Risk-Sensitive and Efficient Reinforcement Learning Algorithms

Aviv Tamar

Risk-Sensitive and Efficient Reinforcement Learning Algorithms

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Aviv Tamar

Submitted to the Senate of
the Technion — Israel Institute of Technology
Tammuz 5775 Haifa June 2015

The research thesis was done under the supervision of Prof. Shie Mannor in the Department of Electrical Engineering.

I would like to thank my supervisor, Professor Shie Mannor for his careful guidance throughout my studies. Shie's profound knowledge, clear intuition, and uncompromising scientific approach have been a great inspiration for me, and helped shape my idea of how to do research. Moreover, Shie has been extraordinarily supportive, and spent much-appreciable amounts of time and resources in pushing my career forward. Overall, I could not have hoped for a better mentor, and I am deeply grateful for that.

I am also indebted to my co-authors: Dotan Di Castro, Prof. Huan Xu, Yonatan Glassner, Yinlam Chow, and Prof. Mohammad Ghavamzadeh. It has been a pleasure to work together, and I have learned so much.

I also thank my thesis committee: Prof. Ron Meir, Prof. Nahum Shimkin, and Prof. Yishay Mansour, for their comments and insights.

The stress and hard work during the last three years has been counterbalanced by many refreshing recesses and enjoyable discussions with many friends and colleagues in the Machine Learning group. Thank you!

Finally, I thank my parents, for their unconditional support and encouragement, and my wife Elinor. Her encouragement in the toughest moments, constant support, and very smart advice have helped me a lot.

The generous financial support of the Technion, the Clore foundation, the Gutwirth fellowship, the Ollendorf fellowship, and the Jury award is gratefully acknowledged. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement n. 306638, and from the European Unions Seventh Framework Programme (FP7/2007-2013) under PASCAL2 (PUMP PRIMING) grant agreement no. 216886.

Contents

List of Figures

Abstract

Reinforcement learning (RL) is a computational framework for sequential decision-making, which combines control methods with machine-learning techniques, and is the state-of-the-art for solving large-scale decision problems. Many real-world decision problems involve uncertainty, either due to noise in the system dynamics or due to parameter uncertainty about the model. In the standard RL formulation, such uncertainty is handled by considering as an objective the *expected* return. In this work, we pursue a more versatile approach towards uncertainty, and extend the RL methodology to take into account the *risk* of the return. For uncertainty due to noisy dynamics, we consider several risk-measures of the return, including mean-variance formulations, conditional value-at-risk, and coherent risk measures. We extend the policy-gradient RL approach to such risk-sensitive objectives. For parameter uncertainty, we extend the robust Markov decision process formulation to the RL setting, using function approximation. Thereby, our approach allows to handle modeling errors in large, or continuous decision problems.

Abbreviations and Notations

RL	—	Reinforcement learning
MDP	—	Markov decision process
TD	—	Temporal differences
VaR	—	Value-at-risk
$CVaR$	—	Conditional value-at-risk

$\mathcal{M}$	—	Markov decision process (MDP)
$\mathcal{X}$	—	State space
x	—	State
$\mathcal{U}$	—	Action space
u	—	Action
R	—	Reward
$R_{\max}$	—	Maximal reward
P	—	Transition probability
x_0	—	Initial state
γ	—	Discount factor
π	—	Policy
J^π	—	Expected discounted return for policy π
J^*	—	Maximal expected discounted return
π^*	—	Optimal policy
V^π	—	Value function for policy π
V^*	—	Optimal value function
$\hat{V}^\pi$	—	Approximate optimal value for policy π
$\hat{V}^*$	—	Approximate optimal value function
θ	—	Policy parameters
π_θ	—	Parameterized policy

$J(\theta)$	—	Expected return for policy π_θ
Q	—	State-action value-function
B	—	Return random variable
T	—	Time horizon
Ω	—	The set of all possible MDP trajectories of length T
$\mathcal{F}$	—	A σ-algebra over Ω
P_θ	—	A probability measure over $\mathcal{F}$ parameterized by θ
$\mathcal{Z}$	—	Space of random variables $Z : \Omega \mapsto (-\infty, \infty)$
ρ	—	Static risk-measure
ρ_{dynamic}	—	Dynamic risk-measure
$J_{\text{mean-var}}$	—	Variance-penalized objective
$J_{\text{var-const}}$	—	Variance-constrained objective
J_{Sharpe}	—	Sharpe ratio objective
V^{π}_{Var}	—	Value function for the return variance
q_α	—	The α-quantile of the return
J_{CVaR_α}	—	The α-CVaR of the return
$\mathcal{P}$	—	Uncertainty set
J_{robust}	—	Robust MDP objective
V^{π}_{robust}	—	Robust MDP value function for policy π
$\hat{V}^{\pi}_{\text{robust}}$	—	Robust MDP approximate value function for policy π
$\phi(x)$	—	State-dependent features

Chapter 1

Introduction

1.1 Reinforcement Learning

Reinforcement learning (RL) [100, 12] is a computational framework for sequential decision-making. In the RL setting, an agent can perform actions in a dynamic environment, where each agent's action induces a (possibly stochastic) change in the system state. In addition, a scalar reward signal is associated with every system-state and agent-action, and quantifies the *immediate* value of performing the action at that state. The agent's goal is to perform optimally in the *long term*, by selecting actions that maximize the *sum of all rewards* along some predefined time horizon.

When the environment dynamics are stochastic, the sum of rewards is stochastic as well, and often the goal in such cases is chosen to be to maximize the *expected* sum of rewards. While this goal leads to particularly simple algorithms and derivations, it is a very naive approach for dealing with uncertainty, and ignores such elements as the *variability* of rewards, or the possibility of rare, but potentially disastrous outcomes.

The view taken in this work is that for some applications, the expected total reward is not a suitable objective, and instead, additional statistical properties of the total reward should be considered. In financial literature, such a view is well-established, and termed *risk-management*. Accordingly, in this work we consider *risk-sensitive RL*.

In this section, we review the mathematical foundations of standard RL, and its fundamental algorithms. Our contribution to risk-sensitive RL will be discussed in the subsequent section.

1.1.1 Markov Decision Processes

The mathematical model underlying RL is the Markov decision process (MDP) [84]. An MDP is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{U}, R, P, x_0, \gamma)$, where $\mathcal{X}$ and $\mathcal{U}$ denote finite state and action spaces; $R(x, u) \in [-R_{\max}, R_{\max}]$ is a state-action dependent reward; $P(\cdot|x, u)$ is the Markov state-transition probability distribution; $\gamma \in [0, 1)$ is a discounting factor, and x_0 is the initial state. Let x_t denote the state of the environment at time $t \in 0, 1, 2 \ldots.$ The agent then selects an action u_t, and obtains a reward $R(x_t, u_t)$. Subsequently, the environment transitions to a new state $x_{t+1} \sim P(\cdot|x_t, u_t)$.

A stationary-Markov policy $\pi(u|x)$ is a mapping from states to a probability over actions. The performance of a particular policy π is measured according to its expected discounted return J^π, defined as

$$J^\pi \doteq \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \right], \tag{1.1}$$

where $\mathbb{E}^\pi [\cdot]$ denotes an expectation given that actions are chosen according to policy π. Let J^* the maximal performance:

$$J^* \doteq \max_\pi J^\pi$$

The agent's goal in an MDP problem is to find an optimal policy π^* that satisfies $J^{\pi^*} = J^*$, or equivalently

$$\pi^* \in \arg\max_\pi J^\pi. \tag{1.2}$$

In general, the approaches for solving problem (1.2) can be categorized as either value-based methods, or policy-search methods, as we now review.

1.1.2 Value Based Methods

Value-based methods for solving problem (1.2) are based on a dynamic-programming principle [11], by the observation that the optimal policy may be derived by first computing a structure known as a *value-function.* For some policy π, the value-function $V^\pi : \mathcal{X} \to \mathbb{R}$ is defined as

$$V^\pi(x) \doteq \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \middle| x_0 = x \right]. \tag{1.3}$$

Similarly, the optimal value-function V^* is given by

$$V^*(x) \doteq \max_{\pi} \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \middle| x_0 = x \right].$$

Value-based methods for MDPs are driven by the well-known *Bellman equation* [11], which shows that the optimal value-function obeys a dynamic-programming decomposition, and that the optimal policy may be derived from it.

Theorem 1.1 (Bellman's principle of optimality) *The value-function and optimal-value-function satisfy the following equations:*

$$\begin{aligned} V^\pi(x) &= \sum_{u \in \mathcal{U}} \left(\pi(u|x) R(x,u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,u) \pi(u|x) V^\pi(x') \right), \\ V^*(x) &= \max_{u \in \mathcal{U}} \left(R(x,u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,u) V^*(x') \right). \end{aligned} \tag{1.4}$$

Furthermore, the greedy policy w.r.t. V^*, given by

$$\pi^*(x) = \arg\max_{u \in \mathcal{U}} \left(R(x,u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,u) V^*(x') \right) \tag{1.5}$$

is an optimal policy.

Theorem 1.1 shows that by calculating the optimal value-function V^*, a solution to problem (1.2) may be derived. Thus, value-based methods for MDPs focus on solving (1.2) by calculating V^*. Within the value-based methods, the most popular approaches for calculating V^* are *value-iteration*, *policy iteration*, and *linear programming*.

In value-iteration [10], an initial value function V_0 is arbitrarily selected, and iteratively updated by the following rule (cf. Eq. 1.4):

$$V_{k+1}(x) := \max_{u \in \mathcal{U}} \left(R(x,u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,u) V_k(x') \right), \quad \forall x \in \mathcal{X}.$$

It is well-known [11] that value iteration converges to the optimal value-function, i.e., $\lim_{k\to\infty} V_k = V^*$, for any initial V_0, at a geometric rate equal to γ.

In policy-iteration [42], an initial policy π_0 is arbitrary selected, and iteratively updated according to the following two-step procedure. Let π_k denote the policy at iteration k of the algorithm. First, the value-function V^{π_k} is calculated, using either value-iteration, or a direct solution of (1.4), by noting that the equation for V^{π_k} is linear. Next, a new policy π_{k+1} is selected by choosing greedy actions w.r.t. V^{π_k} (cf. Eq. 1.5):

$$\pi_{k+1}(x) = \arg\max_{u \in \mathcal{U}} \left(R(x,u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,u) V^{\pi_k}(x') \right).$$

When $\pi_{k+1} = \pi_k$, policy-iteration has converged to an optimal policy π^*, and the algorithm may be halted. It is well-known [11] that this occurs in a finite number of iterations.

In the linear-programming approach to MDPs [58], the optimal value-function is shown to be the solution of the following linear program

$$\begin{aligned} \min_V \quad & \sum_{x \in \mathcal{X}} \nu(x) V(x) \\ \text{s.t.} \quad & V(x) \geq R(x,u) + \gamma \sum_{x'} P(x'|x,u) V(x') \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, \end{aligned} \tag{1.6}$$

where ν is a probability distribution over $\mathcal{X}$, and $\nu(x) > 0$ for all $x \in \mathcal{X}$. Linear programming problems have been studied extensively, and can be solved using methods such as interior-point algorithms [18].

The curse of dimensionality A major caveat of the classic MDP solutions described above, known as the 'curse of dimensionality' [10], is their ineffectiveness in dealing with large, or continuous state-spaces. This is since at each iteration of value-iteration or policy-iteration, a sweep of all the states is performed. Similarly, for the linear-programming approach, the number of constraints in the linear program is $|\mathcal{X}| \times |\mathcal{U}|$. Thus, when $\mathcal{X}$ is very large, these methods become inefficient. This is a serious practical limitation, since many real-world problems involve large state-spaces, often composed of several state-variables.

Scaling-up MDPs using RL The inefficiency of classical MDP methods in handling large problems led to an investigation of *approximate* MDP solutions, that may scale-up to large, or even continuous MDPs. In the operations-research literature, such methods are referred to as approximate dynamic programming [82], while in the machine-learning community they are known as reinforcement-learning [100]. The idea behind value-based RL methods is to use machine-learning methods, such as sampling and regression, to learn an *approximate representation* of the optimal value-function, $\hat{V}^*$, and use this approximate value-function for deriving a policy, according to (1.5). By controlling the number of parameters in the value-function representation, the complexity of the approximate solution algorithms may be contained, and therefore large problems may be handled. We now survey some of the more popular value-based RL methods.

Most value-based RL methods consider a *parametric* representation of the approximate value function

$$\hat{V}^*(x) = f(x; w), \tag{1.7}$$

where $w \in \mathbb{R}^k$ denotes some k-dimensional parameter vector. For example, f may represent a linear combination of state-dependent features $f(x; w) = \phi(x)^\top w$, where $\phi(x) \in \mathbb{R}^k$ denotes the features; such representations have been studied extensively [111, 12, 11, 100], and were used in several successful RL applications [98, 6, 82, 92]. Another popular representation is a neural-network, in which w denotes the network weights; such were used in several impressive RL applications [109, 66]. In this dissertation we focus on parametric representations. However, non-parametric representations such as Gaussian processes [30], and kernel-based methods [74], were also investigated in the literature.

Once a representation has been chosen, the problem becomes how to choose the parameters w, such that $\hat{V}^*$ is similar to V^* in some sense. RL methods use a *machine-learning* approach for this task. Similar to classical MDP solutions, RL approaches can also be classified as either value-iteration, policy-iteration, or linear programming. For the first two approaches, the fundamental idea for learning w is the method of *temporal differences* (TD) [99]. The main idea driving TD learning is the following. From the Bellman equation (1.4), we know that the true (i.e., not approximated) value function for a fixed policy π must satisfy $V^\pi(x) = \mathbb{E}^\pi\left[R(x,u) + \gamma V^\pi(x')\right]$. Therefore, if we have some approximation of the value function $\hat{V}^\pi$, we may improve it by tuning w to minimize the distance between $\hat{V}^\pi(x)$ and $\mathbb{E}^\pi\left[R(x,u) + \gamma \hat{V}^\pi(x')\right]$ in some sense, where typically, the expectation is replaced with a sample average, obtained by *sampling* a trajectory from the MDP.

Different approaches for performing this minimization lead to different flavors of the TD algorithm. Online methods [99, 111, 4, 101] employ a stochastic gradient-descent style approach, while batch methods [17, 72] resemble the least-squares method. More recently, regularized methods have also been investigated [50, 31]. The TD approach for a fixed policy may then be combined with a policy-improvement step, for a policy-iteration style procedure [100, 54, 35, 31]. Alternatively, one may instead start from the Bellman equation for V^* (1.4), and develop an approximate value-iteration method [113, 39, 86].

In the *approximate linear programming* approach [24, 28], the value function V in (1.6) is replaced with its function approximation form (1.7), and sampling of the constraints is employed.

1.1.3 Policy Search Methods

Policy-search approaches for solving problem (1.2) represent the *policy* in a parametric functional-form, and use sampling-based methods to search for optimal policy parameters. Formally, we let $\theta \in \mathbb{R}^k$ denote the policy parameters, i.e.,

$$\pi(u|x) \doteq \pi_\theta(u|x). \tag{1.8}$$

Popular policy representations include the softmax $\pi_\theta(u|x) \propto e^{\phi(x,u)^\top \theta}$, where $\phi(x,u)$ denotes state-action dependent features [102], soft-threshold policies [62], and dynamic movement primitives [49].

For easing the introduction, we replace the infinite horizon objective (1.1) with an un-discounted finite horizon of length T. Let $J(\theta)$ denote the performance of a policy parameterized by θ, i.e.,

$$J(\theta) \doteq \mathbb{E}^{\pi_\theta}\left[\sum_{t=0}^{T} R(x_t, u_t)\right]. \tag{1.9}$$

The goal in policy-search is thus to solve the following optimization problem

$$\max_{\theta} J(\theta). \tag{1.10}$$

Several methods for solving problem (1.10) have been investigated in the literature. Arguably, the most popular policy-search method is the *policy-gradient* approach [114, 9, 62], in which the gradient $\nabla J(\theta)$ is estimated using sampling, and used in a stochastic gradient-descent algorithm for finding a local optimum of (1.10). The key-idea underlying policy gradient methods is the likelihood-ratio formula [38]

$$\nabla J(\theta) = \mathbb{E}^{\pi_\theta}\left[\left(\sum_{t=0}^{T} \nabla \log \pi_\theta(u_t|x_t)\right)\left(\sum_{t=0}^{T} R(x_t, u_t)\right)\right]. \tag{1.11}$$

Eq. 1.11 may be used to design a sampling-based estimator for $\nabla J(\theta)$ by sampling trajectories from the MDP under policy π_θ, and replacing the expectation in (1.11) with a sample average; such an approach is pursued in [114, 9, 62]. Several modifications of this idea have also been investigated, such as replacing the gradient with the natural-gradient [47]; a covariant gradient method [3]; variance-reduction techniques [40]; a Newton-step approach [34]; a model-based variant using Gaussian-processes [25]; and different exploration strategies [93].

