_{yb}(t)}{\beta\varphi_{x_f a_f}(t)} - \varphi_{xa}(t)\right). \tag{7.17}$$

where μ is considered to be time-invariant, since it is updated on the slower timescale. It can be shown that (7.17) has a globally asymptotically stable equilibrium Q_μ, which is the unique solution to the following system of equations:

$$Q(x,a) = \left(\frac{\exp(\beta k(x,a))\sum_y P(y|x,a)\sum_b \mu(b|y)Q(y,b)}{\beta\lambda_\mu}\right), \ \forall x,a,$$
$$Q(x_f,a_f) = \lambda_\mu.$$

Using standard stochastic approximation arguments together with a stability result in Theorem 4.4, for any given policy μ, the faster timescale iterate Q_n converges a.s. to Q_μ.

We now turn our attention to the policy update (7.15). Consider the following ODE:

$$\dot{\chi}_{x\cdot}(t) = Q^{\chi(t)}(x,\cdot) - Q^{\chi(t)}(x,a_f), \forall x \in \mathcal{X} \tag{7.18}$$

This ODE can be re-written as

$$\dot{\chi}_{x\cdot}(t) = K_{x\cdot}(\chi(t)) - K_{xa_f}(\chi(t)), \tag{7.19}$$

where $K_{xa}(\mu) = Q_\mu(x,a) - V_\mu(x)$ is the advantage function. From the definitions of Q_μ and V_μ, it is apparent that $\sum_a \mu(a|x)K_{xa}(\mu) = 0$. Notice that the unique stable point of the ODE (7.19) corresponds to an optimal policy, since otherwise, the advantage function is not zero; thus, one can establish that the trajectories of (7.19) would converge to the set of optimal policies minimizing the exponential cost, with $\Lambda(\cdot)$ serving as a strict Lyapunov function. Finally, the policy update in (7.15) used ϵ-greedy exploration, which implies that the ODE tracked by this algorithm would not exactly coincide with (7.18). Instead, the algorithm tracks the ODE (7.18) with an small error for small ϵ, and hence one can claim that the iterate μ_n converges to a neighbourhood of the set of risk-optimal policies.

7.2 Case 2: Discounted-cost/SSP + CPT as risk

This case considers the following optimization problem with CPT risk measure as the objective:

$$\min_{\theta \in \Theta} \mathcal{C}(D^\theta),$$

where $\mathcal{C}(D^\theta)$ is the CPT-value associated with the r.v. D^θ, with θ denoting the policy parameter, and is defined as follows:

$$\mathcal{C}(D^\theta) \triangleq \int_0^\infty w^+ \left(\mathbb{P}\left(u^+(D^\theta) > z\right)\right) dz - \int_0^\infty w^- \left(\mathbb{P}\left(u^-(D^\theta) > z\right)\right) dz, \tag{7.20}$$

where $u^\pm$ are the utility functions and $w^\pm$ are the weight functions (see Section 3.7 for a detailed description of these quantities). In a discounted-cost MDP, $D^\theta(x_0)$ would be the total discounted cost, while in an SSP, $D^\theta(x_0)$ would be the total cost.

Risk measures such as variance, CVaR, and chance constraints considered in the previous section serve as natural constraint functions when optimizing the expected total/discounted cost. On the other hand, CPT is a risk measure that is not purely concerned with the tail behavior or variability of the cost distribution, as it considers the entire distribution, handling gains/losses, as well as incorporating distortions via a weight function. Hence, it is appealing to optimize the CPT value directly in the objective, e.g., in a human-centered decision-making problem. For a concrete example, one could consider a transportation application where D^θ denotes the delay experienced by a road user, and θ denotes a policy parameter that governs the traffic light switching strategy.

From the discussion in the previous sections, it is apparent that the main technical challenges in handling any risk measure are as follows: (i) estimation of the risk measure from samples; and (ii) gradient estimation for the policy update iteration. For the sake of brevity, we provide the necessary details for handling (i) and (ii), and the rest of the pieces of the resulting actor-critic scheme follows in a manner similar to that for variance or CVaR.

7.2.1 CPT-value estimation

To handle (i), suppose that we are given m i.i.d. samples from the distribution of X, and the goal is to estimate the CPT-value $\mathcal{C}(X)$. Estimating the CPT-value is challenging, because the environment provides samples from the distribution of the r.v. X, while the integrals in (7.20) involve a weight-distorted distribution. Thus, unlike the case of expected value estimation, a sample mean is insufficient for estimating CPT-value $\mathcal{C}(X)$, when the underlying weight functions are nonlinear. CPT-value $\mathcal{C}(X)$ can be estimated if one has an estimate of the *entire* distribution, and a natural candidate to estimate the distribution is the empirical distribution function (EDF). Using the latter, we estimate $\mathcal{C}(X)$ by

$$\overline{\mathcal{C}}_m = \int_0^\infty w^+\left(1 - \hat{F}_m{}^+(x)\right) dx - \int_0^\infty w^-\left(1 - \hat{F}_m{}^-(x)\right) dx, \quad (7.21)$$

where

$$\hat{F}_m{}^+(x) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left\{(u^+(X_i) \le x)\right\} \text{ and}$$
$$\hat{F}_n{}^-(x) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left\{(u^-(X_i) \le x)\right\}.$$

$\hat{F}_m{}^+(x)$ and $\hat{F}_m{}^-(x)$ are the EDFs of the r.v.s $u^+(X)$ and $u^-(X)$, respectively. The first and second integrals on the RHS of (7.21) denoted by $\overline{\mathcal{C}}_m^+$ and $\overline{\mathcal{C}}_m^-$, respectively, can be computed in a straightforward fashion using the order statistics $X_{[1]} \le X_{[2]} \le \ldots \le X_{[m]}$ as follows:

$$\overline{\mathcal{C}}_m^+ := \sum_{i=1}^{m} u^+(X_{[i]}) \left(w^+\left(\frac{m+1-i}{m}\right) - w^+\left(\frac{m-i}{m}\right)\right),$$
$$\overline{\mathcal{C}}_m^- := \sum_{i=1}^{m} u^-(X_{[i]}) \left(w^-\left(\frac{i}{m}\right) - w^-\left(\frac{i-1}{m}\right)\right).$$

Notice that the estimates $\overline{\mathcal{C}}_m^{\pm}$ reduce to sample means for the case when $w(p) = p$, and in this case, the CPT-value itself is the expectation $\mathbb{E}u^+(X) - \mathbb{E}u^-(X)$. Thus, CPT-value estimation can be seen as a generalization of the classic mean estimation procedure, and the deviations

are introduced by a nonlinear weight function that distorts probabilities, which in turn leads to weighing the samples non-uniformly.

7.2.2 Policy gradient for the CPT-value

As far as handling point (ii) concerning the policy gradient for CPT, we use SPSA, since the CPT-value does not admit a Bellman equation, ruling out a procedure based on the likelihood ratio method. The SPSA-based estimate of $\nabla \mathcal{C}(X^{\theta_n})$ with policy θ_n, is given as follows:

$$\widehat{\nabla}_i \mathcal{C}(D^{\theta}) = \frac{\overline{\mathcal{C}}_n^{\theta_n + \delta_n \Delta(n)} - \overline{\mathcal{C}}_n^{\theta_n}}{\delta_n \Delta_i(n)}, i = 1, \ldots, d,$$

where δ_n and $\Delta(n)$ are as described in Section 4.4 and $\overline{\mathcal{C}}_n^{\theta_n + \delta_n \Delta(n)}$ (resp. $\overline{\mathcal{C}}_n^{\theta_n}$) denotes the CPT-value estimate that uses m_n samples of the r.v. $X^{\theta_n + \delta_n \Delta(n)}$ (resp. X^{θ_n}).