Actor-critic methods A particulary useful class of policy-search methods combines the policy-gradient approach with the dynamic programming ideas of value-based RL. The underlying concept driving actor-critic methods is the *policy-gradient theorem* [102]:

$$\nabla J(\theta) = \mathbb{E}^{\pi_\theta}\left[\sum_{t=0}^{T} \nabla \log \pi_\theta(u_t|x_t) Q_t^{\pi_\theta}(x_t, u_t)\right], \tag{1.12}$$

where $Q(x_t, u_t)$ is the state-action value-function, defined as $Q_t^{\pi_\theta}(x, u) = \mathbb{E}^{\pi_\theta}\left[\sum_{s=t}^{T} R(x_s, u_s)\middle| x_t = x, u_t = u\right]$. Actor-critic methods [102, 51] exploit Eq. 1.12 to employ a sampling-based algorithm for solving problem (1.10), composed of two interleaved procedures:

Critic: For a given policy π_θ, calculate an approximate value-function $Q_t^{\pi_\theta}$ using the methods described in Section 1.1.2,

Actor: Using the critic's value-function and Eq. 1.12, use sampling to estimate $\nabla J(\theta)$ and update θ using a gradient-descent update.

The main advantage of actor-critic algorithms is their reduced variance in gradient-estimation, compared to the standard policy-gradient [77]. Furthermore, various improvements of the standard actor-critic algorithm have been proposed recently. Among these modifications are an extension to function approximation [102]; a natural-gradient variant [78, 13]; and an extension to skill-learning [55].

Policy-gradient and actor-critic methods have been thoroughly studied; they enjoy theoretical guarantees such as convergence to a local optimum of $J(\theta)$ [13, 51], and have been applied to various problems in robotics [77], finance [68], and computer-games [47]. Nevertheless, several other policy-search approaches for solving problem (1.10) have been proposed. Among them, we mention evolutionary-algorithms [69], the cross-entropy method [104], expectation-maximization [49], path-integral control [110], and the recent relative-entropy policy-search algorithm [76], which combines policy-search with the approximate linear-programming approach to RL.

1.2 Uncertainty and Risk in Reinforcement Learning

An absolute majority of the RL literature, as summarized in the previous section, has focused on the *expected* return criterion, as in (1.1), (1.9), and similar variants thereof. Thus, the only mitigation of uncertainty in standard RL is through the expectation, with respect to the return distribution induced by the MDP parameters.

The view taken in this work, is that in some scenarios, such a simplistic approach to managing uncertainty is not satisfactory. Before describing our proposed resolution for this problem, we first discuss the different *sources* *of uncertainty* in sequential decision-making.

There are two sources that contribute to the reward uncertainty in MDPs: *internal-uncertainty* and *model-uncertainty.* Internal-uncertainty reflects the uncertainty of the return due to the stochastic transitions (and possibly stochastic rewards), for a single and *known* MDP. Model uncertainty, on the other hand, reflects the uncertainty about the MDP *parameters* – the transition and reward distributions. In general, inherent-uncertainty becomes important when there is significant stochasticity in the MDP transitions, which may lead to significant *variability* in the return [43]. Model-uncertainty is important when the MDP parameters used during planning the policy are different than the parameters used for testing it (for example, if they are estimated from data), or change in time. It is known that such differences in the MDP parameters may have a significant influence on performance [60].

In general, the two sources of uncertainty are *fundamentally* different, and therefore require different algorithmic approaches.

The natural method for dealing with inherent-uncertainty, motivated by classical studies in the financial literature, is through the notion of *risk* [63, 2]. By risk, we mean different statistical properties of the return, such as its variance [63], value-at-risk (VaR) [29], conditional value-at-risk (CVaR) [87], or exponential-utility [43]. Such measures capture the *variability* of the return, or quantify the effect of rare but potentially disastrous outcomes. While the prime application of risk-management is in the financial domain, there are several other domains in which such a view towards uncertainty is important, and gaining popularity, such as health domains [71], robotics [53], and operations-research [23].

For dealing with model-uncertainty, a popular method is the robust-MDP framework [73, 45], in which the true MDP parameters are assumed to belong to some set, termed the uncertainty set, and a policy is sought such that it maximizes the expected return w.r.t. the *worst-case* parameters within this set. Such a prudent approach guarantees that any possible performance degradation due to the unknown MDP parameters is contained.

The contribution of this work, is to extend the RL methodology to accommodate uncertainty, using the two principles described above, namely risk-management and robustness to model uncertainty. In the following, we describe our efforts.

We remark that another approach to model-uncertainty is the *Bayesian method*, in which the MDP parameters are assumed to be generated from a distribution, and the expectation in the objective (1.1) is w.r.t. both the parameters and the state-transitions. Due to the prohibitively challenging difficulties of the current state-of-the-art in Bayesian RL, we do not consider it in this work. We refer the interested reader to a recent survey on this matter [37].

1.2.1 Inherent Uncertainty

We provide an overview of our contribution in dealing with inherent-uncertainty in MDPs, using the notion of risk-sensitive MDPs.

The starting point in our approach, is that the return in an MDP is a *random-variable*, which we denote by B

$$B = \sum_{t=0}^{T} R(x_t, u_t). \tag{1.13}$$

The standard RL objective (1.9) is to maximize $\mathbb{E}^{\pi}[B]$. In our work, we discuss several alternative objectives, and term the corresponding decision problems *risk-sensitive MDPs*. We begin with the formal definitions of risk-measures relevant to sequential decision-making, which form the basis of our approach.

In a sequential decision-making setting, there are two popular methods of measuring risk, termed static and dynamic.

Static Risk In the static risk case, the total return B is considered as a standard random variable, *without any regard to the temporal nature* of the process generating it. Formally, consider a probability space $(\Omega, \mathcal{F}, P_\theta)$, where Ω is the set of all possible MDP trajectories of length T, $\mathcal{F}$ is a σ-algebra over Ω, and P_θ is a probability measure over $\mathcal{F}$ parameterized by some tunable parameter θ, corresponding to the probability of observing a trajectory when the policy parameters are θ. Denote by $\mathcal{Z}$ the space of random variables $Z : \Omega \mapsto (-\infty, \infty)$ defined over the probability space $(\Omega, \mathcal{F}, P_\theta)$. The total cost B in (1.13) is an example of such a random variable. A *static risk-measure* is a function $\rho : \mathcal{Z} \to \mathbb{R}$ that maps an uncertain outcome Z to the extended real line $\mathbb{R} \cup \{+\infty, -\infty\}$. For the case of the return B, the expectation $\mathbb{E}[B]$ and the variance $\mathrm{Var}[B]$ are two examples of static risk-measures.

Dynamic Risk-Measures, on the other hand, explicitly capture the multi-period nature of the decision-making process in the definition of the risk. Consider a probability space $(\Omega, \mathcal{F}, P_\theta)$, a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \mathcal{F}_2 \cdots \subset \mathcal{F}_T \subset \mathcal{F}$, and an adapted sequence of real-valued random variables Z_t, $t \in \{0, \ldots, T\}$. We assume that $\mathcal{F}_0 = \{\Omega, \emptyset\}$, i.e., Z_0 is deterministic. For each $t \in \{0, \ldots, T\}$, we denote by $\mathcal{Z}_t$ the space of random variables defined over the probability space $(\Omega, \mathcal{F}_t, P_\theta)$, and also let $\mathcal{Z}_{t,T} := \mathcal{Z}_t \times \cdots \times \mathcal{Z}_T$ be a sequence of these spaces. The sequence of random variables Z_t may be interpreted as the immediate rewards observed along a trajectory from an MDP with policy parameter θ, i.e., $Z_{0,T} \doteq \big(Z_0 = R(x_0, u_0), \ldots, Z_T = R(x_T, u_T)\big) \in \mathcal{Z}_{0,T}$.

To evaluate risk consistently in a dynamic setting, we no longer construct a single risk metric, but rather a *sequence* of risk metrics $\rho_{t,T} : \mathcal{Z}_{t,T} \to \mathcal{Z}_t$, mapping a future stream of random costs into a risk assessment at time t, based on the information (history) available at that time. That is, for any possible trajectory of length t in $\mathcal{F}_t$, $\rho_{t,T}$ maps all the possible futures in $\mathcal{Z}_{t,T}$ to a scalar that represents the risk of the trajectory.

The primary motivation for this definition of dynamic risk, is to explicitly consider the issue of *time-consistency* [94, 44], usually defined as follows [89]: if a certain outcome is considered less risky in all states of the world at stage $t+1$, then it should also be considered less risky at stage t. It is well-known, that static risk-measures do not necessarily obey time consistency [44], leading to somewhat paradoxical results (see Chapter 6 and [44, 89] for an in-depth discussion). Remarkably, Theorem 1 in [89] shows that the following "multi-stage composition of risk":

$$\rho_{t,T}(Z) = Z_t + \rho_t\Big(Z_{t+1} + \rho_{t+1}\big(Z_{t+2} + \ldots + \rho_{T-1}(Z_T)\big)\Big),$$

with each ρ_t being a static risk-measure, is a *necessary and sufficient condition* for time consistency. In this work we consider a special case of time-consistent dynamic risk-measures termed *Markov dynamic risk-measures*, which are suitable for the MDP setting, and are defined as follows

$$\rho_{\text{dynamic}}(B) = R(x_0, u_0) + \rho\left(R(x_1, u_1) + \ldots + \rho\left(R(x_T, u_T)\right)\right), \qquad (1.14)$$

where ρ is a static risk-measure, and the evaluation ρ is Markov, in the sense that it is not allowed to depend on the whole past, and $x_0, u_0, \ldots, x_T, u_T$ is a trajectory drawn from the MDP.

Risk-Sensitive MDPs, as stated earlier, are MDP problems in which the standard objective is replaced with a risk-measure of the total return. In this work we consider both the static risk, $\rho(B)$, for several risk-measures ρ, and also the Markov dynamic risk $\rho_{\text{dynamic}}(B)$. Such a comprehensive treatment of various risk-measures allows the decision maker great flexibility in designing her risk preferences. Our main contribution is an algorithmic approach for actually solving the resulting decision problems, which is described in Chapters 2, 3, 5, and 6. In the following, we highlight our results for the various cases.

Mean-Variance Risk

Perhaps the most popular risk measure in the financial literature is used in the Markowitz mean-variance model [63]. In this model, the objective is a function of both the expected return and its variance, for example the variance penalized objective[1]:

$$J_{\text{mean-var}} = \max_{\pi} \left\{ \mathbb{E}^{\pi}\left[B\right] - \beta \text{Var}^{\pi}\left[B\right] \right\},$$

where $\beta \in \mathbb{R}$ controls the penalty on return variability, and $\text{Var}^{\pi}\left[\cdot\right]$ denotes variance when the policy is π. Popular variations of this objective include the variance-constrained objective:

$$J_{\text{var-const}} = \max_{\pi} \mathbb{E}^{\pi}\left[B\right] \quad \text{s.t.} \quad \text{Var}^{\pi}\left[B\right] \leq \beta, \tag{1.15}$$

and the Sharpe-ratio objective [96]

$$J_{\text{Sharpe}} = \max_{\pi} \frac{\mathbb{E}^{\pi}\left[B\right]}{\sqrt{\text{Var}^{\pi}\left[B\right]}}. \tag{1.16}$$

While such mean-variance objective are popular in financial applications, using them in the context of MDPs has been known to be challenging. Already

[1] These results are for the static risk setting.

in the early work of [97], it has been noted that the variance in MDPs does not obey a monotonicity property, which is exploited by standard dynamic-programming methods such as policy iteration. More recently, it has been shown [61] that the objective in (1.15) is in general NP-hard to solve. Although these negative results are somewhat discouraging, in this work we show that mean-variance objectives may be successfully optimized, by pursuing a *policy-gradient based* approach.

In Chapter 2, we show that when using a parameterized policy π_θ (cf. Section 1.1.3), the policy-gradient based approach may be extended to mean-variance objectives. Specifically, we show that similarly to Eq. 1.11, the gradient $\nabla J_{\text{mean-var}}(\theta)$ of the mean-variance objective $J_{\text{mean-var}}(\theta) = \mathbb{E}^{\pi_\theta}[B] - \beta \text{Var}^{\pi_\theta}[B]$ may be written in a likelihood-ratio style formula, and this formula may be used to devise a sampling estimator for the gradient. By following a stochastic gradient-descent approach, we use the sampled gradient to optimize over θ. Furthermore, using a penalty-method technique, we extend our approach to objectives such as (1.15) and (1.16).

We guarantee that our approach reaches a *locally-optimal* point of the objective. Since finding a globally-optimal point is NP-hard, in some sense, this is the best we can hope for.

In Chapter 3, we further extend our study of the mean-variance setting by investigating a 'value-function equivalent' for this case. Specifically, we consider the *variance of the reward-to-go*, under some policy π (cf. the value function V^π in Eq. 1.3), defined as

$$V^{\pi}_{Var}(x) \doteq \text{Var}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \middle| x_0 = x \right].$$

The variance value-function $V^{\pi}_{Var}(x)$ is important for assessing the risk of a particular state, and may also be used for developing an actor-critic approach for the mean-variance objective. For large state-spaces, we suggest to represent V^{π}_{Var} using function approximation (cf. Eq. 1.7), and propose a temporal-difference based approach for learning the approximation parameters.

CVaR Risk

The conditional value-at-risk (CVaR; [87]) is a prominent risk measure, that has found extensive use in finance among other fields.

Mathematically, let q_α denote the α-quantile of the return, which, assuming a continuous return distribution, is defined as follows

$$q_\alpha = \{x : P(B \leq x) = \alpha\}.$$

The α-CVaR of the return under policy[2] π_θ is defined as

$$J_{\mathrm{CVaR}_\alpha}(\theta) = \mathbb{E}^{\pi_\theta}\left[B \middle| B \leq q_\alpha\right].$$

Put simply, the CVaR is the expected α worst-cases of the return. Thus, by appropriately tuning α, the CVaR may be tuned to be sensitive to rare, but very low returns, which makes it particularly attractive as a risk measure.

The CVaR has been studied extensively [87, 95], and is known to have favorable mathematical such as coherence [2]. It has also been used in many practical applications, in finance and other domains [112].

In Chapter 5, we show that the policy-gradient based approach may be extended to CVaR based objectives as well. Specifically, we provide a likelihood-ratio formula for the gradient $\nabla J_{\mathrm{CVaR}_\alpha}(\theta)$, and present an algorithm that exploits this formula for estimating the gradient using sampling. We provide an analysis and convergence proof of the resulting algorithm, and demonstrate its applicability on the challenging problem of learning to play Tetris in a risk-averse style.

Coherent Risk Measures

In Chapter 6, we extend the policy gradient method to *the whole class* of coherent risk measures, which is widely accepted in finance and operations research, among other fields, and encompasses popular risk-measures such as the CVaR, and mean-semideviation. These results generalize the CVaR method of Chapter 5, and allow significant flexibility in choosing the risk objective in RL.

For the definition of coherent risk, it is standard to express the objective in terms of *cost* instead of reward, and to assume that the goal is to minimize

[2] These results are for the static risk setting.

the risk associated with a policy[3]. A static risk-measure is called *coherent*, if it satisfies the following four conditions [2] for all $Z, W \in \mathcal{Z}$:

A1 Convexity: $\forall \lambda \in [0,1],\ \rho\big(\lambda Z + (1-\lambda)W\big) \leq \lambda\rho(Z) + (1-\lambda)\rho(W)$;

A2 Monotonicity: if $Z(\omega) \leq W(\omega)$ for all $\omega \in \Omega$, then $\rho(Z) \leq \rho(W)$;

A3 Translation invariance: $\forall a \in \mathbb{R},\ \rho(Z+a) = \rho(Z) + a$;

A4 Positive homogeneity: if $\lambda \geq 0$, then $\rho(\lambda Z) = \lambda\rho(Z)$.

Intuitively, these condition ensure the "rationality" of single-period risk assessments: A1 ensures that diversifying an investment will reduce its risk; A2 guarantees that an asset with a higher cost for every possible scenario is indeed riskier; A3, also known as 'cash invariance', means that the deterministic part of an investment portfolio does not contribute to its risk; the intuition behind A4 is that doubling a position in an asset doubles its risk.

In Chapter 6, we show that the policy-gradient approach may be extended to any static coherent risk-measure, by combining the standard sampling approach with convex programming. In addition, we consider the Markov dynamic risk $\rho_{\text{dynamic}}(B)$, where the static risk in (1.14) is coherent. We extend the policy-gradient theorem (1.12) for this case, and present an actor-critic approach, which combines the policy-gradient with risk-sensitive value-function learning.

1.2.2 Model Uncertainty

Our contribution to model-uncertainty in RL concerns the problem of planning with robust-MDPs.

A robust-MDP [73, 45] is an extension of the MDP that accounts for model-uncertainty. Formally, a robust-MDP is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{U}, R, \mathcal{P}, x_0, \gamma)$, where $\mathcal{X}$, $\mathcal{U}$, R, x_0, and γ are the same as in the MDP definition of Section 1.1.1, and $\mathcal{P}$ denotes a set of plausible MDP transitions, termed the *uncertainty-set*, where each element of $\mathcal{P}$ is composed of transition probabilities $P(\cdot|x,u)$ for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$.