The complete algorithm with CPT-value as the risk measure and the usual value function as the objective is presented in Algorithm 6.

7.2.3 On the batch size m_n per iteration of (7.22)

The challenge involved in choosing an appropriate batch size m_n for policy evaluation in Step 2 of Algorithm 6 is similar to that in Algorithm 1 for optimizing variance as a constraint in a discounted MDP. As in the variance case, the CPT-value has to be estimated from sample trajectories so that the overall policy gradient algorithm (7.22) converges. For the sake of simplicity, we drop the dependence on the parameter θ, and instead, study the CPT-value estimation problem. Subsequently, when we analyze the policy gradient scheme in Algorithm 6 for CPT-value optimization, we shall make the dependence on the policy parameter explicit.

For a given r.v. X, let m denote the number of sample trajectories used to form the estimate $\overline{\mathcal{C}}_m$, using (7.21), of the CPT-value $\mathcal{C}(X)$. Notice that $\mathbb{E}\left(\overline{\mathcal{C}}_m\right) \neq \mathcal{C}(X)$, since the individual components $\overline{\mathcal{C}}_m^{\pm}$ involve order statistics. However, one can derive a bound on the estimation error $\left|\overline{\mathcal{C}}_m - C(X)\right|$, and such a bound would aid the proof of asymptotic

Algorithm 6: Policy gradient algorithm under CPT as a risk measure

Input : initial parameter $\theta_0 \in \Theta$, perturbation constants $\delta_n > 0$, batch sizes $\{m_n\}$, step sizes $\{\zeta(n)\}$, projection operator Γ, number of iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**

  for $m \leftarrow 0$ **to** $m_n - 1$ **do**

    `/* Unperturbed policy simulation */`

    Use the policy μ^{θ_n} to generate the state x_m, draw action $a_m \sim \mu^{\theta_n}(\cdot|x_m)$;

    Observe next state x_{m+1} and cost $k(x_m, a_m)$;

    `/* Perturbed policy simulation */`

    Use the policy $\mu^{\theta_n+\delta_n\Delta(n)}$ to generate the state x_m^+, draw action $a_m^+ \sim \mu^{\theta_n+\delta_n\Delta(n)}(\cdot|x_m^+)$;

    Observe next state x_{m+1}^+ and cost $k(x_m^+, a_m^+)$;

  end

  `/* Monte Carlo policy evaluation */`

  Use the scheme in (7.21) to obtain $\overline{\mathcal{C}}_n^{\theta_n+\delta_n\Delta(n)}$ and $\overline{\mathcal{C}}_n^{\theta_n)}$ - the estimates of the CPT-values $\mathcal{C}\left(X^{\theta_n+\delta_n\Delta(n)}\right)$ and $\mathcal{C}\left(X^{\theta_n}\right)$, respectively;

  `/* Gradient estimates using SPSA */`

  Gradient of the objective: $\widehat{\nabla}_i \mathcal{C}(X^{\theta_n}) = \dfrac{\overline{\mathcal{C}}_n^{\theta_n+\delta_n\Delta(n)} - \overline{\mathcal{C}}_n^{\theta_n}}{\delta_n \Delta_i(n)}$;

  `/* Policy update: Gradient descent using SPSA */`

$$\theta_{n+1} = \Gamma\left[\theta_n - \zeta(n)\left(\widehat{\nabla}\mathcal{C}(X^{\theta_n})\right)\right]. \tag{7.22}$$

end

Output : Policy θ_M

convergence of the risk-sensitive policy gradient algorithm for CPT. The following result presents a bound in expectation for the estimation error.

Proposition 7.3. Assume that the weight functions $w^{\pm}$ are Hölder continuous with common order α and constant H, i.e.,

$$\sup_{x \neq y} \frac{|w^{\pm}(x) - w^{\pm}(y)|}{|x - y|^{\alpha}} \leq H, \forall x, y \in [0, 1].$$

Suppose that the utility functions u^{+} and $-u^{-}$ are continuous and non-decreasing on their support $\mathbb{R}^{+}$ and $\mathbb{R}^{-}$, respectively. Furthermore, the utilities $u^{+}(X)$ and $u^{-}(X)$ are bounded by a constant M. Then, $\forall \epsilon > 0$, we have

$$\mathbb{P}\left(\left|\overline{\mathcal{C}}_n - \mathcal{C}(X)\right| \geq \epsilon\right) \leq 2e^{-2n\left(\frac{\epsilon}{HM}\right)^{\frac{2}{\alpha}}},$$

and

$$\mathbb{E}\left|\overline{\mathcal{C}}_n - \mathcal{C}(X)\right| \leq \frac{(8HM)\,\Gamma\left(\alpha/2\right)}{n^{\alpha/2}}. \tag{7.23}$$

Proof. The proof requires the well-known Dvoretzky–Kiefer–Wolfowitz (DKW) inequality, which shows concentration of the empirical distribution function around the true distribution. We recall this result below.

DKW inequality: Let F denote the cdf of r.v. U and $\hat{F}_n(u) = \frac{1}{n}\sum_{i=1}^{n} I_{[U_i \le u]}$ denote the empirical distribution of U, with $U_1, \ldots, U_n$ sampled from F. Then, for any $\epsilon > 0$, we have

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}} |\hat{F}_n(x) - F(x)| > \epsilon\right) \le 2e^{-2n\epsilon^2}.$$

Notice that

$$\begin{aligned}
&\left|\int_0^\infty w^+\left(\mathbb{P}\left(u^+(X) > t\right)\right) dt - \int_0^\infty w^+\left(1 - \hat{F}_n^+(t)\right) dt\right| \\
&= \left|\int_0^M w^+\left(\mathbb{P}\left(u^+(X) > t\right)\right) dt - \int_0^M w^+\left(1 - \hat{F}_n^+(t)\right) dt\right| \\
&\le HM \sup_{x\in\mathbb{R}} \left|\mathbb{P}\left(u^+(X) < t\right) - \hat{F}_n^+(t)\right|^\alpha.
\end{aligned}$$

Now, plugging in the DKW inequality, we obtain

$$\begin{aligned}
&P\left(\left|\int_0^\infty w^+\left(\mathbb{P}\left(u^+(X) > t\right)\right) dt - \int_0^\infty w^+\left(1 - \hat{F}_n^+(t)\right) dt\right| > \epsilon\right) \\
&\le P\left(HM \sup_{t\in\mathbb{R}} \left|(P(u^+(X) < t) - \hat{F}_n{}^+(t)\right|^\alpha > \epsilon\right) \le 2e^{-2n\left(\frac{\epsilon}{HM}\right)^{\frac{2}{\alpha}}}.
\end{aligned} \tag{7.24}$$

To derive the bound in expectation in (7.23), we integrate the high-probability bound (7.24) to obtain

$$\begin{aligned}
\mathbb{E}\left|\overline{\mathcal{C}}_n - \mathcal{C}(X)\right| &\le \int_0^\infty \mathbb{P}\left(\left|\overline{\mathcal{C}}_n - \mathcal{C}(X)\right| \ge \epsilon\right) d\epsilon \\
&\le 4\int_0^\infty \exp\left(-2n\left(\epsilon/HM\right)^{2/\alpha}\right) d\epsilon \le \frac{8HM\Gamma\left(\alpha/2\right)}{n^{\alpha/2}}.
\end{aligned}$$

□

Now, the discussion in Section 7.2.3 applies to the case of CPT-value, with a minor changes. In particular, from Proposition 7.3, the estimation bias for the case of CPT-value is of the order $\frac{1}{m^{\alpha/2}}$, where α is the Hölder exponent of the weight functions underlying CPT-value definition.