[3] Any reward-based formulation may be easily converted to a cost-based formulation by multiplying the reward by -1, and replacing the maximization in (1.10) with a minimization.

The objective in planning with robust MDPs is to maximize the expected return under the worst-case parameter realization in $\mathcal{P}$, that is

$$J_{\text{robust}} = \sup_{\pi} \inf_{P \in \mathcal{P}} \mathbb{E}^{\pi,P} \left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \right], \tag{1.17}$$

where $\mathbb{E}^{\pi,P}[\cdot]$ denotes an expectation w.r.t. trajectories drawn from an MDP with transitions $P(\cdot|x,u)$, under policy π.

The purpose of the objective in (1.17), is to guarantee that losses due to modeling errors are contained, so long as the *true* model is in $\mathcal{P}$. Since modeling error have been shown to significantly degrade performance in standard MDPs [60], such guarantee is important.

Traditional approaches for solving robust MDPs have exploited a dynamic-programming decomposition of (1.17), that holds when the uncertainty-set obeys certain structural properties [73, 45]. Both a value-iteration algorithm [73] and a policy-iteration method [45] were suggested. However, similar to their standard MDP counterparts (cf. Section 1.1.2), these methods suffer from the curse-of-dimensionality, and do not scale-up to problems with large or continuous state-spaces.

In Chapter 4, we present an RL approach to robust-MDPs, that scales-up to large, or even continuous problems. Our approach is based on using function-approximation to represent the *robust equivalent of the value-function* of a policy π

$$V^{\pi}_{\text{robust}}(x) \doteq \inf_{P \in \mathcal{P}} \mathbb{E}^{\pi,P} \left[\sum_{t=0}^{\infty} \gamma^t R(x_t, u_t) \middle| x_0 = x \right]. \tag{1.18}$$

Specifically, we consider a linear-approximation of the value function (cf. Section 1.1.2)

$$\hat{V}^{\pi}_{\text{robust}}(x) = \phi(x)^{\top} w, \tag{1.19}$$

and suggest a temporal-difference based approach for learning the approximation parameters w. We then use this approximation method within an approximate policy-iteration algorithm, to iteratively improve the policy. To our knowledge, this is the first approach to scale-up robust MDP beyond small-scale problems.

1.2.3 Related Work

The topic of risk-aware decision making has been of interest for quite a long time, and several frameworks for incorporating risk into decision making have been suggested. In the context of MDPs and addressing inherent uncertainty, Howard and Matheson [43] proposed to use an exponential utility function, where the factor of the exponent controls the risk sensitivity. Liu and Koenig [56] later extended this approach to combinations of exponential and linear utilities, and Moldovan and Abbeel [67] considered a related measure that is defined using Chernoff bounds. Another approach considers the *percentile* performance criterion [32], in which the average reward has to exceed some value with a given probability, and the related chance-constraint formulation was studied in [115]. Variance-based risk criteria, have also been considered, but were shown to be computationally demanding [97, 61].

In-line with the mathematical-finance literature, more recent studies have considered static-CVaR in MDPs [16, 8]. However, neither of these works proposed practical algorithms. The dynamic-coherent risk was studied in [79], for systems with linear-dynamics, and [14], for small-scale problems based on dynamic-programming. In addition, Osogami [75] has shown that dynamic coherent risk is equivalent to a certain robust MDP.

The novelty of our work with respect to these studies, is in pursuing an RL approach, with approximation in the policy and value-function, that scales-up to large or continuous problems, for systems with general dynamics.

Indeed, much less work has been done on risk sensitive criteria within the RL framework. Basu et al. [7] considered exponential utility functions, and Geibel and Wysotzki [36] considered models in which some states are "error states," representing a bad or even catastrophic outcome. Mihatsch and Neuneier [64] suggested a different approach, in which a risk-measure was enforced on the temporal-difference error, instead of the return. In a different context, Morimura et al. [70] consider the expected return, but use the a CVaR based risk-sensitive policy for guiding the exploration while learning.

Addressing parameter uncertainty has been done within the Bayesian framework (where a prior is assumed on the unknown parameters, see [81, 37]), or within the robust MDP framework [73, 45, 116, 59], where a worst-case approach is taken over the parameters inside an uncertainty set. As stated earlier, our approach scales-up robust MDPs to potentially much larger problems.

Chapter 2

Policy Gradients with Variance Related Risk Criteria

This chapter was published as:

A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 935–942. Omnipress, 2012.

Abbreviations

RL	—	Reinforcement learning
MDP	—	Markov decision process
SR	—	Sharpe ratio
SSP	—	Stochastic shortest-path problem
COP	—	Constrained optimization problem
ODE	—	Ordinary differential equation

Notations

J	—	Cumulative expected reward
V	—	Variance of cumulative reward
X	—	State space
U	—	Action space
x	—	State
x^*	—	Recurrent state
u	—	Action
r	—	Reward
P_u	—	Transition probability for action u
θ	—	Policy parameters
μ_θ	—	Policy
P_θ	—	Transition probability induced by μ_θ
π_θ	—	Stationary distribution induced by μ_θ
η_θ	—	Average reward criterion
B	—	Accumulated reward random variable
J	—	Value function
V	—	Trajectory variance function
S	—	Sharpe ratio objective
P	—	Transition matrix
P'	—	Matrix P with x^* column zeroed out
ρ	—	Auxiliary variable for calculating V
$\circ$	—	Element-wise (Hadamard) product
g	—	Penalty function
α	—	Step-size (fast rate)
β	—	Step-size (slow rate)

Policy Gradients with Variance Related Risk Criteria

Aviv Tamar AVIVT@TX.TECHNION.AC.IL
Dotan Di Castro DOT@TX.TECHNION.AC.IL
Shie Mannor SHIE@EE.TECHNION.AC.IL
Department of Electrical Engineering, The Technion - Israel Institute of Technology, Haifa, Israel 32000

Abstract

Managing risk in dynamic decision problems is of cardinal importance in many fields such as finance and process control. The most common approach to defining risk is through various variance related criteria such as the Sharpe Ratio or the standard deviation adjusted reward. It is known that optimizing many of the variance related risk criteria is NP-hard. In this paper we devise a framework for local policy gradient style algorithms for reinforcement learning for variance related criteria. Our starting point is a new formula for the variance of the cost-to-go in episodic tasks. Using this formula we develop policy gradient algorithms for criteria that involve both the expected cost and the variance of the cost. We prove the convergence of these algorithms to local minima and demonstrate their applicability in a portfolio planning problem.

1. Introduction

In both Reinforcement Learning (RL; Bertsekas & Tsitsiklis, 1996) and planning in Markov Decision Processes (MDPs; Puterman, 1994), the typical objective is to maximize the cumulative (possibly discounted) expected reward, denoted by J. When the model's parameters are known, several well-established and efficient optimization algorithms are known. When the model parameters are not known, learning is needed and there are several algorithmic frameworks that solve the learning problem efficiently, at least when the model is finite. In many applications, however, the decision maker is also interested in minimizing some form of *risk* of the policy. By risk, we mean reward criteria that take into account not only the expected reward, but also some additional statistics of the total reward such as its variance, its Value at Risk, etc. (Luenberger, 1998). Risk can be measured with respect to two types of uncertainties. The first type, termed *parametric uncertainty* is related to the imperfect knowledge of the problem parameters. The second type, termed *inherent uncertainty* is related to the stochastic nature of the system (random reward and transition function). Both types of uncertainties can be important, depending on the application at hand.

Risk aware decision making is important both in planning and in learning. Two prominent examples are in finance and process control. In financial decision making, a popular performance criterion is the *Sharpe Ratio* (SR; Sharpe, 1966) – the ratio between the expected profit and its standard deviation. This measure is so popular that it is one of the reported metrics each mutual fund reports annually. When deciding how to allocate a portfolio both types of uncertainties are important: the decision maker does not know the model parameters or the actual realization of the market behavior. In process control as well, both uncertainties are essential and a robust optimization framework (Nilim & El Ghaoui, 2005) is often adopted to overcome imperfect knowledge of the parameters and uncertainty in the transitions. In this paper we focus on inherent uncertainty and use learning to mitigate parametric uncertainty.

The topic of risk-aware decision making has been of interest for quite a long time, and several frameworks for incorporating risk into decision making have been suggested. In the context of MDPs and addressing inherent uncertainty, Howard & Matheson (1972) proposed to use an exponential utility function, where the factor of the exponent controls the risk sensitivity. Another approach considers the *percentile* performance criterion (Filar et al., 1995), in which the average reward has to exceed some value with a given probability. Addressing parameter uncertainty has been done within

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

the Bayesian framework (where a prior is assumed on the unknown parameters, see Poupart et al., 2006) or within the robust MDP framework (where a worst-case approach is taken over the parameters inside an uncertainty set). Much less work has been done on risk sensitive criteria within the RL framework, with a notable exception of Borkar & Meyn (2002) who considered exponential utility functions and of Geibel & Wysotzki (2005) who considered models where some states are "error states," representing a bad or even catastrophic outcome.

In this work we consider an RL setup and focus on risk measures that involve the *variance of the cumulative reward*, denoted by V. Typical performance criteria that fall under this definition include

(a) Maximize J s.t. $V \leq c$

(b) Minimize V s.t. $J \geq c$

(c) Maximize the Sharpe Ratio: $J/\sqrt{V}$

(d) Maximize $J - c\sqrt{V}$

The rationale behind our choice of risk measure is that these performance criteria, such as the SR mentioned above, are being used in practice. Moreover, it seems that human decision makers understand how to use variance well, and that exponential utility functions require determining the exponent coefficient which is non-intuitive.

Variance-based risk criteria, however, are computationally demanding. It has long been recognized (Sobel, 1982) that optimization problems such as (a) are not amenable to standard dynamic programming techniques. Furthermore, Mannor & Tsitsiklis have shown that even when the MDP's parameters are known, many of these problems are computationally intractable, and some are not even approximable. This is not surprising given that other risk related criteria such as percentile optimization are also known to be hard except in special cases.

Despite these somewhat discouraging results, in this work we show that this important problem may be tackled successfully, by considering policy gradient type algorithms that optimize the problem *locally*. We present a framework for dealing with performance criteria that include the variance of the cumulative reward. Our approach is based on a new fundamental result for the variance of episodic tasks. Previous work by Sobel, 1982 presented similar equations for the infinite horizon discounted case, however, the importance of our result is that the episodic setup allows us to derive policy gradient type *algorithms*. We present both model-based and model-free algorithms for solving problems (a) and (c), and prove that they converge. Extension of our algorithms to other performance criteria such as (b) and (d) listed above is immediate. The effectiveness of our approach is further demonstrated numerically in a risk sensitive portfolio management problem.

2. Framework and Background

In this section we present the framework considered in this work and explain the difficulty in mean-variance optimization.

2.1. Definitions and Framework

We consider an agent interacting with an unknown environment that is modeled by an MDP in discrete time with a finite state set $X \triangleq \{1, \ldots, n\}$ and finite action set $U \triangleq \{1, \ldots, m\}$. Each selected action $u \in U$ at a state $x \in X$ determines a stochastic transition to the next state $y \in X$ with a probability $P_u(y|x)$.

For each state x the agent receives a corresponding reward $r(x)$ that is bounded and depends only on the current state[1]. The agent maintains a parameterized *policy function* that is in general a probabilistic function, denoted by $\mu_\theta(u|x)$, mapping a state $x \in X$ into a probability distribution over the controls U. The parameter $\theta \in \mathbb{R}^{K_\theta}$ is a tunable parameter, and we assume that $\mu_\theta(u|x)$ is a differentiable function w.r.t. θ. Note that for different values of θ, different probability distributions over U are associated for each $x \in X$. We denote by $x_0, u_0, r_0, x_1, u_1, r_1, \ldots$ a state-action-reward trajectory where the subindex specifies time. For notational easiness, we define x_i^k, u_i^k, and r_i^k to be $x_i \ldots, x_k$, $u_i \ldots, u_k$, and $r_i \ldots, r_k$, respectively, and R_i^k to be the cumulative reward along the trajectory $R_i^k = \sum_{j=i}^k r_j$.

Under each policy induced by $\mu_\theta(u|x)$, the environment and the agent induce together a Markovian transition function, denoted by $P_\theta(y|x)$, satisfying $P_\theta(y|x) = \sum_u \mu_\theta(u|x) P_u(y|x)$. The following assumption will be valid throughout the rest of the paper.

Assumption 2.1. Under all policies, the induced Markov chain P_θ is ergodic, i.e., aperiodic, recurrent, and irreducible.

Under assumption 2.1 the Markovian transition function $P_\theta(y|x)$ induces a stationary distribution over the state space X, denoted by π_θ. We denote by $\mathbb{E}_\theta[\cdot]$ and

[1] Generalizing the results presented here to the case where the reward depends on the state and the action rewards is straightforward.

$\mathrm{Var}_\theta[\cdot]$ to be the expectation and variance operators w.r.t. the measure $P_\theta(y|x)$.

There are several performance criteria investigated in the RL literature that differ mainly on their time horizon and the treatment of future rewards (Bertsekas & Tsitsiklis, 1996). One popular criterion is the *average reward* defined by $\eta_\theta = \sum_x \pi_\theta(x) r(x)$. Under this criterion, the agent's goal is to find the parameter θ that maximizes η_θ. One appealing property of this criterion is the possibility of obtaining estimates of $\nabla \eta_\theta$ from simulated trajectories efficiently, which leads to a class of stochastic gradient type algorithms known as *policy gradient* algorithms. In this work, we also follow the policy gradient approach, but focus on the mean-variance tradeoff. While one can consider the tradeoff between η_θ and $\mathrm{Var}_\pi[r(x)]$, defined as the variance w.r.t the measure π_θ, these expressions are not sensitive to the trajectory but only to the induced stationary distribution, and represent the per-round variability.

Consequently, we focus on the finite horizon case, also known as the episodic case, that is important in many applications. Assume (without lost of generality) that x^* is some recurrent state for all policies and let $\tau \triangleq \min\{k > 0 | x_k = x^*\}$ denote the first passage time to x^*.

Let the random variable B denote the accumulated reward along the trajectory terminating at the recurrent state x^*

$$B \triangleq \sum_{k=0}^{\tau-1} r(x_k). \qquad (1)$$

Clearly, it is desirable to choose a policy for which B is large in some sense [2]. In this work, we are interested in the mean-variance tradeoff in B.

We define the *value function* as

$$J(x) \triangleq \mathbb{E}_\theta \left[B | x_0 = x\right], \quad , x = 1, \ldots, n, \qquad (2)$$

and the *trajectory variance function* as

$$V(x) \triangleq \mathrm{Var}_\theta \left[B | x_0 = x\right], \quad , x = 1, \ldots, n.$$

Note that the dependence of $J(x)$ and $V(x)$ on θ is suppressed in notation.

The questions explored in this work are the following stochastic optimization problems:

(a) The constrained trajectory-variance problem:

$$\max_\theta J(x^*) \quad \text{s.t.} \quad V(x^*) \leq b, \qquad (3)$$

where b is some positive value.

(b) The maximal SR problem:

$$\max_\theta S(x^*) \triangleq \frac{J(x^*)}{\sqrt{V(x^*)}}. \qquad (4)$$

In order for these problems to be well defined, we make the following assumption:

Assumption 2.2. Under all policies $J(x^*)$ and $V(x^*)$ are bounded.

For the SR problem we also require the following:

Assumption 2.3. We have $V(x^*) > \epsilon$ for some $\epsilon > 0$.

In the next subsection we discuss the challenges involved in solving problems (3) and (4), which motivate our gradient based approach.

2.2. The Challenges of Trajectory-Variance Problems

As was already recognized by Sobel (1982), optimizing the mean-variance tradeoff in MDPs cannot be solved using traditional dynamic programming methods such as policy iteration. Mannor & Tsitsiklis showed that for the case of a finite horizon T, in general, solving problem (3) is hard and is equivalent to solving the subset-sum problem. Since our case can be seen as a generalization of a finite horizon problem, (3) is a hard problem as well. One reason for the hardness of the problem is that, as suggested by Mannor & Tsitsiklis, the underlying optimization problem is not necessarily convex. In the following, we give an example where the set of all $(J(x^*), V(x^*))$ pairs spanned by all possible policies is not convex.

Consider the following symmetric deterministic MDP with 8 states $X = \{x^*, x_{1a}, x_{1b}, x_{2a}, x_{2b}, x_{2c}, x_{2d}, t\}$, and two actions $U = \{u_1, u_2\}$. The reward is equal to 1 or -1 when action u_1 or u_2 are chosen, respectively. The MDP is sketched in Figure 1, left pane. We consider a set of random policies parameterized by $\theta_1 \in [0, 1]$ and $\theta_2 \in [0, 1]$, such that $\mu(u_1|x^*) = \theta_1$ and $\mu(u_1|x_{1a}) = \mu(u_1|x_{1b}) = \theta_2$.