Following arguments similar to those employed in Section 7.2.3, using a SPSA-based gradient estimator for CPT-value would require the batch size m_n to diverge so that the estimation bias does not affect

convergence of policy gradient algorithm (7.22). In addition to the usual conditions on the step-size sequence and perturbation constant δ_n, one possible choice for m_n that ensures that the bias in the gradient estimate vanishes and the overall algorithm converges is the following: $\frac{1}{m_n^{\alpha/2}\delta_n} \to 0$.

7.3 Case 3: Any MDP + a coherent risk measure

In this section, we consider optimizing a coherent risk measure. Let $(\Omega, \mathcal{F}, \mathcal{P}_\theta)$ denote a probability space, where $\mathcal{P}_\theta$ denotes a parameterized probability measure, with θ as the parameter that belongs to a convex and compact set $\Theta \subset \mathbb{R}^d$. Let D be a r.v. with a finite mean, and let $\rho(D)$ denote its coherent risk measure. We consider the following problem:

$$\min_{\theta \in \Theta} \rho(D), \tag{7.25}$$

As in the previous section, D could be the total/discounted cost of the policy parameterized by θ.

The overall algorithm for optimizing the coherent risk measure is given in Algorithm 7. The schema of this algorithm resembles the one used in Algorithm 6 for optimizing CPT value, the difference being in the way the coherent risk measure is estimated, and its gradient computed. We elaborate on these two aspects below.

For coherent risk measure estimation, we require the dual representation. Let $\mathbb{E}(D) = \int_\Omega D(\omega) d\mathcal{P}_\theta(\omega)$ denote the expectation of a given r.v. D. Let $\mathfrak{P} = \{\xi \mid \int \xi d\mathcal{P}_\theta = 1\}$ denote the set of probability densities. Then there exists a convex and compact subset $\mathfrak{U}$ of $\mathfrak{P}$ such that

$$\rho(D) = \sup_{\xi \in \mathfrak{U}} \left\{ \mathbb{E}(\xi D) = \int_\Omega \xi(\omega) D(\omega) d\mathcal{P}_\theta(\omega) \right\}. \tag{7.26}$$

We shall refer to the set $\mathfrak{U}$ as the risk envelope associated with a coherent risk measure. For the case of CVaR, the risk envelope $\mathfrak{U}$ can be shown to be

$$\mathfrak{U} = \left\{ \xi \;\middle|\; \xi(\omega) \in \left[0, \frac{1}{1-\beta}\right], \mathbb{E}[\xi] = 1 \right\}. \tag{7.27}$$

Let $\mathcal{P}_{n,\theta}$ denote the empirical distribution function formed from n i.i.d. samples $\{\omega_1, \ldots, \omega_n\}$. Then, the estimate $\hat{\rho}_n$ of the coherent risk measure $\rho(D)$ is formed as follows:

$$\hat{\rho}_n = \sup_{\xi \in \mathfrak{U}} \sum_{i=1}^{n} \xi(\omega_i) D(\omega_i) \mathcal{P}_{n,\theta}(\omega_i). \tag{7.28}$$

Next, we turn to the estimate of the gradient of a coherent risk measure. For the purpose of gradient estimation, we shall assume the following form for the risk envelope:

$$\begin{aligned} \mathfrak{U}(\mathcal{P}_\theta) = \{ \xi \mid & g_{k_1}(\xi, \mathcal{P}_\theta) = 0,\ k_1 = 1, \ldots, K_1, \\ & f_{k_2}(\xi, \mathcal{P}_\theta) \leq 0,\ k_2 = 1, \ldots, K_2,\ \mathbb{E}[\xi] = 1,\ \xi(\omega) \geq 0 \}, \end{aligned}$$

where g_{k_1}, $k_1 = 1, \ldots, K_1$ and f_{k_2}, $k_2 = 1, \ldots, K_2$ are the equality and inequality constraints. Using the above form for the risk envelope, the Lagrangian of (7.26) turns out to be

$$\begin{aligned} L_\theta(\xi, \eta, \lambda, \tilde{\lambda}) = \mathbb{E}[\xi D] - \eta(\mathbb{E}[\xi] - 1) - \sum_{k_1=1}^{K_1} \lambda(k_1) g_{k_1}(\xi, \mathcal{P}_\theta) \\ - \sum_{k_2=1}^{K_2} \tilde{\lambda}(k_2) f_{k_2}(\xi, \mathcal{P}_\theta), \end{aligned} \tag{7.29}$$

where η is the Lagrange multiplier associated with the $\mathbb{E}[\xi] = 1$ constraint. Furthermore, $\lambda = (\lambda(1), \ldots, \lambda(K_1))$ and $\tilde{\lambda} = (\tilde{\lambda}(1), \ldots, \tilde{\lambda}(K_2))$ are the Lagrange multipliers associated with equality constraints defined by $(g_1, \ldots, g_{K_1})$, and inequality constraints defined by $(f_1, \ldots, f_{K_2})$, respectively.

We now present the expression for the gradient of the coherent risk measure $\rho(X)$. For deriving this expression, we make the following assumptions:

A7.2. The constraints g_{k_1} is an affine function of the parameter ξ for $k_1 = 1, \ldots, K_1$, and f_{k_2} is a convex function of the parameter ξ for $k_2 = 1, \ldots, K_2$.

A7.3. There exists a strictly feasible point for the problem in (7.25).

A7.4. The family of functions $\{L_\theta(\xi, \eta, \lambda, \tilde{\lambda})\}_{\xi,\eta,\lambda,\tilde{\lambda}}$ is equi-differentiable in θ[1].

We now discuss these assumptions. The motivation for A7.2 comes from the result that a risk measure $\rho(D)$ is coherent if and only if the underlying risk envelope $\mathfrak{U}$ is convex and weakly compact. The conditions on g_{k_1} and f_{k_2} can be inferred from the convexity requirement on $\mathfrak{U}$. A7.3 ensures strong duality holds for (7.25), which in turn implies $\max_\xi \min_{\eta,\lambda,\tilde{\lambda}} L_\theta(\cdot,\cdot,\cdot,\cdot) = \min_{\eta,\lambda,\tilde{\lambda}} \max_\xi L_\theta(\cdot,\cdot,\cdot,\cdot)$. This interchange facilitates the application of the envelope theorem. For the application of the latter theorem, we also require the equi-differentiability condition imposed in A7.4.

Proposition 7.4. Assume A7.2–A7.4. Let $\lambda_\theta^* = (\lambda_\theta^*(1), \ldots, \lambda_\theta^*(K_1))$, $\tilde{\lambda}_\theta^* = (\tilde{\lambda}_\theta^*(1), \ldots, \tilde{\lambda}_\theta^*(K_2))$, and let $(\xi_\theta^*, \eta_\theta^*, \lambda_\theta^*, \tilde{\lambda}_\theta^*)$ denote a saddle point of (7.29). Then,

$$\nabla\rho(D) = \mathbb{E}_{\xi_\theta^*}\left[\nabla \log \mathcal{P}_\theta(\omega)(D - \eta_\theta^*)\right] - \sum_{k_1=1}^{K_1} \lambda_\theta^*(k_1)\nabla g_{k_1}(\xi_\theta^*, \mathcal{P}_\theta) - \sum_{k_2=1}^{K_2} \tilde{\lambda}_\theta^*(k_2)\nabla f_{k_2}(\xi_\theta^*, \mathcal{P}_\theta).$$

Proof. A7.3 implies Slater's condition holds for the problem (7.25). Furthermore, by A7.2, we have that the Lagrangian $L_\theta(\xi, \eta, \lambda, \tilde{\lambda})$ is convex in ξ, and concave in the Lagrange multipliers η, λ, and $\tilde{\lambda}$. Thus, strong duality holds, implying

$$\max_{\xi\geq 0} \min_{\eta,\lambda,\tilde{\lambda}\geq 0} L_\theta(\xi, \eta, \lambda, \tilde{\lambda}) = \min_{\eta,\lambda,\tilde{\lambda}\geq 0} \max_{\xi\geq 0} L_\theta(\xi, \eta, \lambda, \tilde{\lambda}).$$

[1] A family $\{f(x,\cdot)\}_{x\in\mathcal{X}}$ is equi-differentiable at $p \in [0,1]$ if $\frac{(f(x,t')-f(x,t))}{t'-t}$ converges uniformly as $t' \to t$.

Since the family $\{L_\theta(\xi, \eta, \lambda, \tilde{\lambda})\}_{\xi,\eta,\lambda,\tilde{\lambda}}$ is equi-differentiable by assumption A7.4, and $L_\theta(\xi, \eta, \lambda, \tilde{\lambda})$ is a smooth function of θ, we obtain the following by an application of the envelope theorem:

$$\begin{aligned}
&\nabla \max_{\xi \geq 0} \min_{\eta, \lambda_1, \ldots, \lambda, \tilde{\lambda} \geq 0} L_\theta(\xi, \eta, \lambda, \tilde{\lambda}) \\
&= \nabla L_\theta(\xi_\theta^*, \eta_\theta^*, \lambda_\theta^*, \tilde{\lambda}_\theta^*) \\
&= \mathbb{E}_{\xi_\theta^*} \left[\nabla \log \mathcal{P}_\theta(\omega)(D - \eta_\theta^*) \right] - \sum_{k_1=1}^{K_1} \lambda_\theta^*(k_1) \nabla g_{k_1}(\xi_\theta^*, \mathcal{P}_\theta) \\
&\qquad - \sum_{k_2=1}^{K_2} \tilde{\lambda}_\theta^*(k_2) \nabla f_{k_2}(\xi_\theta^*, \mathcal{P}_\theta),
\end{aligned}$$

where the final equality uses the following fact:

$$\nabla \mathbb{E}\left[\xi D\right] - \eta \nabla \left(\mathbb{E}\left[\xi\right] - 1 \right) = \mathbb{E}_\xi \left[\nabla \log \mathcal{P}_\theta(\omega)(D - \eta) \right].$$

The equality above can be inferred using the likelihood ratio method. To elaborate for the discrete case, notice that

$$\begin{aligned}
\nabla \mathbb{E}\left[\xi D\right] &= \sum_\omega \xi(\omega) D(\omega) \nabla P_\theta(\omega) \\
&= \sum_\omega \xi(\omega) D(\omega) \nabla \log P_\theta(\omega) P_\theta(\omega) \\
&= \mathbb{E}_\xi \left[\nabla \log \mathcal{P}_\theta(\omega) D \right].
\end{aligned}$$

A similar argument works for the other term involving the η factor above. □

We now specialize the expression derived above for the policy gradient of a coherent risk measure to the case of CVaR. Recall that the risk envelope for CVaR is given by $\mathfrak{U} = \left\{ \xi \mid \xi(\omega) \in \left[0, \frac{1}{1-\beta}\right], \mathbb{E}[\xi] = 1 \right\}$. Thus, λ_θ^* and $\tilde{\lambda}_\theta^*$ are both zero vectors, leading to

$$\nabla \mathrm{CVaR}_\alpha(D) = \mathbb{E}_{\xi_\theta^*} \left[\nabla \log \mathcal{P}_\theta(\omega)(D - \eta_\theta^*) \right].$$

It can be shown that $\xi_\theta^* = \frac{1}{1-\beta}$ for $X > \mathrm{VaR}_\alpha(D)$, and $\xi_\theta^* = 0$ otherwise. Thus, we have

$$\nabla \mathrm{CVaR}_\alpha(D) = \mathbb{E}\left[\nabla \log \mathcal{P}_\theta(\omega)(D - \mathrm{VaR}_\alpha(D)) \mid D > \mathrm{VaR}_\alpha(D) \right]. \tag{7.30}$$

To obtain an estimate of $\nabla\rho(D^\theta)$, we require an estimate of $\rho(D^\theta)$, which is obtained by solving the convex optimization problem in (7.28). Let $\xi^*_{n,\theta}, \eta^*_{n,\theta}, \lambda^*_{n,\theta}, \tilde{\lambda}^*_{n,\theta}$ denote the optimal parameter and Lagrange multipliers obtained by solving the estimation problem in (7.28). Using these quantities, the gradient estimate $\widehat{\nabla}\rho(D^\theta)$ is formed as follows:

$$\widehat{\nabla}_n\rho(D) = \sum_{i=1}^{n}\left[\xi^*_{n,\theta}(\omega_i)\mathcal{P}_{n,\theta}(\omega_i)\nabla\log\mathcal{P}_{n,\theta}(\omega_i)(D(\omega_i)-\eta^*_{n,\theta})\right] - \sum_{k_1=1}^{K_1}\lambda^*_{n,\theta}(k_1)\nabla g_{k_1}(\xi^*_{n,\theta},\mathcal{P}_{n,\theta}) - \sum_{k_2=1}^{K_2}\tilde{\lambda}^*_{n,\theta}(k_2)\nabla f_{k_2}(\xi^*_{n,\theta},\mathcal{P}_{n,\theta}). \tag{7.31}$$

The complete algorithm using a coherent risk measure is presented in Algorithm 7.

Algorithm 7: Policy gradient algorithm under a coherent risk measure

Input : initial parameter $\theta_0 \in \Theta$, perturbation constants $\delta_n > 0$, trajectory lengths $\{m_n\}$, step sizes $\{\zeta(n)\}$, projection operator Γ, number of iterations $M \gg 1$.

for $n \leftarrow 0$ **to** $M-1$ **do**
  for $m \leftarrow 0$ **to** $m_n - 1$ **do**
    Use the policy μ^{θ_n} to generate the state x_m, draw action $a_m \sim \mu^{\theta_n}(\cdot|x_m)$;
    Observe next state x_{m+1} and cost $k(x_m, a_m)$;
  end
  `/* Monte Carlo policy evaluation */`
  Use the scheme in (7.28) to obtain the estimate $\hat{\rho}_n$ of the coherent risk measure;
  `/* Gradient estimate using likelihood ratio */`
  Form the gradient estimate $\widehat{\nabla}_{n,\theta}\ \rho(D)$ using (7.31) ;
  `/* Policy update: Gradient descent */`
  $\theta_{n+1} = \Gamma\left[\theta_n - \zeta(n)\left(\widehat{\nabla}_{n,\theta}\ \rho(D)\right)\right]$;
end

Output : Policy θ_M

7.4 Bibliographic remarks

We provide below bibliographic remarks for each case studied in this section.