Now, we can achieve $J(x^*) \in \{-2, 0, 2\}$ with zero variance if we choose $(\theta_1, \theta_2) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$, i.e., only with the deterministic policies. Any $-2 < J(x^*) < 2$, $J(x^*) \neq 0$, can be achieved but only with a random policy, i.e., with some variance. Thus, the region is not convex. The achievable $(J(x^*), V(x^*))$ pairs are depicted in the right pane of Figure 1.

[2] Note that finite horizon MDPs can be formulated as a special case of (1).

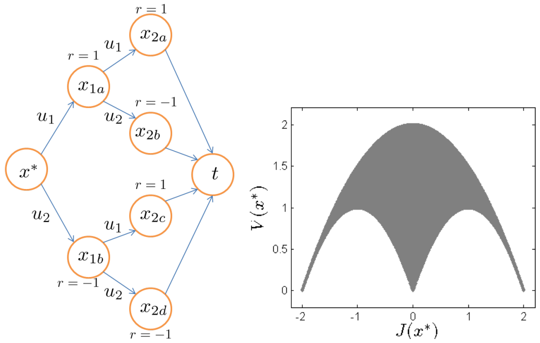


Figure 1. (left) A diagram of the MDP considered in Section 2.2. *(right)* A phase plane describing the non-convex nature of $J(x^*) - V(x^*)$ optimization.

3. Formulae for the Trajectory Variance and its Gradient

In this section we present formulae for the mean and variance of the cumulated reward between visits to the recurrent state. The key point in our approach is the following observation. By definition (1), a transition to x^* always terminates the accumulation in B and does not change its value. Therefore, the following Bellman like equation can be written for the value function

$$J(x) = r(x) + \sum_{y \neq x^*} P_\theta(y|x) J(y) \quad x = 1, \ldots, n. \quad (5)$$

Similar equations can be written for the trajectory variance, and in the following lemma we show that these equations are solvable, yielding expressions for J and V.

Proposition 3.1. *Let P be a stochastic matrix corresponding to a policy satisfying Assumption 2.1, where its (i,j)-th entry is the transition from state i to state j. Define P' to be a matrix equal to P except that the column corresponding to state x^* is zeroed (i.e., $P'(i, x^*) = 0$ for $i = 1, \ldots, n$). Then,*

(a) the matrix $I - P'$ is invertible;

(b) $J = (I - P')^{-1} r$;

(c) $V = (I - P')^{-1} \rho$,

where $\rho \in \mathbb{R}^n$ and

$$\rho(x) = \sum_y P'(y|x) J(y)^2 - \left(\sum_y P'(y|x) J(y) \right)^2 .$$

Proof. (a) Consider an equivalent Stochastic Shortest Path (SSP) problem where x^* is the termination state. The corresponding transition matrix P_{ssp} is defined by $P_{\text{ssp}}(i,j) = P(i,j)$ for $i \neq x^*$, $P_{\text{ssp}}(x^*, j) = 0$ for $j \neq x^*$, and $P_{\text{ssp}}(x^*, x^*) = 1$. Furthermore, let $P^* \in \mathbb{R}^{n-1 \times n-1}$ denote the matrix P with the x^*'th row and column removed, which is also the transition matrix of the SSP problem without the terminal state. By the irreducibility of P in Assumption 2.1 P_{ssp} is *proper*, and by proposition 2.2.1 in (Bertsekas, 2006) we have that $I - P^*$ is invertible.
Finally, observe that by the definition of P' we have

$$\det(I - P') = \det(I - P^*),$$

thus, $\det(I - P') \neq 0$.

(b) Choose $x \in \{1, \ldots, n\}$. Then,

$$J(x) = r(x) + \sum_{y \neq x^*} P(y|x) J(y),$$

where we excluded the recurrent state from the sum since after reaching the recurrent state there is no further rewards by definition (2). In vectorial form, $J = r + P'J$ where using (a) we conclude that $J = (I - P')^{-1} r$.

(c) Choose $x \in \{1, \ldots, n\}$. Then,

$$\begin{aligned}
V(x) =& \mathbb{E}\left[\left(\sum_{k=0}^{\tau-1} r(x_k) \right)^2 \middle| x_0 = x \right] - J(x)^2 \\
=& r(x)^2 + 2r(x) \sum_{y \neq x^*} P(y|x) \mathbb{E}\left[\sum_{k=1}^{\tau-1} r(x_k) \middle| x_1 = y \right] + \\
& \sum_{y \neq x^*} P(y|x) \mathbb{E}\left[\left(\sum_{k=1}^{\tau-1} r(x_k) \right)^2 \middle| x_1 = y \right] - J(x)^2, \\
=& r(x)^2 + 2r(x) \sum_{y \neq x^*} P(y|x) J(y) + \sum_{y \neq x^*} P(y|x) V(y) \\
& + \sum_{y \neq x^*} P(y|x) J(y)^2 - J(x)^2,
\end{aligned}$$

where in the second equality we took the first term out of the summation, and in the third equality we used the definition of J and V. Next, we show that $r(x)^2 + 2r(x) \sum_{y \neq x^*} P(y|x) J(y) + \sum_{y \neq x^*} P(y|x) J(y)^2 - J(x)^2$ is equal to $\rho(x)$:

$$\begin{aligned}
r(x)^2 + 2r(x) & \sum_{y \neq x^*} P(y|x) J(y) - J(x)^2 + \sum_{y \neq x^*} P(y|x) J(y)^2 \\
&= (r(x) + J(x)) (r(x) - J(x)) \\
&\quad + 2r(x) \sum_{y \neq x^*} P(y|x) J(y) + \sum_{y \neq x^*} P(y|x) J(y)^2 \\
&= (r(x) + J(x)) \left(- \sum_{y \neq x^*} P(y|x) J(y) \right)
\end{aligned}$$

$$+\,2r(x)\sum_{y\neq x^*}P(y|x)J(y)+\sum_{y\neq x^*}P(y|x)J(y)^2$$
$$=(r(x)-J(x))\sum_{y\neq x^*}P(y|x)J(y)+\sum_{y\neq x^*}P(y|x)J(y)^2$$
$$=\sum_{y\neq x^*}P(y|x)J(y)^2-\left(\sum_{y\neq x^*}P(y|x)J(y)\right)^2.$$

□

Proposition 3.1 can be used to derive expressions for the gradients w.r.t. θ of J and V. Let $A\circ B$ denote the element-wise product between vectors A and B. The gradient expressions are presented in the following lemma.

Lemma 3.2. *We have*

$$\nabla J=\left(I-P'\right)^{-1}\nabla P' J, \tag{6}$$

and

$$\nabla V=\left(I-P'\right)^{-1}\left(\nabla\rho+\nabla P' V\right), \tag{7}$$

where

$$\nabla\rho=\nabla P' J^2+2P'\left(J\circ\nabla J\right)-2P'J\circ(\nabla P'J+P'\nabla J). \tag{8}$$

The proof is a straightforward differentiation of the expressions in Lemma 3.1, and is described in Section A of the supplementary material [3].

We remark that similar equations for the infinite horizon discounted return case were presented by Sobel (1982), in which $I-P'$ is replaced with $I-\beta P$, where $\beta<1$ is the discount factor. The analysis in (Sobel, 1982) makes use of the fact that $I-\beta P$ is invertible, therefore an extension of their results to the undiscounted case is not immediate.

4. Gradient Based Algorithms

In this section we derive gradient based algorithms for solving problems (3) and (4). We present both exact algorithms, which may be practical for small problems, and simulation based algorithms for larger problems. Our algorithms deal with the constraint based on the penalty method, which is described in the following subsection.

4.1. Penalty methods

One approach for solving constrained optimization problems (COPs) such as (3) is to transform the COP to an equivalent unconstrained problem, which can be solved using standard unconstrained optimization techniques. These methods, generally known as *penalty methods*, add to the objective a penalty term for infeasibility, thereby making infeasible solutions suboptimal. Formally, given a COP

$$\max f(x),\quad \text{s.t.}\quad c(x)\leq 0, \tag{9}$$

we define an unconstrained problem

$$\max f(x)-\lambda g\left(c(x)\right), \tag{10}$$

where $g(x)$ is the *penalty function*, typically taken as $g(x)=\left(\max\left(0,x\right)\right)^2$, and $\lambda>0$ is the penalty coefficient. As λ increases, the solution of (10) converges to the solution of (9), suggesting an iterative procedure for solving (9): solve (10) for some λ, then increase λ and solve (10) using the previous solution as an initial starting point.

In this work we use the penalty method to solve the COP in (3). An alternative approach, which is deferred to future work, is to use *barrier methods*, in which a different penalty term is added to the objective that forces the iterates to remain within the feasible set (Boyd & Vandenberghe, 2004).

4.2. Exact Gradient Algorithm

When the MDP transitions are known, the expressions for the gradients in Lemma 3.2 can be immediately plugged into a gradient ascent algorithm for the following penalized objective function of problem (3)

$$f_\lambda=J(x^*)-\lambda g\left(V(x^*)-b\right).$$

Let α_k denote a sequence of positive step sizes. Then, a gradient ascent algorithm for maximizing f_λ is

$$\theta_{k+1}=\theta_k+\alpha_k\left(\nabla J(x^*)-\lambda g'\left(V(x^*)-b\right)\nabla V(x^*)\right). \tag{11}$$

Let us make the following assumption on the smoothness of the objective function and on the set of its local optima. [4]

Assumption 4.1. For all $\theta\in\mathbb{R}^{K_\theta}$ and $\lambda>0$, the objective function f_λ has bounded second derivatives. Furthermore, the set of local optima of f_λ is countable.

Then, under Assumption 4.1, and suitable conditions on the step sizes, the gradient ascent algorithm (11) can be shown to converge to a locally optimal point of f_λ.

For the SR optimization problem (4), using the quotient derivative rule for calculating the gradient of S,

[3] http://tx.technion.ac.il/~avivt/icml12supp.pdf

[4] Note that the smoothness of $J(x^*)$ and $V(x^*)$ may be satisfied by choosing a suitable policy function such as the softmax function.

we obtain the following algorithm

$$\theta_{k+1} = \theta_k + \frac{\alpha_k}{\sqrt{V(x^*)}} \left(\nabla J(x^*) - \frac{J(x^*)}{2V(x^*)} \nabla V(x^*) \right), \tag{12}$$

which can be shown to converge under similar conditions to a locally optimal point of (4).
When the state space is large, or when the model is not known, computation of the gradients using equations (6) and (7) is not feasible. In these cases, we can use simulation to obtain unbiased estimates of the gradients, as we describe in the next section, and perform a *stochastic* gradient ascent.

4.3. Simulation based optimization

When a simulator of the MDP dynamics is available, it is possible to obtain unbiased estimates of the gradients ∇J and ∇V from a sample trajectory between visits to the recurrent state. The technique is called the *likelihood ratio* method, and it underlies all policy gradient algorithms (Baxter & Bartlett, 2001; Marbach & Tsitsiklis, 1998). The following lemma gives the necessary gradient estimates for our case.

Lemma 4.2. *We have*

$$\nabla J(x) = \mathbb{E}[R_0^{\tau-1} \nabla \log P\left(x_0^{\tau-1}\right) | x_0 = x],$$

and

$$\nabla V(x) = \mathbb{E}[\left(R_0^{\tau-1}\right)^2 \nabla \log P\left(x_0^{\tau-1}\right) | x_0 = x] - 2J(x)\nabla J(x),$$

where the expectation is over trajectories.

The proof is given in Section B of the supplementary material.

Given an observed trajectory $x_0^{\tau-1}, u_0^{\tau-1}, r_0^{\tau-1}$, and using Lemma 4.2 we devise the estimator $\hat{\nabla} J(x^*) \triangleq R_0^{\tau-1} \nabla \log P\left(x_0^{\tau-1}\right)$ which is an unbiased estimator of $\nabla J(x^*)$. Furthermore, using the Markov property of the state transition and the fact that the only dependance on θ is in the policy μ_θ, the term $\nabla \log P\left(x_0^{\tau-1}\right)$ can be reduced to

$$\nabla \log P\left(x_0^{\tau-1}\right) = \sum_{k=0}^{\tau-1} \nabla \log \mu_\theta\left(u_k | x_k\right),$$

making the computation of $\hat{\nabla} J(x^*)$ from an observed trajectory straightforward. Assume for the moment that we know $J(x^*)$ and $V(x^*)$. Then $\hat{\nabla} V(x^*) \triangleq (R_0^{\tau-1})^2 \nabla \log P\left(x_0^{\tau-1}\right) - 2J(x^*)\hat{\nabla} J(x^*)$ is an unbiased estimate of $\nabla V(x^*)$, and plugging $\hat{\nabla} V$ and $\hat{\nabla} J$ in (11) gives a proper stochastic gradient ascent algorithm. Unfortunately, we cannot calculate $J(x^*)$ exactly without knowing the model, and obtaining an unbiased estimate of $J(x)\nabla J(x)$ from a single trajectory is impossible (for a similar reason that the variance of a random variable cannot be estimated from a single sample of it). We overcome this difficulty by using a two timescale algorithm, where estimates of J and V are calculated on the fast time scale, and θ is updated on a slower time scale.

The algorithm updates the parameters every episode, upon visits to the recurrent state x^*. Let τ^k where $k = 0, 1, 2, \ldots$ denote the times of these visits. To ease notation, we also define $x^k = (x_{\tau_{k-1}}, \ldots, x_{\tau_k - 1})$ and $R^k = \sum_{t=\tau_{k-1}}^{\tau_k - 1} r_t$ to be the trajectories and accumulated rewards observed between visits, and denote $z^k \triangleq \nabla \log P(x^k)$ to be the likelihood ratio derivative. The simulation based algorithm for the constrained optimization problem (3) is

$$\begin{aligned}
\tilde{J}_{k+1} &= \tilde{J}_k + \alpha_k \left(R^k - \tilde{J}_k\right) \\
\tilde{V}_{k+1} &= \tilde{V}_k + \alpha_k \left((R^k)^2 - \tilde{J}_k^2 - \tilde{V}_k\right) \\
\theta_{k+1} &= \theta_k + \beta_k \left(R^k - \lambda g'\left(\tilde{V}_k - b\right)\left((R^k)^2 - 2\tilde{J}_k\right)\right) z^k,
\end{aligned} \tag{13}$$

where α_k and β_k are positive step sizes. Similarly, for optimizing the SR (4), we change the update rule for θ to

$$\theta_{k+1} = \theta_k + \frac{\beta_k}{\sqrt{\tilde{V}_k}} \left(R^k - \frac{\tilde{J}_k (R^k)^2 - 2R^k \tilde{J}_k^2}{2\tilde{V}_k} \right) z^k. \tag{14}$$

In the next theorem we prove that algorithm (13) converges almost surely to a locally optimal point of the corresponding objective function. The proof for Algorithm (14) is essentially the same and thus omitted. For notational clarity, throughout the remainder of this section, the dependence of $J(x^*)$ and $V(x^*)$ on θ is made explicit using a subscript.

Theorem 4.3. *Consider algorithm (13), and let Assumptions 2.1, 2.2, and 4.1 hold. If the step size sequences satisfy* $\sum_k \alpha_k = \sum_k \beta_k = \infty$, $\sum_k \alpha_k^2$, $\sum_k \beta_k^2 < \infty$, *and* $\lim_{k\to\infty} \frac{\beta_k}{\alpha_k} = 0$, *then almost surely*

$$\lim_{k\to\infty} \nabla \left(J_{\theta_k}(x^*) - \lambda g\left(V_{\theta_k}(x^*) - b\right)\right) = 0. \tag{15}$$

Proof. (sketch) The proof relies on representing Equation (13) as a stochastic approximation with two timescales (Borkar, 1997), where $\tilde{J}_k$ and $\tilde{V}_k$ are updated on a fast schedule while θ_k is updated on a slow schedule. Thus, θ_k may be seen as quasi-static w.r.t. $\tilde{J}_k$ and $\tilde{V}_k$, suggesting that $\tilde{J}_k$ and $\tilde{V}_k$ may be associated with the following ordinary differential equations (ODE)

$$\begin{aligned}\dot{J} &= \mathbb{E}_\theta[B|x_0 = x^*] - J,\\ \dot{V} &= \mathbb{E}_\theta[B^2|x_0 = x^*] - J^2 - V.\end{aligned} \tag{16}$$

For each θ, the ODE (16) can be solved analytically to yield $J(t) = J^\infty + c_1 e^{-t}$ and $V(t) = V^\infty - 2J^\infty c_1 t e^{-t} + c_1^2 e^{-2t} + c_2 e^{-t}$, where c_1 and c_2 are constants, and $\{J^\infty, V^\infty\}$ is a globally asymptotically stable fixed point which satisfies

$$J^\infty = J_\theta(x^*), \quad V^\infty = V_\theta(x^*). \tag{17}$$

In turn, due to the timescale difference, $\tilde{J}_k$ and $\tilde{V}_k$ in the iteration for θ_k may be replaced with their stationary limit points J^∞ and V^∞, suggesting the following ODE for θ

$$\dot{\theta} = \nabla\left(J_\theta(x^*) - \lambda g\left(V_\theta(x^*) - b\right)\right). \tag{18}$$

Under Assumption 4.1, the set of stable fixed point of (18) is just the set of locally optimal points of the objective function f_λ. Let $\mathcal{Z}$ denote this set, which by Assumption 4.1 is countable. Then, by Theorem 5 in Leslie & Collins, 2002 (which is extension of Theorem 1.1 in Borkar, 1997), θ_k converges to a point in $\mathcal{Z}$ almost surely. □

5. Experiments

In this section we apply the simulation based algorithms of Section 4 to a portfolio management problem, where the available investment options include both liquid and non-liquid assets. In the interest of understanding the performance of the different algorithms, we consider a rather simplistic model of the corresponding financial problem. We emphasize that dealing with richer models requires no change in the algorithms.