7.1 The theory of risk-sensitive control using an exponential utility formulation has a long history; see Whittle (1990) for a detailed introduction. However, work on the learning side of things is a more recent development; see, e.g. Borkar (2001), Borkar (2002), Borkar and Meyn (2002), Bhatnagar *et al.* (2006), and Basu *et al.* (2008), and the survey article by Borkar (2010). Our presentation of the policy gradient theorem and the algorithm for optimizing the exponential cost in an average-cost MDP is based on Borkar (2001), where the author provides a sketch of the convergence analysis. For the missing details, the reader is referred to the analysis of a two-timescale policy gradient algorithm in a risk-neutral setting in Konda and Borkar (1999), in particular, Lemmas 5.6–5.7, and Theorem 5.8 there. More recently, Moharrami *et al.* (2022) propose a policy gradient algorithm for solving a truncated version of the exponential cost MDP; see also Proposition 5 in Moharrami *et al.* (2022) for a variational formula for exponential cost, which establishes stability of the latter risk measure against model uncertainties.

7.2 The presentation of the risk-sensitive RL algorithm with CPT as the underlying risk measure is based on Prashanth *et al.* (2016) and Jie *et al.* (2018). In Proposition 7.3, we presented a concentration bound for CPT-value estimation assuming that the underlying distribution has bounded support. Recent work in Bhat and Prashanth (2019) provides more general concentration bound results assuming sub-Gaussian and sub-exponential distributions.

7.3 The case of coherent risk measure is based on Tamar *et al.* (2015a). For an introduction to coherent risk measures and their dual representation, the reader is referred to Shapiro *et al.* (2014, Section 6.3). In particular, the risk envelope for CVaR presented in (7.27) and the justification for the saddle point ξ_θ^* leading to (7.30) are based on Example 6.16 of Shapiro *et al.* (2014).

8

Conclusions and Future Challenges

In this monograph, we considered MDP problems that incorporate a variety of risk measures in discounted-cost, average-cost, and SSP settings. The risk measures considered were variance (both total and per period), CVaR, chance constraints, CPT, and coherent risk measures. Challenges encountered in the various problem settings include the following: (i) lack of structure, leading to failure of classic DP methods (e.g., policy iteration for variance-constrained MDPs); (ii) lack of gradient information for the risk measures; and (iii) challenges in estimation, in the case of CVaR and CPT.

We briefly summarize some possible future research directions for risk-sensitive MDPs:

i) For the discounted MDP setting, the current algorithm employs SPSA only, because a direct gradient estimate cannot be easily obtained for the risk measure. However, a likelihood ratio gradient estimator is available for the cost function, so combining SPSA with direct gradient-based search in a hybrid algorithm might improve the computational efficiency of the algorithm for work along this line (but in the general SA setting, not specific to MDPs). Even more critical is that SPSA requires simulation of

two system trajectories, which might be infeasible in real-time or online settings, so developing a risk-sensitive algorithm that uses only a single trajectory is of practical interest.

ii) For CVaR-constrained MDPs, variance reduction techniques such as importance sampling and conditional Monte Carlo are essential for keeping CVaR estimation variance at reasonable levels, and as far as we are aware, there is no such *provably convergent* CVaR-estimation algorithm in an RL context. Another important need is incorporating function approximation to handle the curse of dimensionality for large state spaces.

iii) A critical challenge is to obtain finite-time bounds for the risk-sensitive RL algorithms, which usually operate on multiple time-scales. To the best of our knowledge, there are no non-asymptotic bounds available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms, even in the risk-neutral RL setting.

iv) Risk measures that have not been explored in an RL context include spectral risk measures (SRMs) and utility-based shortfall risk (UBSR); see the bibliographic remarks at the end of Section 3 for references. The algorithm presented in Section 7.3 handles a general coherent risk measure, so could in principle be specialized to handle an SRM, but a direct algorithm might be more efficient. On the other hand, a risk-sensitive RL algorithm with either UBSR, or more generally, a convex risk measure as the objective/constraint, has not been developed in the literature as far as we are aware.

v) Finally, CPT-value of the return of an MDP does not have a Bellman equation. Here, we proposed treating the CPT-value MDP problem as a black-box stochastic optimization problem, but certainly other approaches, especially ones that exploit special structure such as the Markovian property of an MDP, might be more computationally efficient in certain contexts.

Acknowledgements

The work of Michael Fu was supported in part by the Air Force Office of Scientific Research under Grant FA95502010211.

References

Abdellaoui, M. (2000). "Parameter-free elicitation of utility and probability weighting functions". *Management Science*. 46(11): 1497–1512.

Abounadi, J., D. P. Bertsekas, and V. S. Borkar. (2001). "Learning algorithms for Markov decision processes with average cost". *SIAM Journal on Control and Optimization*. 40(3): 681–698.

Abounadi, J., D. P. Bertsekas, and V. S. Borkar. (2002). "Stochastic approximation for nonexpansive maps: Application to Q-learning algorithms". *SIAM Journal on Control and Optimization*. 41(1): 1–22.

Acerbi, C. (2002). "Spectral measures of risk: A coherent representation of subjective risk aversion". *Journal of Banking & Finance*. 26(7): 1505–1518.

Aleksandrov, V., V. Sysoyev, and V. Shemeneva. (1968). "Stochastic optimization". *Engineering Cybernetics,* 5: 11–16.

Altman, E. (1999). *Constrained Markov Decision Processes*. Vol. 7. CRC Press.

Arapostathis, A., V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. (1993). "Discrete-time controlled Markov processes with average cost criterion: A survey". *SIAM Journal on Control and Optimization*. 31: 282–344.

Arrow, K. J. (1971). *Essays in the Theory of Risk Bearing*. Chicago, IL: Markham.

Artzner, P., F. Delbaen, J. Eber, and D. Heath. (1999). "Coherent measures of risk". *Mathematical Finance*. 9(3): 203–228.

Asmussen, S. and P. Glynn. (2007). *Stochastic Simulation: Algorithms and Analysis*. New York: Springer.

Balbás, A., J. Garrido, and S. Mayoral. (2009). "Properties of distortion risk measures". *Methodology and Computing in Applied Probability*. 11(3): 385–399.

Barakat, A., P. Bianchi, W. Hachem, and S. Schechtman. (2021). "Stochastic optimization with momentum: Convergence, fluctuations, and traps avoidance". *Electronic Journal of Statistics*. 15(2): 3892–3947.

Barberis, N. C. (2013). "Thirty years of prospect theory in economics: A review and assessment". English. *Journal of Economic Perspectives*: 173–196.

Bardou, O., N. Frikha, and G. Pages. (2009). "Computing VaR and CVaR using stochastic approximation and adaptive unconstrained importance sampling". *Monte Carlo Methods and Applications*. 15(3): 173–210.

Bartlett, P. L. and J. Baxter. (2011). "Infinite-horizon policy-gradient estimation". *arXiv preprint arXiv:1106.0665*.

Barto, A., R. S. Sutton, and C. Anderson. (1983). "Neuron-like elements that can solve difficult learning control problems". *IEEE Transaction on Systems, Man and Cybernetics*. 13: 835–846.

Basu, A., T. Bhattacharyya, and V. S. Borkar. (2008). "A learning algorithm for risk-sensitive cost". *Mathematics of Operations Research*. 33(4): 880–898.