We consider a portfolio that is composed of two types of assets. A liquid asset (e.g., short term T-bills), which has a fixed interest rate r_l but may be sold at every time step $t = 1, \ldots, T$, and a non-liquid asset (e.g., low liquidity bonds or options) that has a time dependent interest rate $r_{nl}(t)$, yet may be sold only after a maturity period of N steps. In addition, the non-liquid asset has some risk of not being paid (i.e., a default) with a probability p_{risk}. A common investment strategy in this setup is *laddering*–splitting the investment in the non-liquid assets to chunks that are reinvested in regular intervals, such that a regular cash flow is maintained. In our model, at each time step the investor may change his portfolio by investing a fixed fraction α of his total available cash in a non-liquid asset. Of course, he can only do that when he has at least α invested in liquid assets, otherwise he has to wait until enough non-liquid assets mature. In addition, we assume that at each t the interest rate $r_{nl}(t)$ takes one of two values - r_{nl}^{high} or r_{nl}^{low}, and the transitions between these values occur stochastically with switching probability p_{switch}. The state of the model at each time step is represented by a vector $x(t) \in \mathbb{R}^{N+2}$, where $x_1 \in [0,1]$ is the fraction of the investment in liquid assets, $x_2, \ldots, x_{N+1} \in [0,1]$ is the fraction in non-liquid assets with time to maturity of $1, \ldots, N$ time steps, respectively, and $x_{N+2}(t) = r_{nl}(t) - \mathbb{E}[r_{nl}(t)]$. At time $t = 0$ we assume that all investments are in liquid assets, and we denote $x^* = x(t = 0)$. The binary action at each step is determined by a stochastic policy, with probability $\mu_\theta(x) = \epsilon + (1 - 2\epsilon)/\left(1 + e^{-\theta x}\right)$ of investing in a non-liquid asset. Note that this 'ϵ-constrained' softmax policy comes to satisfy Assumption 2.3. Our reward is just the logarithm of the return from the investment (which is additive at each step). The dynamics of the investment chunks are illustrated in Figure 2.

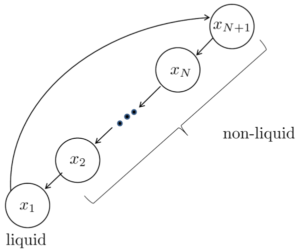


Figure 2. Dynamics of the investment.

We optimized the policy parameters using the simulation based algorithms of Section 4 with three different performance criteria: (a) Average reward: $\max J(x^*)$, (b) Variance constrained reward $\max J(x^*)$ s.t. $V(x^*) \leq b$, and (c) the SR $\max J(x^*)\sqrt{V(x^*)}$. Figure 3 shows the distribution of the accumulated reward. As anticipated, the policy for criterion (a) was risky, and yielded higher gain than the policy for the variance constrained criterion (b). Interestingly, maximizing the SR resulted in a very conservative policy, that almost never invested in the non-liquid asset. The parameters for the experiments are detailed in the supplementary material, Section C.

6. Conclusion

This work presented a novel algorithmic approach for RL with variance related risk criteria, a subject that while being important for many applications, has been

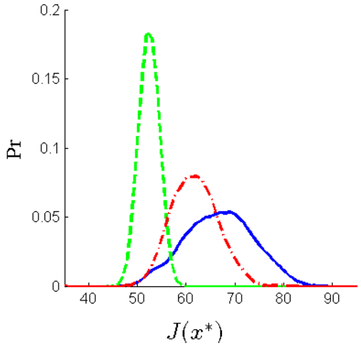


Figure 3. Distribution of the accumulated reward. Solid line: corresponds to the policy obtained by maximizing total reward. Dash-dotted line: maximizing total reward s.t. variance less than 20. Dashed line : maximize the SR.

notoriously known to pose significant algorithmic challenges. Since getting to an optimal solution seems hard even when the model is known, we adopted a gradient based approach that achieves local optimality.

A few issues are in need of further investigation. First, we note a possible extension to other risk measures such as the percentile criterion (Delage & Mannor, 2010). This will require a result reminiscent to Proposition 3.1 that would allow us to drive the optimization. Second, we could consider variance in the optimization process to improve convergence time in the style of *control variates*. Policy gradient algorithms are known to suffer from high variance when the recurrent state in not visited frequently. One technique for dealing with this difficulty is by using control variates (Greensmith et al., 2004). Imposing a variance constraint as described in this work also acts along this direction, and may in fact improve performance of such algorithms even if variance is not part of the criterion we are optimizing. Third, policy gradients are just one family of algorithms we can consider. It would be interesting to see if a temporal-difference style algorithm can be developed for the risk measures considered here. Lastly, we note that experimentally, maximizing the SR resulted in a very risk averse behavior. This interesting phenomenon deserves more research. It suggests that it might be more prudent to consider other risk measures instead of the SR.

Acknowledgements

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under PASCAL2 (PUMP PRIMING) grant agreement no. 216886 and under a Marie Curie Reintegration Fellowship (IRG) grant agreement no. 249254.

References

Baxter, J. and Bartlett, P. L. Infinite-horizon policy-gradient estimation. *JAIR*, 15:319–350, 2001.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol II.* Athena Scientific, third edition, 2006.

Bertsekas, D. P. and Tsitsiklis, J. N. Neuro-dynamic programming. *Athena Scientific*, 1996.

Borkar, V. S. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291 – 294, 1997.

Borkar, V. S. and Meyn, S. P. Risk-sensitive optimal control for markov decision processes with monotone cost. *Math. Oper. Res.*, 27(1):192–209, 2002.

Boyd, S. P. and Vandenberghe, L. *Convex optimization.* Cambridge Univ Pr, 2004.

Delage, E. and Mannor, S. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58(1):203–213, 2010.

Filar, J. A., Krass, D., and Ross, K. W. Percentile performance criteria for limiting average markov decision processes. *IEEE Trans. Auto. Control*, 40(1):2–10, 1995.

Geibel, P. and Wysotzki, F. Risk-sensitive reinforcement learning applied to control under constraints. *JAIR*, 24 (1):81–108, 2005.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *JMLR*, 5:1471–1530, 2004.

Howard, R. A. and Matheson, J. E. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

Leslie, D. S. and Collins, E.J. Convergent multiple-timescales reinforcement learning algorithms in normal form games. *Annals of App. Prob.*, 13:1231–1251, 2002.

Luenberger, D. *Investment Science.* Oxford University Press, 1998.

Mannor, S. and Tsitsiklis, J. N. Mean-variance optimization in markov decision processes. In *ICML*.

Marbach, P. and Tsitsiklis, J. N. Simulation-based optimization of markov reward processes. *IEEE Trans. Auto. Control*, 46(2):191–209, 1998.

Nilim, A. and El Ghaoui, L. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Poupart, P., Vlassis, N., Hoey, J., and Regan, K. An analytic solution to discrete bayesian reinforcement learning. In *ICML*, 2006.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, Inc., 1994.

Sharpe, W. F. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

Sobel, M. J. The variance of discounted markov decision processes. *J. Applied Probability*, pp. 794–802, 1982.

Chapter 3

Temporal Difference Methods for the Variance of the Reward-To-Go

This chapter was published as:

A. Tamar, D. Di Castro, and S. Mannor. Temporal difference methods for the variance of the reward-to-go. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, JMLR Workshop and Conference Proceedings 28 (3) :495-503, 2013.

Abbreviations

TD	—	Temporal differences
$LSTD$	—	Least-squares temporal differences
MDP	—	Markov decision process
RL	—	Reinforcement learning
SSP	—	Stochastic shortest-path problem
RMS	—	Root mean square
MC	—	Monte carlo

Notations

J	—	Expected reward-to-go
V	—	Variance of reward-to-go
M	—	Second moment of reward-to-go
X	—	State space
x	—	State
x_k	—	State at time k
x^*	—	Terminal state
r	—	Reward
P	—	Transition probability
π	—	Policy
τ	—	First visit time to terminal state
B	—	Accumulated reward random variable
R	—	Diagonal matrix of reward
T	—	Bellman operator on augmented space
$\tilde{J}$	—	Approximate expected reward-to-go
$\tilde{M}$	—	Approximate second moment of reward-to-go
$\phi_J(x)$	—	State-dependent features for J
$\phi_M(x)$	—	State-dependent features for M
Φ_J	—	Features matrix for J
Φ_M	—	Features matrix for M
w_J	—	Weight vector for J
w_M	—	Weight vector for M
S_J	—	Approximation subspace for J
S_M	—	Approximation subspace for M

q	—	State occupancy probabilities
Q	—	Diagonal matrix of q
$\|\cdot\|_q$	—	A q-weighted Euclidean norm
z^*	—	Fixed point
w_J^*	—	Fixed point weights for J
w_M^*	—	Fixed point weights for M
ξ	—	Step-size

Temporal Difference Methods for the Variance of the Reward To Go

Aviv Tamar AVIVT@TX.TECHNION.AC.IL
Dotan Di Castro DOT@TX.TECHNION.AC.IL
Shie Mannor SHIE@EE.TECHNION.AC.IL
Department of Electrical Engineering, The Technion - Israel Institute of Technology, Haifa, Israel 32000

Abstract

In this paper we extend temporal difference policy evaluation algorithms to performance criteria that include the variance of the cumulative reward. Such criteria are useful for risk management, and are important in domains such as finance and process control. We propose variants of both TD(0) and LSTD(λ) with linear function approximation, prove their convergence, and demonstrate their utility in a 4-dimensional continuous state space problem.

1. Introduction

In sequential decision making within the Markov Decision Process (MDP) framework, policy evaluation refers to the process of mapping each state of the system to some statistical property of its long-term outcome, most commonly its *expected reward to go*. In the fields of Reinforcement Learning (RL; Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998) and planning in MDPs (Puterman, 1994), policy evaluation is a fundamental step in many policy improvement algorithms. Yet in domains where policies are mostly hand designed, for example in clinical decision making, policy evaluation is also important, for a prudent choice of strategy must depend on it (Shortreed et al., 2011).

A principal challenge in policy evaluation arises when the state space is large, or continuous, necessitating some means of *approximation* for the process to be tractable. This difficulty is even more pronounced when a model of the process is not available, and the evaluation has to be *estimated* from a limited amount of samples. Fortunately, for the case of the expected reward to go, also known as the value function and denoted by J, the sequential nature of the problem may be exploited to overcome these difficulties. Temporal Difference methods (TD; Sutton, 1988) employ *function approximation* to represent J in a lower dimensional subspace, and tune the approximation parameters efficiently from data. Enjoying both theoretical guarantees (Bertsekas, 2012; Lazaric et al., 2010) and empirical success (Tesauro, 1995), these methods are considered the state of the art in policy evaluation.

However, when it comes to evaluating additional statistics of the reward to go, such as its variance, little is known. This is due to the fact that the expectation plays a key role in the Bellman equation, which drives TD algorithms.

Yet, the incentives to evaluate such statistics are extensive. In the context of RL and planning, incorporating such statistics into the performance evaluation criteria leads to *risk sensitive* optimization, a topic that has gained significant interest recently (Filar et al., 1995; Mihatsch & Neuneier, 2002; Geibel & Wysotzki, 2005; Mannor & Tsitsiklis, 2011). In a more general context, uncertainty in a policy's long-term outcome is critical for decision making in many areas, such as financial, process control, and clinical domains. In these domains, considering the variance of the total reward is particularly important, as it is both common-practice and intuitive to understand (Sharpe, 1966; Shortreed et al., 2011).

In this paper we present a TD framework for estimating the *variance of the reward to go*, denoted by V, using function approximation, in problems where a model is not available. To our knowledge, this is the first work that addresses the challenge of large state spaces, by considering an approximation scheme for V. Our approach is based on the following observation: the second moment of the reward to go, denoted by M, together with the value function J, obey a linear 'Bellman-like' equation. By extending TD methods to jointly estimate J and M with linear function approximation, we obtain a solution for estimating the

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

variance, using the relation $V = M - J^2$.

We propose both a variant of Least Squares Temporal Difference (LSTD) (Boyan, 2002) and of TD(0) (Sutton & Barto, 1998) for jointly estimating J and M with a linear function approximation. For these algorithms, we provide convergence guarantees and error bounds. In addition, we introduce a novel method for enforcing the approximate variance to be positive, through a constrained TD equation. An empirical evaluation on a challenging continuous maze problem demonstrates the applicability of our approach to large domains, and highlights the importance of the variance function in understanding the risk of a policy.

A previous study by Sato et al. (2001) suggested TD equations for J and V, without function approximation. Their approach relied on a non-linear equation for V, and it is not clear how it may be extended to handle large state spaces. More recently, Morimura et al. (2012) proposed TD learning rules for a parametric distribution of the return, albeit without function approximation nor formal guarantees. In the Bayesian GPTD framework of Engel et al. (2005), the reward-to-go is assumed to have a Gaussian posterior distribution, and its mean and variance are estimated. However, the resulting variance is a product of both stochastic transitions and model uncertainty, and is thus different than the variance considered here.

2. Framework and Background

We consider a Stochastic Shortest Path (SSP) problem[1,2] (Bertsekas, 2012), where the environment is modeled by an MDP in discrete time with a finite state space $X \triangleq \{1, \ldots, n\}$ and a terminal state x^*. A fixed policy π determines, for each $x \in X$, a stochastic transition to a subsequent state $x' \in \{X \cup x^*\}$ with probability $P(x'|x)$. We consider a deterministic and bounded reward function $r : X \to \mathbb{R}$, and assume zero reward at the terminal state. We denote by x_k the state at time k, where $k = 0, 1, 2, \ldots$.

A policy is said to be *proper* (Bertsekas, 2012) if there is a positive probability that the terminal state x^* will be reached after at most n transitions, from any initial state. In this paper we make the following assumption

Assumption 1. *The policy π is proper.*

Let $\tau \triangleq \min\{k > 0 | x_k = x^*\}$ denote the first visit time to the terminal state, and let the random variable B denote the accumulated reward along the trajectory until that time

$$B \triangleq \sum_{k=0}^{\tau-1} r(x_k).$$

In this work, we are interested in the mean-variance tradeoff in B, represented by the *value function*

$$J(x) \triangleq \mathbb{E}\left[B | x_0 = x\right], \quad x \in X,$$

and the *variance of the reward to go*

$$V(x) \triangleq \mathrm{Var}\left[B | x_0 = x\right], \quad x \in X.$$

We will find it convenient to define also the *second moment of the reward to go*

$$M(x) \triangleq \mathbb{E}\left[B^2 | x_0 = x\right], \quad x \in X.$$

Our goal is to estimate $J(x)$ and $V(x)$ from trajectories obtained by simulating the MDP with policy π.

3. Approximation of the Variance of the Reward To Go

In this section we derive a projected equation method for approximating $J(x)$ and $M(x)$ using linear function approximation. The estimation of $V(x)$ will then follow from the relation $V(x) = M(x) - J(x)^2$.

Our starting point is a system of equations for $J(x)$ and $M(x)$, first derived by Sobel (1982) for a discounted infinite horizon case, and extended here to the SSP case. The equation for J is the well known Bellman equation for a fixed policy, and independent of the equation for M.

Proposition 2. *The following equations hold for $x \in X$*

$$J(x) = r(x) + \sum_{x' \in X} P(x'|x) J(x'), \tag{1}$$
$$M(x) = r(x)^2 + 2r(x) \sum_{x' \in X} P(x'|x) J(x') + \sum_{x' \in X} P(x'|x) M(x').$$

Furthermore, under Assumption 1 a unique solution to (1) exists.

A straightforward proof is given in Appendix A.

At this point the reader may wonder why an equation for V is not presented. While such an equation may be derived, as was done by Tamar et al. (2012), it is not linear. The linearity of (1) in J and M is the key to our approach. As we show in the next subsection,

[1] This is also known as an episodic setting.

[2] The popular infinite horizon discounted setting is actually simpler than the SSP considered here, as the discount factor simplifies the verification of the contraction properties presented in the sequel. Therefore, all of our results may easily be extended to that setting as well, with even simpler proofs.

the solution to (1) may be expressed as the fixed point of a linear mapping in the joint space of J and M. We will then show that a projection of this mapping onto a linear feature space is contracting, thus allowing us to use existing TD theory to derive estimation algorithms for J and M.