Bäuerle, N. and U. Rieder. (2014). "More risk-sensitive Markov decision processes". *Mathematics of Operations Research*. 39(1): 105–120.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.

Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control, Vols. 1 & 2*. 3rd. Athena Scientific.

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control, Vol. II, 4th edition*. Athena Scientific.

Bertsekas, D. P. and J. N. Tsitsiklis. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bhandari, J., D. Russo, and R. Singal. (2018). "A Finite Time Analysis of Temporal Difference Learning With Linear Function Approximation". In: *Conference On Learning Theory*. 1691–1692.

Bhat, S. P. and L. A. Prashanth. (2019). "Concentration of risk measures: A Wasserstein distance approach". In: *Advances in Neural Information Processing Systems*. 11739–11748.

Bhatnagar, S. (2010). "An actor–critic algorithm with function approximation for discounted cost constrained Markov decision processes". *Systems & Control Letters*. 59(12): 760–766.

Bhatnagar, S., V. S. Borkar, and M. Akarapu. (2006). "A Simulation-Based Algorithm for Ergodic Control of Markov Chains Conditioned on Rare Events". *Journal of Machine Learning Research*. 7(70): 1937–1962.

Bhatnagar, S., H. L. Prasad, and L. Prashanth. (2013). *Stochastic Recursive Algorithms for Optimization*. Vol. 434. Springer.

Bhatnagar, S., R. Sutton, M. Ghavamzadeh, and M. Lee. (2009). "Natural actor-critic algorithms". *Automatica*. 45(11): 2471–2482.

Borkar, V. S. (2001). "A sensitivity formula for risk-sensitive cost and the actor–critic algorithm". *Systems & Control Letters*. 44(5): 339–346.

Borkar, V. S. (2002). "Q-learning for risk-sensitive control". *Mathematics of operations research*. 27(2): 294–311.

Borkar, V. S. (2005). "An actor-critic algorithm for constrained Markov decision processes". *Systems & Control Letters*. 54(3): 207–213.

Borkar, V. S. (2008). *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Borkar, V. S. (2010). "Learning algorithms for risk-sensitive control". In: *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*. Vol. 5. No. 9.

Borkar, V. S. and R. Jain. (2010). "Risk-constrained Markov decision processes". In: *IEEE Conference on Decision and Control*. 2664–2669.

Borkar, V. S. and S. P. Meyn. (2000). "The ODE method for convergence of stochastic approximation and reinforcement learning". *SIAM Journal on Control and Optimization*. 38(2): 447–469.

Borkar, V. S. and S. P. Meyn. (2002). "Risk-sensitive optimal control for Markov decision processes with monotone cost". *Mathematics of Operations Research*. 27(1): 192–209.

Borkar, V. S. (1997). "Stochastic approximation with two time scales". *Systems & Control Letters*. 29(5): 291–294.

Bottou, L., F. E. Curtis, and J. Nocedal. (2018). "Optimization methods for large-scale machine learning". *Siam Review*. 60(2): 223–311.

Brandiere, O. and M. Duflo. (1996). "Les algorithmes stochastiques contournent-ils les pieges?" In: *Annales de l'IHP Probabilités et Statistiques*. Vol. 32. No. 3. 395–427.

Brown, D. B. (2007). "Large deviations bounds for estimating conditional value-at-risk". *Operations Research Letters*. 35(6): 722–730.

Browne, S. (1995). "Optimal Investment Policies for a Firm With a Random Risk Process: Exponential Utility and Minimizing the Probability of Ruin". *Mathematics of Operations Research*. 20(4): 937–958. DOI: 10.1287/moor.20.4.937.

Camerer, C. F. (1989). "An experimental test of several generalized utility theories". *Journal of Risk and Uncertainty*. 2(1): 61–104.

Camerer, C. F. (1992). "Recent tests of generalizations of expected utility theory". In: *Utility Theories: Measurements and Applications*. Springer. 207–251.

Camerer, C. F. and T.-H. Ho. (1994). "Violations of the betweenness axiom and nonlinearity in probability". *Journal of Risk and Uncertainty*. 8(2): 167–196.

Cavus, O. and A. Ruszczynski. (2014). "Risk-averse control of undiscounted transient Markov models". *SIAM Journal on Control and Optimization*. 52(6): 3935–3966.

Chang, H. S., M. C. Fu, J. Hu, and S. I. Marcus. (2007). *Simulation-based Algorithms for Markov Decision Processes*. Springer.

Chang, H. S., J. Hu, M. C. Fu, and S. I. Marcus. (2013). *Simulation-based Algorithms for Markov Decision Processes*. Springer.

Charnes, A., W. W. Cooper, and G. H. Symonds. (1958). "Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil". *Management Science*. 4: 253–263.

Choi, S. (2009). "Risk-averse Newsvendor Models". *PhD thesis*. Rutgers.

Chow, Y., M. Ghavamzadeh, L. Janson, and M. Pavone. (2017). "Risk-constrained reinforcement learning with percentile risk criteria". *The Journal of Machine Learning Research*. 18(1): 6070–6120.

Conlisk, J. (1989). "Three variants on the Allais example". *The American Economic Review*: 392–407.

Coraluppi, S. P. and S. I. Marcus. (1999a). "Risk-Sensitive and Minimax Control of Discrete-Time, Finite-State Markov Decision Processes". *Automatica*. 35: 301–309.

Coraluppi, S. P. and S. I. Marcus. (1999b). "Risk-Sensitive, Minimax, and Mixed Risk Neutral/Minimax Control of Markov Decision Processes". In: *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming*. Boston: Birkhauser. 21–40.

Coraluppi, S. P. and S. I. Marcus. (2000). "Mixed risk-neutral/minimax control of discrete-time, finite-state Markov decision processes". *IEEE Transactions on Automatic Control*. 45: 528–532.

Dalal, G., B. Szörényi, and G. Thoppe. (2020). "A tale of two-timescale reinforcement learning with the tightest finite-time bound". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 04. 3701–3708.

Dalal, G., B. Szörényi, G. Thoppe, and S. Mannor. (2018). "Finite sample analyses for TD(0) with function approximation". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Dentcheva, D. and A. Ruszczynski. (2003). "Optimization with stochastic dominance constraints". *SIAM Journal on Optimization*. 14(2): 548–566.

Derman, C. (1970). *Finite State Markovian Decision Processes*. Academic Press.

Fernández-Gaucherand, E. and S. I. Marcus. (1997). "Risk-sensitive optimal control of hidden Markov models: Structural results". *IEEE Transactions on Automatic Control*. 42: 1418–1422.

Filar, J., L. Kallenberg, and H. Lee. (1989). "Variance-penalized Markov decision processes". *Mathematics of Operations Research*. 14(1): 147–161.

Filar, J., D. Krass, and K. Ross. (1995). "Percentile performance criteria for limiting average Markov decision processes". *IEEE Transaction of Automatic Control*. 40(1): 2–10.

Fleming, W. H. and W. M. McEneaney. (1995). "Risk-sensitive control on an infinite time horizon". *SIAM Journal on Control and Optimization*. 33(6): 1881–1915.

Föllmer, H. and A. Schied. (2002). "Convex measures of risk and trading constraints". *Finance and stochastics*. 6(4): 429–447.

Föllmer, H. and A. Schied. (2004). *Stochastic Finance: An Introduction in Discrete Time*. de Gruyter.

Föllmer, H. and A. Schied. (2016). *Stochastic finance*. de Gruyter.

Fu, M. C. (2006). "Gradient Estimation". In: *Handbooks in Operations Research and Management Science: Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Elsevier. Chap. 19. 575–616.

Fu, M. C. (2015). "Stochastic Gradient Estimation". In: *Handbook on Simulation Optimization*. Ed. by M. C. Fu. Springer. Chap. 5.

Ge, R., F. Huang, C. Jin, and Y. Yuan. (2015). "Escaping from saddle points—online stochastic gradient for tensor decomposition". In: *Conference on Learning Theory*. PMLR. 797–842.

Ghadimi, S. and G. Lan. (2013). "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". *SIAM Journal on Optimization*. 23(4): 2341–2368.

Glynn, P. W. (1987). "Likelilood ratio gradient estimation: an overview". In: *Proceedings of the 19th conference on Winter simulation*. ACM. 366–375.

Gonzalez, R. and G. Wu. (1999). "On the shape of the probability weighting function". *Cognitive psychology*. 38(1): 129–166.

Gopalan, A., L. A. Prashanth, M. C. Fu, and S. I. Marcus. (2017). "Weighted bandits or: How bandits learn distorted values that are not expected". In: *AAAI Conference on Artificial Intelligence*. 1941–1947.

Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer.

Gower, R. M., M. Schmidt, F. Bach, and P. Richtárik. (2020). "Variance-reduced methods for machine learning". *Proceedings of the IEEE*. 108(11): 1968–1983.

Harless, D. W. (1992). "Predictions about indifference curves inside the unit triangle: A test of variants of expected utility theory". *Journal of Economic Behavior & Organization*. 18(3): 391–414.

Heidergott, B. and F. Vázquez-Abad. (2000). "Measure-valued differentiation for stochastic processes: the finite horizon case". *Tech. rep.* No. Report 2000-033. EURANDOM.

Hernández-Hernández, D. and S. I. Marcus. (1996). "Risk sensitive control of Markov processes in countable state space". *Systems & Control Letters*. 29(3): 147–155.

Hernández-Hernández, D. and S. I. Marcus. (1999). "Existence of risk sensitive optimal stationary policies for controlled Markov processes". *Applied Mathematics and Optimization*. 40: 273–285.

Howard, R. A. and J. E. Matheson. (1972). "Risk-sensitive Markov decision processes". *Management Science*. 18: 356–369.

Iyengar, G. N. (2005). "Robust dynamic programming". *Mathematics of Operations Research*. 30(2): 257–280. URL: http:https://doi.org/10.1287/moor.1040.0129.

Jiang, D. R. and W. B. Powell. (2017). "Risk-averse approximate dynamic programming with quantile-based risk measures". *Mathematics of Operations Research*. 43(2): 554–579.

Jie, C., L. A. Prashanth, M. C. Fu, S. I. Marcus, and C. Szepesvári. (2018). "Stochastic optimization in a cumulative prospect theory framework". *IEEE Transactions on Automatic Control*. 63(9): 2867–2882.

Jin, C., R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. (2017). "How to escape saddle points efficiently". In: *International Conference on Machine Learning*. PMLR. 1724–1732.

Kahneman, D. and A. Tversky. (1979). "Prospect theory: An analysis of decision under risk". *Econometrica*: 263–291.

Kiefer, J. and J. Wolfowitz. (1952). "Stochastic estimation of the maximum of a regression function". *Annals of Mathematical Statistics*. 23: 462–266.

Kolla, R. K., L. A. Prashanth, S. P. Bhat, and K. P. Jagannathan. (2019). "Concentration bounds for empirical conditional value-at-risk: The unbounded case". *Operations Research Letters*. 47(1): 16–20.

Konda, V. R. and V. S. Borkar. (1999). "Actor-Critic–Type Learning Algorithms for Markov Decision Processes". *SIAM Journal on control and Optimization*. 38(1): 94–123.

Konda, V. R. and J. N. Tsitsiklis. (2004). "Convergence rate of linear two-time-scale stochastic approximation". *The Annals of Applied Probability*. 14(2): 796–819.

Kushner, H. and D. Clark. (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag. 191–196.

Lin, K. (2013). "Stochastic Systems with Cumulative Prospect Theory". *PhD thesis*. University of Maryland, College Park.

Lin, K., C. Jie, and S. I. Marcus. (2018). "Probabilistically distorted risk-sensitive infinite-horizon dynamic programming". *Automatica*.

Lin, K. and S. I. Marcus. (2013a). "Cumulative weighting optimization: The discrete case". In: *Proceedings of the 2013 Winter Simulation Conference*. Washington, D.C.: Institute of Electrical and Electronic Engineers, Inc.

Lin, K. and S. I. Marcus. (2013b). "Dynamic programming with non-convex risk-sensitive measures". In: *Proceedings of the 2013 American Control Conference*. Washington, D.C.

Mannor, S. and J. N. Tsitsiklis. (2013). "Algorithmic aspects of mean–variance optimization in Markov decision processes". *European Journal of Operational Research*. 231(3): 645–653.

Marbach, P. and J. N. Tsitsiklis. (2001). "Simulation-based optimization of Markov reward processes". *IEEE Transactions on Automatic Control*. 46(2): 191–209.

Marcus, S. I., E. Fernández-Gaucherand, S. C. D. Hernández-Hernández, and P. Fard. (1997). "Risk sensitive Markov decision processes". In: *Systems and Control in the Twenty-First Century*. Ed. by C. I. Byrnes. Boston: Birkhauser. 263–279.

Markowitz, H. (1952). "Portfolio selection". *The Journal of Finance*. 7(1): 77–91.

Mas-Colell, A., M. Whinston, and J. Green. (1995). *Microeconomic theory*. Oxford University Press.

Mihatsch, O. and R. Neuneier. (2002). "Risk-sensitive reinforcement learning". *Machine Learning*. 49(2): 267–290.

Miller, L. B. and H. Wagner. (1965). "Chance-constrained programming with joint constraints". *Operations Research*. 13: 199–213.

Moharrami, M., Y. Murthy, A. Roy, and R. Srikant. (2022). "A Policy Gradient Algorithm for the Risk-Sensitive Exponential Cost MDP". arXiv: 2202.04157 [eess.SY].

Mokkadem, A. and M. Pelletier. (2006). "Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms". *The Annals of Applied Probability*. 16(3): 1671–1702.

Nemirovski, A. and A. Shapiro. (2007). "Convex Approximations of Chance Constrained Programs". *SIAM Journal on Optimization*. 17(4): 969–996.

Papini, M., D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli. (2018). "Stochastic variance-reduced policy gradient". In: *International Conference on Machine Learning*. Vol. 80. *Proceedings of Machine Learning Research*. PMLR. 4026–4035.

Pemantle, R. (1990). "Nonconvergence to unstable points in urn models and stochastic approximations". *The Annals of Probability*. 18(2): 698–712.

Pflug, G. C. (1989). "Sampling derivatives of probabilities". *Computing*. 42: 315–328.

Pflug, G. C. (1996). *Optimization of Stochastic Models*. Kluwer Academic.

Polyak, B. T. and A. B. Juditsky. (1992). "Acceleration of stochastic approximation by averaging". *SIAM Journal on Control and Optimization*. 30(4): 838–855.

Prashanth, L. A. (2014). "Policy gradients for CVaR-constrained MDPs". In: *Algorithmic Learning Theory (ALT)*. 155–169.

Prashanth, L. A., S. Bhatnagar, M. C. Fu, and S. I. Marcus. (2018). "Adaptive system optimization using random directions stochastic approximation". *IEEE Transactions on Automatic Control*. 62(5): 2223–2238.