3.1. A Projected Fixed Point Equation in the Joint Space of J and M

For the sequel, we introduce the following vector notations. We denote by $P \in \mathbb{R}^{n\times n}$ and $r \in \mathbb{R}^n$ the SSP transition matrix and reward vector, i.e., $P_{x,x'} = P(x'|x)$ and $r_x = r(x)$, where $x, x' \in X$. Also, we define the diagonal matrix $R \triangleq diag(r)$.

For a vector $z \in \mathbb{R}^{2n}$ we let $z_J \in \mathbb{R}^n$ and $z_M \in \mathbb{R}^n$ denote its leading and ending n components, respectively. Thus, such a vector belongs to the joint space of J and M.

We define the mapping $T : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by

$$\begin{aligned} [Tz]_J &= r + Pz_J, \\ [Tz]_M &= Rr + 2RPz_J + Pz_M. \end{aligned} \tag{2}$$

It may easily be verified that a fixed point of T is a solution to (1), and by Proposition 2 such a fixed point exists and is unique.

When the state space X is large, a direct solution of (1) is not feasible, even if P may be accurately obtained. A popular approach in this case is to approximate $J(x)$ by restricting it to a lower dimensional subspace, and use simulation based TD algorithms to adjust the approximation parameters (Bertsekas, 2012). In this paper we extend this approach to the approximation of $M(x)$ as well.

We consider a linear approximation architecture of the form

$$\tilde{J}(x) = \phi_J(x)^\top w_J, \quad \tilde{M}(x) = \phi_M(x)^\top w_M,$$

where $w_J \in \mathbb{R}^l$ and $w_M \in \mathbb{R}^m$ are the approximation parameter vectors, $\phi_J(x) \in \mathbb{R}^l$ and $\phi_M(x) \in \mathbb{R}^m$ are state dependent features, and $(\cdot)^\top$ denotes the transpose of a vector. The low dimensional subspaces are therefore

$$S_J = \{\Phi_J w | w \in \mathbb{R}^l\}, \quad S_M = \{\Phi_M w | w \in \mathbb{R}^m\},$$

where Φ_J and Φ_M are matrices whose rows are $\phi_J(x)^\top$ and $\phi_M(x)^\top$, respectively. We make the following standard independence assumption on the features

Assumption 3. *The matrix Φ_J has rank l and the matrix Φ_M has rank m.*

As outlined earlier, our goal is to estimate w_J and w_M from simulated trajectories of the MDP. Thus, it is constructive to consider projections onto S_J and S_M with respect to a norm that is weighted according to the state occupancy in these trajectories.

For a trajectory $x_0, \ldots, x_{\tau-1}$, where x_0 is drawn from a fixed distribution $\zeta_0(x)$, and the states evolve according to the MDP with policy π, define the state occupancy probabilities

$$q_t(x) = P(x_t = x), \quad x \in X, \quad t = 0, 1, \ldots$$

and let

$$\begin{aligned} q(x) &= \sum_{t=0}^{\infty} q_t(x), \quad x \in X \\ Q &\triangleq diag(q). \end{aligned}$$

We make the following assumption on the policy π and initial distribution ζ_0

Assumption 4. *Each state has a positive probability of being visited, namely, $q(x) > 0$ for all $x \in X$.*

For vectors in $\mathbb{R}^n$, we introduce the weighted Euclidean norm

$$\|y\|_q = \sqrt{\sum_{i=1}^{n} q(i) \left(y(i)\right)^2}, \quad y \in \mathbb{R}^n,$$

and we denote by Π_J and Π_M the projections from $\mathbb{R}^n$ onto the subspaces S_J and S_M, respectively, with respect to this norm. For $z \in \mathbb{R}^{2n}$ we denote by Π the projection of z_J onto S_J and z_M onto S_M, namely [3]

$$\Pi = \begin{pmatrix} \Pi_J & 0 \\ 0 & \Pi_M \end{pmatrix}. \tag{3}$$

We are now ready to fully describe our approximation scheme. We consider the *projected* fixed point equation

$$z = \Pi T z, \tag{4}$$

and, letting z^* denote its solution, propose the approximate value function $\tilde{J} = z_J^* \in S_J$ and second moment function $\tilde{M} = z_M^* \in S_M$.

We proceed to derive some properties of the projected fixed point equation (4). We begin by stating a well known result regarding the contraction properties of the *projected Bellman operator* $\Pi_J T_J$, where $T_J y = r + Py$. A proof can be found at (Bertsekas, 2012), proposition 7.1.1.

[3] The projection operators Π_J and Π_M are linear, and may be written explicitly as $\Pi_J = \Phi_J(\Phi_J^\top Q \Phi_J)^{-1}\Phi_J^\top Q$, and similarly for Π_M.

Lemma 5. *Let Assumptions 1, 3, and 4 hold. Then, there exists some norm $\|\cdot\|_J$ and some $\beta_J < 1$ such that*

$$\|\Pi_J P y\|_J \leq \beta_J \|y\|_J, \quad \forall y \in \mathbb{R}^n.$$

Similarly, there exists some norm $\|\cdot\|_M$ and some $\beta_M < 1$ such that

$$\|\Pi_M P y\|_M \leq \beta_M \|y\|_M, \quad \forall y \in \mathbb{R}^n.$$

Next, we define a weighted norm on $\mathbb{R}^{2n}$

Definition 6. *For a vector $z \in \mathbb{R}^{2n}$ and a scalar $0 < \alpha < 1$, the α-weighted norm is*

$$\|z\|_\alpha = \alpha \|z_J\|_J + (1-\alpha)\|z_M\|_M, \tag{5}$$

where $\|\cdot\|_J$ and $\|\cdot\|_M$ are defined in Lemma 5.

Our main result of this section is given in the following proposition, where we show that the projected operator ΠT is a contraction with respect to the α-weighted norm.

Proposition 7. *Let Assumptions 1, 3, and 4 hold. Then, there exists some $0 < \alpha < 1$ and some $\beta < 1$ such that ΠT is a β-contraction with respect to the α-weighted norm, i.e.,*

$$\|\Pi T z_1 - \Pi T z_2\|_\alpha \leq \beta \|z_1 - z_2\|_\alpha, \quad \forall z_1, z_2 \in \mathbb{R}^{2n}.$$

Proof. First, using (2) and (3) we have that $\|\Pi T z_1 - \Pi T z_2\|_\alpha = \|\Pi \mathcal{P}(z_1 - z_2)\|_\alpha$, where

$$\Pi \mathcal{P} = \begin{pmatrix} \Pi_J P & 0 \\ 2\Pi_M R P & \Pi_M P \end{pmatrix}.$$

Thus, it suffices to show that for all $z \in \mathbb{R}^{2n}$

$$\|\Pi \mathcal{P} z\|_\alpha \leq \beta \|z\|_\alpha.$$

We will now show that $\|\Pi \mathcal{P} z\|_\alpha$ may be separated into two terms which may be bounded by Lemma 5, and an additional cross term. By balancing α and β, this term may be contained to yield the required contraction.

We have

$$\begin{aligned} \|\Pi \mathcal{P} z\|_\alpha =& \alpha \|\Pi_J P z_J\|_J \\ &+ (1-\alpha)\|2\Pi_M R P z_J + \Pi_M P z_M\|_M \\ \leq& \alpha \|\Pi_J P z_J\|_J + (1-\alpha)\|\Pi_M P z_M\|_M \\ &+ (1-\alpha)\|2\Pi_M R P z_J\|_M \\ \leq& \alpha \beta_J \|z_J\|_J + (1-\alpha)\beta_M \|z_M\|_M \\ &+ (1-\alpha)\|2\Pi_M R P z_J\|_M, \end{aligned} \tag{6}$$

where the equality is by definition of the α weighted norm (5), the first inequality is from the triangle inequality, and the second inequality is by Lemma 5. Now, we claim that there exists some finite C such that

$$\|2\Pi_M R P y\|_M \leq C \|y\|_J, \quad \forall y \in \mathbb{R}^n. \tag{7}$$

To see this, note that since $\mathbb{R}^n$ is a finite dimensional real vector space, all vector norms are equivalent (Horn & Johnson, 1985) therefore there exist finite C_1 and C_2 such that for all $y \in \mathbb{R}^n$

$$C_1 \|2\Pi_M R P y\|_2 \leq \|2\Pi_M R P y\|_M \leq C_2 \|2\Pi_M R P y\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Let λ denote the spectral norm of the matrix $2\Pi_M R P$, which is finite since all the matrix elements are finite. We have that

$$\|2\Pi_M R P y\|_2 \leq \lambda \|y\|_2, \quad \forall y \in \mathbb{R}^n.$$

Using again the fact that all vector norms are equivalent, there exists a finite C_3 such that

$$\|y\|_2 \leq C_3 \|y\|_J, \quad \forall y \in \mathbb{R}^n.$$

Setting $C = C_2 \lambda C_3$ we get the desired bound. Let $\tilde{\beta} = \max\{\beta_J, \beta_M\} < 1$, and choose $\epsilon > 0$ such that

$$\tilde{\beta} + \epsilon < 1.$$

Now, choose α such that $\alpha = \frac{C}{\epsilon + C}$. We have that

$$(1-\alpha)C = \alpha\epsilon,$$

and plugging this into (7) yields

$$(1-\alpha)\|2\Pi_M R P y\|_M \leq \alpha \epsilon \|y\|_J. \tag{8}$$

We now return to (6), where we have

$$\begin{aligned} &\alpha \beta_J \|z_J\|_J + (1-\alpha)\beta_M \|z_M\|_M + (1-\alpha)\|2\Pi_M R P z_J\|_M \\ \leq& \alpha \beta_J \|z_J\|_J + (1-\alpha)\beta_M \|z_M\|_M + \alpha\epsilon \|z_J\|_J \\ \leq& (\tilde{\beta} + \epsilon)\left(\alpha \|z_J\|_J + (1-\alpha)\|z_M\|_M\right), \end{aligned}$$

where the first inequality is by (8), and the second is by the definition of $\tilde{\beta}$. We have thus shown that

$$\|\Pi \mathcal{P} z\|_\alpha \leq (\tilde{\beta} + \epsilon)\|z\|_\alpha.$$

Finally, choose $\beta = \tilde{\beta} + \epsilon$. □

Proposition 7 guarantees that the projected operator ΠT has a unique fixed point. Let us denote this fixed point by z^*, and let w_J^*, w_M^* denote the corresponding weights, which are unique due to Assumption 3

$$\begin{aligned} \Pi T z^* &= z^*, \\ z_J^* &= \Phi_J w_J^*, \\ z_M^* &= \Phi_M w_M^*. \end{aligned} \tag{9}$$

In the next proposition we provide a bound on the approximation error. The proof is in Appendix B.

Proposition 8. *Let Assumptions 1, 3, and 4 hold. Denote by $z_{true} \in \mathbb{R}^{2n}$ the true value and second moment functions, i.e., $[z_{true}]_J = J$, and $[z_{true}]_M = M$. Then,*

$$\|z_{true} - z^*\|_\alpha \leq \frac{1}{1-\beta}\|z_{true} - \Pi z_{true}\|_\alpha,$$

with α and β defined in Proposition 7.

4. Simulation Based Estimation Algorithms

In this section we propose algorithms that estimate $\tilde{J}$ and $\tilde{M}$ from sampled trajectories of the MDP, based on the approximation architecture of the previous section.

We begin by writing the projected equation (9) in matrix form. First, let us write the equation explicitly as

$$\begin{aligned} \Pi_J\left(r + P\Phi_J w_J^*\right) &= \Phi_J w_J^*, \\ \Pi_M\left(Rr + 2RP\Phi_J w_J^* + P\Phi_M w_M^*\right) &= \Phi_M w_M^*. \end{aligned} \qquad (10)$$

Projecting a vector y onto Φw satisfies the following orthogonality condition

$$\Phi^\top Q(y - \Phi w) = 0,$$

we therefore have

$$\begin{aligned} \Phi_J^\top Q\left(\Phi_J w_J^* - (r + P\Phi_J w_J^*)\right) &= 0, \\ \Phi_M^\top Q\left(\Phi_M w_M^* - (Rr + 2RP\Phi_J w_J^* + P\Phi_M w_M^*)\right) &= 0, \end{aligned}$$

which can be written as

$$Aw_J^* = b, \quad Cw_M^* = d, \qquad (11)$$

with

$$\begin{aligned} &A = \Phi_J^\top Q\left(I - P\right)\Phi_J, \quad b = \Phi_J^\top Qr, \qquad (12) \\ &C = \Phi_M^\top Q\left(I - P\right)\Phi_M, \quad d = \Phi_M^\top QR\left(r + 2P\Phi_J A^{-1} b\right), \end{aligned}$$

and the matrices A and C are invertible since Proposition 7 guarantees a unique solution to (9) and Assumption 3 guarantees the unique weights of its projection.

4.1. A Least Squares TD Algorithm

Our first simulation-based algorithm is an extension of the Least Squares Temporal Difference (LSTD) algorithm (Boyan, 2002). We simulate N trajectories of the MDP with the policy π and initial state distribution ζ_0. Let $x_0^k, x_1^k, \ldots, x_{\tau^k - 1}^k$ and τ^k, where $k = 0, 1, \ldots, N$, denote the state sequence and visit times to the terminal state within these trajectories, respectively. We now use these trajectories to form the following estimates of the terms in (12)

$$\begin{aligned} A_N &= \mathbb{E}_N\left[\sum_{t=0}^{\tau-1} \phi_J(x_t)(\phi_J(x_t) - \phi_J(x_{t+1}))^\top\right], \\ b_N &= \mathbb{E}_N\left[\sum_{t=0}^{\tau-1} \phi_J(x_t) r(x_t)\right], \qquad (13) \\ C_N &= \mathbb{E}_N\left[\sum_{t=0}^{\tau-1} \phi_M(x_t)(\phi_M(x_t) - \phi_M(x_{t+1}))^\top\right], \\ d_N &= \mathbb{E}_N\left[\sum_{t=0}^{\tau-1} \phi_M(x_t) r(x_t)\big(r(x_t) + 2\phi_J(x_{t+1})^\top A_N^{-1} b_N\big)\right], \end{aligned}$$

where $\mathbb{E}_N$ denotes an empirical average over trajectories, i.e., $\mathbb{E}_N\left[f(x,\tau)\right] = \frac{1}{N}\sum_{k=1}^N f(x^k, \tau^k)$. The LSTD approximation is given by

$$\hat{w}_J^* = A_N^{-1} b_N, \quad \hat{w}_M^* = C_N^{-1} d_N.$$

The next theorem shows that LSTD converges.

Theorem 9. *Let Assumptions 1, 3, and 4 hold. Then $\hat{w}_J^* \to w_J^*$ and $\hat{w}_M^* \to w_M^*$ as $N \to \infty$ with probability 1.*

The proof involves a straightforward application of the law of large numbers and is described in Appendix C. Convergence rates for regular LSTD were derived by Konda (2002) and Lazaric et al. (2010), and may be extended to the algorithm presented here. This issue is deferred to the full version of this paper.

4.2. An Online TD(0) Algorithm

Our second estimation algorithm is an extension of the well known TD(0) algorithm (Sutton & Barto, 1998). Again, we simulate trajectories of the MDP corresponding to the policy π and initial state distribution ζ_0, and we iteratively update our estimates at every visit to the terminal state[4]. For some $0 \leq t < \tau^k$ and weights w_J, w_M, we introduce the TD terms

$$\begin{aligned} \delta_J^k(t, w_J, w_M) =& r(x_t^k) + \left(\phi_J(x_{t+1}^k)^\top - \phi_J(x_t^k)^\top\right) w_J, \\ \delta_M^k(t, w_J, w_M) =& r^2(x_t^k) + 2r(x_t^k)\phi_J(x_{t+1}^k)^\top w_J \\ &+ \left(\phi_M(x_{t+1}^k)^\top - \phi_M(x_t^k)^\top\right) w_M. \end{aligned}$$

Note that δ_J^k is the standard TD error (Sutton & Barto, 1998). For the intuition behind δ_M^k, observe that M in (1) is equivalent to the value function of an MDP with stochastic reward $r(x)^2 + 2r(x)J(x')$, where $x' \sim P(x'|x)$. δ_M^k is then the equivalent TD error, with $\phi_J(x')^\top w_J$ substituting $J(x')$. The TD(0) algorithm

[4] An extension to an algorithm that updates at every state transition is possible, but we do not pursue such here.

is given by

$$\hat{w}_{J;k+1} = \hat{w}_{J;k} + \xi_k \sum_{t=0}^{\tau^k-1} \phi_J(x_t)\delta_J^k(t, \hat{w}_{J;k}, \hat{w}_{M;k}),$$
$$\hat{w}_{M;k+1} = \hat{w}_{M;k} + \xi_k \sum_{t=0}^{\tau^k-1} \phi_M(x_t)\delta_M^k(t, \hat{w}_{J;k}, \hat{w}_{M;k}),$$

where $\{\xi_k\}$ are positive step sizes.

The next theorem shows that TD(0) converges.

Theorem 10. *Let Assumptions 1, 3, and 4 hold, and let the step sizes satisfy*

$$\sum_{k=0}^{\infty} \xi_k = \infty, \quad \sum_{k=0}^{\infty} \xi_k^2 < \infty.$$

Then $\hat{w}_{J;k} \to w_J^*$ and $\hat{w}_{M;k} \to w_M^*$ as $k \to \infty$ with probability 1.

The proof, provided in Appendix D, is based on representing the algorithm as a stochastic approximation, and using a result of Borkar (2008) to show that the iterates asymptotically track a certain ordinary differential equation (ODE). This ODE is then shown to have a unique asymptotically stable equilibrium exactly at w_J^*, w_M^*. Convergence rates for TD(0) may be derived along the lines of Konda (2002), with the details deferred to the full version of this paper.

4.3. Multistep LSTD(λ) Algorithms

A common method in value function approximation is to replace the single step mapping T_J with a multistep version of the form

$$T_J^{(\lambda)} = (1-\lambda)\sum_{l=0}^{\infty} \lambda^l T_J^{l+1}$$

with $0 < \lambda < 1$. The projected equation (10) then becomes $\Pi_J T_J^{(\lambda)}\left(\Phi_J w_J^{*(\lambda)}\right) = \Phi_J w_J^{*(\lambda)}$. Similarly, we may write a multistep equation for M

$$\Pi_M T_M^{(\lambda)}\left(\Phi_M w_M^{*(\lambda)}\right) = \Phi_M w_M^{*(\lambda)}, \qquad (14)$$

where

$$T_M^{(\lambda)} = (1-\lambda)\sum_{l=0}^{\infty} \lambda^l T_{M^*}^{l+1},$$

and

$$T_{M^*}(y) = Rr + 2RP\Phi_J w_J^{*(\lambda)} + Py.$$

Note the difference between T_{M^*} and $[T]_M$ defined earlier; We are no longer working on the joint space of J and M but instead we have an independent equation for approximating J, and its solution $w_J^{*(\lambda)}$ is part of Equation (14) for approximating M. By Proposition 7.1.1 of Bertsekas (2012) both $\Pi_J T_J^{(\lambda)}$ and $\Pi_M T_M^{(\lambda)}$ are contractions with respect to the weighted norm $\|\cdot\|_q$, therefore both multistep projected equations admit a unique solution. In a similar manner to the single step version, the projected equations may be written in matrix form

$$A^{(\lambda)} w_J^{*(\lambda)} = b^{(\lambda)}, \quad C^{(\lambda)} w_M^{*(\lambda)} = d^{(\lambda)}, \qquad (15)$$

where

$$A^{(\lambda)} = \Phi_J^\top Q\left(I - P^{(\lambda)}\right)\Phi_J, \quad b^{(\lambda)} = \Phi_J^\top Q(I-\lambda P)^{-1} r,$$
$$C^{(\lambda)} = \Phi_M^\top Q\left(I - P^{(\lambda)}\right)\Phi_M,$$
$$d^{(\lambda)} = \Phi_M^\top Q(I-\lambda P)^{-1} R\left(r + 2P\Phi_J w_J^{*(\lambda)}\right),$$

and $P^{(\lambda)} = (1-\lambda)\sum_{l=0}^{\infty} \lambda^l P^{l+1}$.

Simulation based estimates $A_N^{(\lambda)}$ and $b_N^{(\lambda)}$ of the expressions above may be obtained by using eligibility traces, as described by Bertsekas (2012), and the LSTD(λ) approximation is then given by $\hat{w}_J^{*(\lambda)} = (A_N^{(\lambda)})^{-1} b_N^{(\lambda)}$. By substituting $w_J^{*(\lambda)}$ with $\hat{w}_J^{*(\lambda)}$ in the expression for $d^{(\lambda)}$, a similar procedure may be used to derive estimates $C_N^{(\lambda)}$ and $d_N^{(\lambda)}$, and to obtain the LSTD(λ) approximation $\hat{w}_M^{*(\lambda)} = (C_N^{(\lambda)})^{-1} d_N^{(\lambda)}$. A convergence result similar to Theorem 9 may also be obtained. Due to the similarity to the LSTD procedure in (13), the exact details are omitted.

5. Non Negative Approximate Variance by Constrained Projection

The TD algorithms of the preceding section approximated J and M by the solution to the fixed point equation (9). While Proposition 8 shows that the approximation errors of $\tilde{J}$ and $\tilde{M}$ are bounded, it does not guarantee that the approximated variance $\tilde{V}$, given by $\tilde{M} - \tilde{J}^2$, is non-negative for all states. A trivial remedy is to set all negative values of $\tilde{V}$ to zero; however, by such we lose all information in these states. In this section we propose an alternative method, based on modifying the fixed point equation (9) to include constraints for variance non-negativeness. We thus obtain a different approximation architecture, in which a non-negative variance is inherent. We now present the constrained equation and discuss how its solution may be computed.

First, let us write the multistep equation for the second moment weights (14) with the projection operator as an explicit minimization

$$w_M^{*(\lambda)} = \arg\min_w \|\Phi_M w - \left(\tilde{r} + \tilde{\Phi} w_M^{*(\lambda)}\right)\|_q,$$

with

$$\tilde{\Phi} = P^{(\lambda)}\Phi_M, \quad \tilde{r} = (I - \lambda P)^{-1}\left(Rr + 2RP\Phi_J w_J^{*(\lambda)}\right).$$

Observe that a non-negative variance in some state x may be written as a *linear* inequality in $w_M^{*(\lambda)}$

$$\phi_M(x)^\top w_M^{*(\lambda)} - (\phi_J(x)^\top w_J^{*(\lambda)})^2 \geq 0.$$

We now propose to add such inequality constraints to the projection operator. Let $\{x_1, \ldots, x_s\}$ denote a set of states in which we demand that the variance be non-negative. Let $H \in \mathbb{R}^{s\times m}$ denote a matrix with the features $-\phi_M^\top(x_i)$ as its rows, and let $g \in \mathbb{R}^s$ denote a vector with elements $-(\phi_J(x_i)^\top w_J^{*(\lambda)})^2$. We write the non-negative-variance projected equation for the second moment as

$$w_M^+ = \begin{cases} \arg\min_w & \|\Phi_M w - \left(\tilde{r} + \tilde{\Phi} w_M^+\right)\|_q \\ \text{s.t.} & Hw \leq g \end{cases} \qquad (16)$$

Here, w_M^+ denotes the weights of $\tilde{M}$ in the modified approximation architecture. We now discuss whether a solution to (16) exists, and how it may be obtained.

Let us assume that the constraints in (16) admit a feasible solution:

Assumption 11. *There exists w such that $Hw < g$.*

Note that a trivial way to satisfy Assumption 11 is to have some feature vector that is positive for all states.

Equation (16) is a form of projected equation studied by Bertsekas (2011), the solution of which exists, and may be obtained by the following iterative procedure

$$w_{k+1} = \Pi_{\Xi,\hat{W}_M}[w_k - \gamma\Xi^{-1}(C^{(\lambda)}w_k - d^{(\lambda)})], \qquad (17)$$

where Ξ is an arbitrary positive definite matrix, and $\Pi_{\Xi,\hat{W}_M}$ denotes a projection onto the convex set $\hat{W}_M = \{w | Hw \leq g\}$ with respect to the Ξ weighted Euclidean norm. The following lemma, which is based on a convergence result of Bertsekas (2011), guarantees that algorithm (17) converges.

Lemma 12. *Assume $\lambda > 0$, and let Assumption 11 hold. Then (16) admits a unique solution w_M^+, and there exists $\bar{\gamma} > 0$ such that $\forall \gamma \in (0, \bar{\gamma})$ and $\forall w_0 \in \mathbb{R}^m$ the algorithm (17) converges at a linear rate to w_M^+.*

Proof. This is a direct application of the convergence result of Bertsekas (2011). The only nontrivial assumption that needs to be verified is that $T_M^{(\lambda)}$ is a contraction in the $\|\cdot\|_q$ norm (Proposition 1 in Bertsekas, 2011). For $\lambda > 0$ Proposition 7.1.1. of Bertsekas (2012) guarantees that $T_M^{(\lambda)}$ is indeed contracting in the $\|\cdot\|_q$ norm. □

Generally, $C^{(\lambda)}$, $d^{(\lambda)}$, and $w_J^{*(\lambda)}$ are not known in advance, and should be replaced in (17) with their simulation based estimates, $C_N^{(\lambda)}$, $d_N^{(\lambda)}$, and $\hat{w}_J^{*(\lambda)}$, proposed in the previous section. The convergence of these estimates, together with the result of Lemma 12, lead to the following convergence result, which is given without proof.

Theorem 13. *Consider the algorithm in (17) with $C^{(\lambda)}$, $d^{(\lambda)}$, and $w_J^{*(\lambda)}$ replaced by $C_N^{(\lambda)}$, $d_N^{(\lambda)}$, and $\hat{w}_J^{*(\lambda)}$, respectively, and with $k(N)$ replacing k for a specific N. Also, let the assumptions in Lemma 12 hold, and let $\gamma \in (0, \bar{\gamma})$, with $\bar{\gamma}$ defined in Lemma 12. Then $w_{k(N)} \to w_M^+$ as $N \to \infty$ and $k \to \infty$ almost surely.*

An in-depth study of the approximation architecture (16) is deferred to the full version of this paper. However, an illustration on a toy problem is provided in Appendix E.

6. Experiments

In this section we present numerical simulations of policy evaluation on a challenging continuous maze domain. The goal of this presentation is threefold; first, we show that the variance of the reward-to-go may be estimated successfully on a large state space. Second, the intuitive maze domain highlights the insight that may be gleaned from this variance, and third, we show that in terms of sample efficiency, our LSTD(λ) algorithm significantly outperforms the current state-of-the-art. We begin by describing the domain and then present our policy evaluation results.

The Pinball Domain (Konidaris & Barto, 2009) is a continuous 2-dimensional maze where a small ball needs to be maneuvered between obstacles to reach some target area, as depicted in Figure 1A. The ball is controlled by applying a constant force in one of the 4 directions at each time step, which causes acceleration in the respective direction. In addition, the ball's velocity is susceptible to additive Gaussian noise (zero mean, standard deviation 0.03) and friction (drag coefficient 0.995). The obstacles are sharply shaped, and collisions are fully elastic. The state of the ball is thus 4-dimensional $(x, y, \dot{x}, \dot{y})$, and the action set is discrete, with 4 available controls. The reward is -1 for all states until reaching the target. A Java implementation of the pinball domain used by Konidaris & Barto (2009) is available on-line[5] and was used for our simulations as well, with the addition of noise to the velocity.

A near-optimal policy π was obtained using SARSA (Sutton & Barto, 1998) with radial basis function fea-

[5] http://people.csail.mit.edu/gdk/software.html

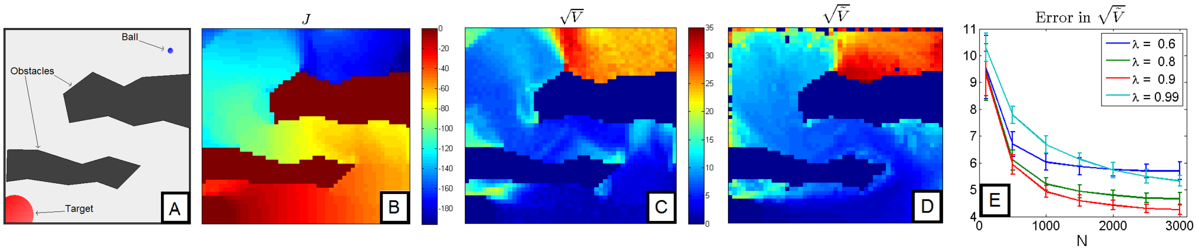


Figure 1. Experimental evaluation. A: The pinball domain. B,C: The 'true' value function J (left color bar) and standard deviation of the reward to go $\sqrt{V}$ (right color bar), estimated by monte carlo. D: Approximate standard deviation $\sqrt{\tilde{V}}$, using LSTD(0.9); same color bar as in (C). E: RMS error of $\sqrt{\tilde{V}}$ vs. number of trajectories N. Standard deviation error-bars from 10 runs are shown.

tures. The value J and standard deviation of the reward-to-go $\sqrt{V}$ for this policy are plotted in Figure 1(B;C), for 1816 equally spaced states between the obstacles with zero velocity. These plots were obtained by Monte Carlo (MC) estimation of the mean and variance, using over 2 million trajectories starting from these states. To our knowledge, MC is the current state-of-the-art technique for obtaining such variance estimates. As should be expected, the value is approximately a linear function of the distance to the target. In contrast, the standard deviation is clearly not linear in the distance, and in some places not even monotone. Furthermore, we see that an area in the top part of the maze before the first turn is very risky, even more than the farthest point from the target. We stress that this information cannot be gleaned from inspecting the value function alone.

Figure 1D shows the approximate standard deviation $\sqrt{\tilde{V}}$ obtained by the LSTD(λ) algorithm of Section 4.3. We used uniform tile features for $\tilde{J}$ and $\tilde{M}$ (50×50 non-overlapping tiles in x and y without dependence on velocity, for the same resolution as the MC estimate), and set $\lambda = 0.9$. To emphasize the efficiency of our method, we used only one sample trajectory per each state in the MC evaluation – a total of $N = 1816$ trajectories, with uniformly distributed initial states. Clearly, a single sample for each evaluation point is insufficient for a meaningful MC variance estimate. However, by exploiting relations *between* states (1), LSTD provides a reasonable approximation.

We further explore LSTD(λ) in Figure 1E, where we show the RMS error of $\sqrt{\tilde{V}}$ (compared to the MC estimate) for different values of λ and N. As in regular LSTD, λ trades off estimation bias and variance.

7. Conclusion

This work presented a novel framework for policy evaluation in RL with respect to the variance of the reward to go. We presented both formal guarantees and empirical evidence that this approach is useful in problems with a large state space. To the best of our knowledge, such problems are beyond the capabilities of previous approaches.

A requirement of variance evaluation is that it be non-negative. We approached this issue by adding constraints to the second moment approximation. An alternative is through the choice of features. Interestingly, in our experiments we found that using non-overlapping tile features produces a non-negative approximate variance. For this choice of features (identical for J and M), we can show that the *direct* approximation is always non-negative, i.e., $\Pi M - (\Pi J)^2 \geq 0$, where the square is element-wise. Whether this holds also for the fixed-point approximation, and if there are other features with this property, is an open question.

We conclude with a discussion on policy optimization with respect to a mean-variance tradeoff. While a naive variance-penalized policy iteration algorithm may be easily conceived, its usefulness should be questioned, as it was shown to be problematic for the standard deviation adjusted reward (Sobel, 1982) and the variance constrained reward (Mannor & Tsitsiklis, 2011). Perhaps a wiser approach would be to consider gradient based updates. Tamar et al. (2012) proposed policy gradient algorithms for a class of variance related criteria, and showed their convergence to local optima. These algorithms may be extended to use the variance function in an actor-critic type scheme. Such a study is left for future research.

Acknowledgments

The research leading to these results has received funding from the European Research Counsel under the European Union's Seventh Framework Program (FP7/2007-2013) / ERC Grant Agreement No 306638.

References

Bertsekas, D. P. Temporal difference methods for general projected equations. *IEEE Trans. Auto. Control*, 56(9):2128–2139, 2011.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol II.* Athena Scientific, fourth edition, 2012.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint.* Cambridge Univ Press, 2008.

Boyan, J. A. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2): 233–246, 2002.

Engel, Y., Mannor, S., and Meir, R. Reinforcement learning with Gaussian processes. In *ICML*, 2005.

Filar, J. A., Krass, D., and Ross, K. W. Percentile performance criteria for limiting average Markov decision processes. *IEEE Trans. Auto. Control*, 40(1): 2–10, 1995.

Geibel, P. and Wysotzki, F. Risk-sensitive reinforcement learning applied to control under constraints. *JAIR*, 24(1):81–108, 2005.

Horn, R. A. and Johnson, C. R. *Matrix Analysis.* Cambridge University Press, 1985.

Konda, V. *Actor-Critic Algorithms.* PhD thesis, Dept. Comput. Sci. Elect. Eng., MIT, Cambridge, MA, 2002.

Konidaris, G. D. and Barto, A. G. Skill discovery in continuous reinforcement learning domains using skill chaining. In *NIPS*, 2009.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of LSTD. In *ICML*, 2010.

Mannor, S. and Tsitsiklis, J. N. Mean-variance optimization in Markov decision processes. In *ICML*, 2011.

Mihatsch, O. and Neuneier, R. Risk-sensitive reinforcement learning. *Machine Learning*, 49(2):267–290, 2002.

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., and Tanaka, T. Parametric return density estimation for reinforcement learning. *arXiv preprint arXiv:1203.3497*, 2012.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, Inc., 1994.

Sato, M., Kimura, H., and Kobayashi, S. TD algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, 16:353–362, 2001.

Sharpe, W. F. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J., and Murphy, S. A. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84(1):109–136, 2011.