Prashanth, L. A. and M. Ghavamzadeh. (2013). "Actor-critic algorithms for risk-sensitive MDPs". In: *Advances in Neural Information Processing Systems (NIPS)*. 252–260.

Prashanth, L. A. and M. Ghavamzadeh. (2016). "Variance-constrained actor-critic algorithms for discounted and average reward MDPs". *Machine Learning*. 105(3): 367–417.

Prashanth, L. A., K. Jagannathan, and R. K. Kolla. (2020). "Concentration bounds for CVaR estimation: The cases of light-tailed and heavy-tailed distributions". In: *International Conference on Machine Learning*. Vol. 119. PMLR. 5577–5586.

Prashanth, L. A., C. Jie, M. C. Fu, S. I. Marcus, and C. Szepesvári. (2016). "Cumulative prospect theory meets reinforcement learning: prediction and control". In: *International Conference on Machine Learning*. 1406–1415.

Prashanth, L. A., N. Korda, and R. Munos. (2021). "Concentration bounds for temporal difference learning with linear function approximation: The case of batch data and uniform sampling". *Mach. Learn.* 110(3): 559–618.

Prekopa, A. (2003). "Probabilistic programming". In: *Stochastic Programming*. Ed. by A. Ruszczyński and Shapiro. Elsevier, Amsterdam.

Prékopa, A. (1970). "On probabilistic constrained programming". In: *Proceedings of the Princeton Symposium on Mathematical Programming*. Princeton University Press, Princeton, NJ. 113–138.

Prelec, D. (1998). "The probability weighting function". *Econometrica*: 497–527.

Puterman, M. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Reiman, M. and A. Weiss. (1989). "Sensitivity analysis for simulations via likelihood ratios". *Operations Research*. 37: 830–844.

Riedel, F. (2004). "Dynamic coherent risk measures". *Stochastic Processes and Their Applications*. 112: 185–200.

Robbins, H. and S. Monro. (1951). "A stochastic approximation method". *The Annals of Mathematical Statistics*: 400–407.

Rockafellar, R. T. and S. Uryasev. (2000). "Optimization of conditional value-at-risk". *Journal of Risk*. 2: 21–42.

Ross, S. M. (1983). *Introduction to Stochastic Dynamic Programming*. Academic Press.

Rubinstein, R. Y. (1989). "Sensitivity analysis of computer simulation models via the score efficient". *Operations Research*. 37: 72–81.

Ruppert, D. (1991). "Stochastic approximation". *Handbook of Sequential Analysis*: 503–529.

Ruszczyński, A. (2010). "Risk-averse dynamic programming for Markov decision processes". *Mathematical Programming*. 125: 235–261.

Ruszczyński, A. and A. Shapiro. (2006). "Conditional risk mappings". *Mathematics of Operations Research*. 31(3): 544–561.

Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*. Vol. 162. John Wiley & Sons.

Shapiro, A., D. Dentcheva, and A. Ruszczyński. (2014). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

Shen, Y., W. Stannat, and K. Obermayer. (2013). "Risk-sensitive Markov control processes". *SIAM Journal on Control and Optimization*. 51(5): 3652–3672.

Shen, Z., A. Ribeiro, H. Hassani, H. Qian, and C. Mi. (2019). "Hessian aided policy gradient". In: *International Conference on Machine Learning*. PMLR. 5729–5738.

Sion, M. (1958). "On general minimax theorems". *Pacific J. Math*. 8(1): 171–176.

Sobel, M. (1982). "The variance of discounted Markov decision processes". *Journal of Applied Probability*: 794–802.

Sopher, B. and G. Gigliotti. (1993). "A test of generalized expected utility theory". *Theory and Decision*. 35(1): 75–106.

Spall, J. C. (1992). "Multivariate stochastic approximation using simultaneous perturbation gradient approximation". *IEEE Transactions on Automatic Control*. 37: 332–341.

Srikant, R. and L. Ying. (2019). "Finite-Time Error Bounds For Linear Stochastic Approximation and TD Learning". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Vol. 99. *Proceedings of Machine Learning Research*. PMLR. 2803–2830.

Starmer, C. (2000). "Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk". *Journal of Economic Literature*: 332–382.

Sutton, R. S. (1984). "Temporal Credit Assignment in Reinforcement Learning". *PhD thesis*. University of Massachusetts Amherst.

Sutton, R. S. (1988). "Learning to predict by the methods of temporal differences". *Machine Learning*. 3(1): 9–44.

Sutton, R. S., D. A. McAllester, S. P. Singh, and Y. Mansour. (1999). "Policy gradient methods for reinforcement learning with function approximation." In: *NIPS*. Vol. 99. 1057–1063.

Sutton, R. S. and A. G. Barto. (2018). *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press.

Szepesvári, C. (2011). "Reinforcement learning algorithms for MDPs". *Wiley Encyclopedia of Operations Research and Management Science*.

Tallec, Y. L. (2007). "Robust, Risk-sensitive, and Data-driven Control of Markov Decision Processes". *PhD thesis*. MIT.

Tamar, A., D. D. Castro, and S. Mannor. (2012). "Policy gradients with variance related risk criteria". In: *Proceedings of the Twenty-Ninth International Conference on Machine Learning*. 387–396.

Tamar, A., Y. Chow, M. Ghavamzadeh, and S. Mannor. (2015a). "Policy gradient for coherent risk measures". In: *Advances in Neural Information Processing Systems*. Vol. 28. 1468–1476.

Tamar, A., Y. Chow, M. Ghavamzadeh, and S. Mannor. (2015b). "Policy gradient for coherent risk measures". *CoRR*. abs/1502.03919. arXiv: 1502.03919.

Tamar, A., D. Di Castro, and S. Mannor. (2013). "Temporal difference methods for the variance of the reward to go". In: *International Conference on Machine Learning*. 495–503.

Tamar, A., Y. Glassner, and S. Mannor. (2014a). "Optimizing the CVaR via sampling". *arXiv preprint arXiv:1404.3862*.

Tamar, A., Y. Glassner, and S. Mannor. (2014b). "Policy gradients beyond expectations: Conditional Value-at-Risk". *arXiv preprint arXiv:1404.3862*.

Thomas, P. and E. Learned-Miller. (2019). "Concentration Inequalities for Conditional Value at Risk". In: *International Conference on Machine Learning*. 6225–6233.

Tsitsiklis, J. N. and B. Van Roy. (1997). "An analysis of temporal-difference learning with function approximation". *IEEE Transactions on Automatic Control*. 42(5): 674–690.

Tsitsiklis, J. N. and B. Van Roy. (1999). "Average cost temporal-difference learning". *Automatica*. 35(11): 1799–1808.

Tversky, A. and D. Kahneman. (1992). "Advances in prospect theory: Cumulative representation of uncertainty". *Journal of Risk and Uncertainty*. 5(4): 297–323.

Wang, Y. and F. Gao. (2010). "Deviation inequalities for an estimator of the conditional value-at-risk". *Operations Research Letters*. 38(3): 236–239.

Whittle, P. (1990). *Risk-sensitive Optimal Control. Wiley-Interscience series in systems and optimization*. Wiley. ISBN: 9780471926221.

Zhang, K., A. Koppel, H. Zhu, and T. Basar. (2020). "Global convergence of policy gradient methods to (almost) locally optimal policies". *SIAM Journal on Control and Optimization*. 58(6): 3586–3612.