Sobel, M. J. The variance of discounted Markov decision processes. *J. Applied Probability*, pp. 794–802, 1982.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning.* MIT Press, 1998.

Tamar, A., Di Castro, D., and Mannor, S. Policy gradients with variance related risk criteria. In *ICML*, 2012.

Tesauro, G. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Chapter 4

Scaling Up Robust MDPs using Function Approximation

This chapter was published as:

A. Tamar, S. Mannor, and H. Xu. Scaling up robust MDPs using function approximation. In *Proceedings of the 31th International Conference on Machine Learning (ICML-14)*, JMLR Workshop and Conference Proceedings 32 :181–189, 2014.

Abbreviations

MDP	—	Markov decision process
ADP	—	Approximate dynamic programming
RL	—	Reinforcement learning
$RMDP$	—	Robust Markov decision process
$ARPI$	—	Approximate Robust Policy Iteration
$LSTD$	—	Least-squares temporal differences
RBF	—	Radial basis functions

Notations

$\mathcal{M}(\mathcal{B})$	—	Set of probability measures on $\mathcal{B}$
$\mathcal{X}$	—	State space
$\mathcal{Z}$	—	Set of terminal states
$\mathcal{U}$	—	Action space
r	—	Reward
P	—	Transition probability
γ	—	Discount factor
π	—	Policy
$V^{\pi,P}$	—	Value function for policy π and transitions P
$\mathcal{P}$	—	Uncertainty set
V^{π}	—	Robust value function for policy π
V^{*}	—	Optimal robust value function
$\sigma_{\mathcal{P}(x,u)}$	—	Worst-case expectation operator for state-action (x, u)
σ_{π}	—	Worst-case expectation operator for policy π
T^{π}	—	Robust Bellman operator for policy π
T	—	Robust Bellman operator
$\tilde{V}^{\pi}$	—	Approximate robust value function for policy π
$\phi(x)$	—	State-dependent features
Φ	—	Features matrix for J
P^{π}	—	Transition probability matrix under policy π
Π	—	Projection operator
d	—	Projection weights
$\hat{P}$	—	Transition probability for sampling
$\hat{\pi}$	—	Policy for sampling

$\hat{\xi}$	—	Initial state distribution for sampling
Q^{π}	—	Robust state-action value function for policy π
$\tilde{Q}^{\pi}$	—	Approximate robust state-action value function for policy π
K	—	Strike price
T	—	Maturity time

Scaling Up Robust MDPs using Function Approximation

Aviv Tamar AVIVT@TX.TECHNION.AC.IL
Electrical Engineering Department, The Technion - Israel Institute of Technology, Haifa 32000, Israel

Shie Mannor SHIE@EE.TECHNION.AC.IL
Electrical Engineering Department, The Technion - Israel Institute of Technology, Haifa 32000, Israel

Huan Xu MPEXUH@NUS.EDU.SG
Mechanical Engineering Department, National University of Singapore, Singapore 117575, Singapore

Abstract

We consider *large-scale* Markov decision processes (MDPs) with parameter uncertainty, under the robust MDP paradigm. Previous studies showed that robust MDPs, based on a minimax approach to handling uncertainty, can be solved using dynamic programming for *small to medium sized* problems. However, due to the "curse of dimensionality", MDPs that model real-life problems are typically prohibitively large for such approaches. In this work we employ a reinforcement learning approach to tackle this planning problem: we develop a *robust approximate dynamic programming* method based on a projected fixed point equation to approximately solve large scale robust MDPs. We show that the proposed method provably succeeds under certain technical conditions, and demonstrate its effectiveness through simulation of an option pricing problem. To the best of our knowledge, this is the first attempt to scale up the robust MDP paradigm.

1. Introduction

Markov decision processes (MDPs) are standard models for sequential decision making problems in stochastic dynamic environments (Puterman, 1994; Bertsekas & Tsitsiklis, 1996). Given the parameters, namely, transition probability and reward, the strategy that achieves maximal expected accumulated reward is considered optimal. However, in practice, these parameters are typically estimated from noisy data, or even worse, they may change during the execution of a policy. It is thus not surprising that the actual performance of the chosen strategy can significantly degrade from the model's prediction due to such *parameter uncertainty* – the deviation of the model parameters from the true ones (see experiments in Mannor et al. 2007).

To mitigate performance deviation due to parameter uncertainty, the robust MDP framework (Iyengar, 2005; Nilim & El Ghaoui, 2005; Bagnell et al., 2001) is now a common method. In this context, it is assumed that the *uncertain* parameters can be any member of a known set (termed the "uncertainty set"), and solutions are ranked based on their performance under the (respective) worst parameter realizations. Under mild technical conditions, the optimal solution of a robust MDP can be obtained using dynamic programming, at least for small to medium sized MDPs.

This paper considers planning in large robust MDPs, a setting largely untouched in literature. It is widely known that, due to the "curse of dimensionality", practical problems modeled as MDPs often have prohibitively large state-spaces, under which dynamic programming becomes intractable. Many approximation schemes have been proposed to alleviate the curse of dimensionality of large scale MDPs, among them approximate dynamic programming (ADP) is a popular approach (Powell, 2011). ADP considers approximations of the optimal value function, for example, as a linear functional of some features of the state, that can be solved efficiently using a sampling based approach. Of course, selecting good features is an art by itself. However, ADP has been used successfully in large-scale problems with hundreds of state dimensions (Powell, 2011). Inspired by the empirical success of ADP, we adapt it to the robust MDP setting, and develop and analyze methods that handle large scale robust MDPs. From a high level, we indeed solve a planning problem via a reinforcement learning (RL; Sutton & Barto 1998) approach: while the robust MDP model, the parameters, and the uncertainty sets are all known, and hence the optimal solution is well defined, we still use an RL approach to approximately find the solution

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

due to the scale of the problem.

Our specific contributions are a framework for approximate solution of large-scale robust MDPs; algorithms for approximate robust policy evaluation and policy improvement, with convergence proofs and error bounds; and an application of our framework to an option trading domain.

2. Background

We describe our problem formulation and some preliminaries from robust MDPs and ADP.

2.1. Robust Markov Decision Processes

For a discrete set $\mathcal{B}$, let $\mathcal{M}(\mathcal{B})$ denote the set of probability measures on $\mathcal{B}$, and let $|\mathcal{B}|$ denote its cardinality. A Markov Decision Process (MDP; Puterman 1994) is a tuple $\{\mathcal{X}, \mathcal{Z}, \mathcal{U}, P, r, \gamma\}$ where $\mathcal{X}$ is a finite set of states, $\mathcal{Z}$ is a (possibly empty) set of absorbing terminal states, and $\mathcal{U}$ is a finite set of actions. Also, $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is a deterministic and bounded reward function, γ is a discount factor, and $P : \mathcal{X} \times \mathcal{U} \to \mathcal{M}(\mathcal{X} \cup \mathcal{Z})$ denotes the probability distribution of next states, given the current state and action. We assume zero reward at terminal states.

A stationary policy $\pi : \mathcal{X} \to \mathcal{M}(\mathcal{U})$ maps each state to a probability distribution over the actions. The value of a state x under policy π and state transition model P is denoted $V^{\pi,P}(x)$ and represents the expected sum of discounted returns when starting from that state and executing π,

$$V^{\pi,P}(x) = \mathbb{E}^{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \middle| x_0 = x\right],$$

where $\mathbb{E}^{\pi,P}$ denotes expectation w.r.t. the state-action distribution induced by the transitions P and the policy π. Note that for any terminal state $z \in \mathcal{Z}$ and all π and P we have $V^{\pi,P}(z) = 0$.

Typically in MDPs, one is interested in finding a policy that maximizes the value of certain (or all) states. When the state space is small enough, and all the parameters are known, efficient methods exist (Puterman, 1994). In practice, however, the state transition probabilities may not be exactly known. A widely-applied approach in this setting is the Robust MDP (RMDP; Nilim & El Ghaoui 2005; Iyengar 2005, also termed Ambiguous MDP). In this framework, the unknown transition probabilities are assumed to lie in some *known* uncertainty set. Such a set may be obtained, for example, from statistical confidence intervals when the transition probabilities are estimated from data. Mathematically, an RMDP is a tuple $\{\mathcal{X}, \mathcal{Z}, \mathcal{U}, \mathcal{P}, r, \gamma\}$ where $\mathcal{X}, \mathcal{Z}, \mathcal{U}, r$, and γ are as defined for MDPs. The uncertainty set $\mathcal{P}$, where $\mathcal{P}(x, u) \subset \mathcal{M}(\mathcal{X} \cup \mathcal{Z})$, denotes a known uncertainty in the state transitions. Note that this definition implicitly assumes a *rectangularity* of the uncertainty set (Iyengar, 2005). In robust MDPs, one is typically interested in maximizing the *worst case* performance. Formally, we define the robust value function (Iyengar, 2005; Nilim & El Ghaoui, 2005) for a policy π as its worst-case value function

$$V^{\pi}(x) = \inf_{P \in \mathcal{P}} V^{\pi,P}(x),$$

and we seek for the optimal robust value function $V^*(x) = \sup_{\pi} \left\{\inf_{P \in \mathcal{P}} V^{\pi,P}(x)\right\}$. Iyengar (2005) and Nilim & El Ghaoui (2005) showed that similarly to the regular value function, the robust value function is obtained by a deterministic policy, and satisfies a (robust) Bellman recursion of the form

$$V^*(x) = \sup_{u \in \mathcal{U}} \left\{ r(x, u) + \gamma \inf_{P \in \mathcal{P}} \mathbb{E}^P \left[V^*(x') | x, u\right] \right\},$$

where x' denotes the state following the state x and action u. Thus, in the sequel we shall only consider deterministic policies, and write $\pi(x)$ as the action prescribed by policy π at state x.

Iyengar (2005) proposed a policy iteration algorithm for the robust MDP framework. This algorithm repeatedly improves a policy π by choosing greedy actions with respect to V^{π}. The key step in this approach is therefore policy evaluation: calculating V^{π}, which satisfies

$$V^{\pi}(x) = r(x, \pi(x)) + \gamma \inf_{P \in \mathcal{P}} \mathbb{E}^P \left[V^{\pi}(x') | x, \pi(x)\right]. \quad (1)$$

The non-linear equation (1) may be solved for V^{π} using an iterative method as follows. Let us first write (1) in vector notation. For some x and u we define the operator $\sigma_{\mathcal{P}(x,u)} : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}$ as

$$\sigma_{\mathcal{P}(x,u)} v \doteq \inf \left\{ p^{\top} v : p \in \mathcal{P}(x, u)] \right\},$$

where $v \in \mathbb{R}^{|\mathcal{X}|}$ and, slightly abusing notation, we ignore transitions to terminal states in $\mathcal{P}(x, u)$. Also, for some policy π let the operator $\sigma_{\pi} : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ be defined such that $\{\sigma_{\pi} v\}(x) \doteq \sigma_{\mathcal{P}(x,\pi(x))} v$. Then (1) may be written as $V^{\pi} = r^{\pi} + \gamma \sigma_{\pi} V^{\pi}$. Let $T^{\pi} : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ denote the robust Bellman operator for a fixed policy, defined by

$$T^{\pi} v \doteq r^{\pi} + \gamma \sigma_{\pi} v. \quad (2)$$

We see that V^{π} is a fixed point of T^{π}, i.e., $V^{\pi} = T^{\pi} V^{\pi}$. Furthermore, since T^{π} is known to be a contraction in the sup norm (Iyengar, 2005), V^{π} may be found by iteratively applying T^{π} to some vector v.

The robust Bellman operator $T : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ is defined by

$$T v(x) \doteq \sup_{\pi} T^{\pi} v(x),$$

and was shown to be a contraction (Iyengar, 2005), with V^* as its fixed point.

2.2. Projected Fixed Point Equation Methods

For MDPs, when the state space is large, dynamic programming methods become intractable, and one has to resort to an approximation procedure. A popular approach involves a projection of the value function onto a lower dimensional subspace by means of linear function approximation (Bertsekas & Tsitsiklis, 1996), and solving the solution of a *projected* Bellman equation. We briefly review this approach.

Assume a standard MDP setting without uncertainty, where the Bellman equation (1) for a fixed policy is reduced to $V^\pi(x) = r(x, \pi(x)) + \gamma \mathbb{E}^P V^\pi(x')$, and let T^π_{reg} denote the corresponding fixed policy Bellman operator. When the state space is large, calculating $V^\pi(x)$ for every x is prohibitively computationally expensive, and a lower dimensional approximation of V^π is sought. Consider the linear approximation given by a weighted sum of features

$$\tilde{V}^\pi(x) = \phi(x)^\top w, \quad x \in \mathcal{X},$$

where $\phi(x) \in \mathbb{R}^k$, $k < |\mathcal{X}|$ contains the features of state x and $w \in \mathbb{R}^k$ are the approximation weights. Let $\Phi \in \mathbb{R}^{|\mathcal{X}| \times k}$ denote a matrix with the feature vectors in its rows. We assume that the features are linearly independent, i.e., $rank(\Phi) = k$. A popular approach for finding w is by solving the *projected Bellman equation* (Bertsekas, 2012), given by

$$\tilde{V}^\pi = \Pi T^\pi_{reg} \tilde{V}^\pi, \tag{3}$$

where Π is a projection operator onto the subspace spanned by Φ with respect to a d-weighted Euclidean norm. At this point we only assume that $d \in \mathbb{R}^{|\mathcal{X}|}$ is positive. Since there is no uncertainty, T^π_{reg} is a linear mapping, and Equation (3) may be written in matrix form as follows

$$\Phi^\top D \Phi w = \Phi^\top D r + \gamma \Phi^\top D P^\pi \Phi w, \tag{4}$$

where $D = \mathrm{diag}(d)$, and $P^\pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ is the Markov transition matrix induced by policy π. Given $\Phi^\top D \Phi$, $\Phi^\top D r$, and $\Phi^\top D P^\pi \Phi$, Eq. (4) may be solved for w either by matrix inversion (Boyan, 2002), or iteratively (known as Projected Value Iteration; PVI; Bertsekas 2012)

$$w_{k+1} = \left(\Phi^\top D \Phi\right)^{-1} \left(\Phi^\top D r + \gamma \Phi^\top D P^\pi \Phi w_k\right). \tag{5}$$

When d corresponds to the steady state distribution over states for policy π, the iterative procedure in (5) can be shown to converge using contraction properties of ΠT^π_{reg} (Bertsekas, 2012). For a large state space, the terms in (5) cannot be calculated explicitly. However, the strength of this approach is that these terms may be sampled efficiently, using trajectories from the MDP (Bertsekas, 2012).

Recall that our ultimate goal is policy improvement. For a regular MDP, the policy evaluation procedure described above may be combined with a policy improvement step using Least Squares Policy Iteration (LSPI; Lagoudakis & Parr 2003), which extends policy iteration to the function approximation setting.

3. Robust Policy Evaluation

In this section we propose an extension of ADP to the robust setting. We do this as follows. First, we consider policy evaluation, and extend the projected fixed point equation (3) to the robust case, with the robust T^π operator replacing T^π_{reg}. We discuss the conditions under which this equation has a solution, and how it may be obtained. We then propose a sampling based procedure to solve the equation for large state spaces, and prove its convergence. Finally, in Section 4, we will use our policy evaluation procedure as part of a policy improvement algorithm in the spirit of LSPI (Lagoudakis & Parr, 2003), for obtaining an (approximately) optimal robust policy.

3.1. A Projected Fixed Point Equation

Throughout this section we consider a fixed policy π. For some positive d, let the projection operator Π be defined as above. Consider the following *projected robust Bellman equation* for a fixed policy

$$\tilde{V}^\pi = \Pi T^\pi \tilde{V}^\pi. \tag{6}$$

Note that here, as opposed to (3), T^π is not necessarily linear, and hence it is not clear whether Eq. (6) has a solution at all. We now show that under suitable conditions the operator ΠT^π is a contraction and Equation (6) has a *unique* solution. We consider two different cases, depending on the existence of terminal states $\mathcal{Z}$. Let $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$ represent a given policy, state transition probabilities, and initial state distribution, respectively. We let $\Pr(x_t = j | \hat{\pi}, \hat{P}, \hat{\xi})$ denote the probability that the state at time t is j, given that the states evolve according to a Markov chain with transitions $\hat{P}$, policy $\hat{\pi}$, and initial state distribution $\hat{\xi}$. In the sequel, $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$ will be used to represent the *exploration* policy of the MDP in an offline learning setting. We make the following assumption on $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$, which also defines the projection weights d.

Assumption 1. *Either $\mathcal{Z} = \emptyset$, and there exists positive numbers d_j such that*

$$d_j = \lim_{t \to \infty} \Pr(x_t = j | x_0 = i, \hat{\pi}, \hat{P}) \quad \forall i, j \in \mathcal{X},$$

or $\mathcal{Z} \neq \emptyset$, and the policy $\hat{\pi}$ is proper *(Bertsekas, 2012), that is, for $\bar{t} = |\mathcal{X}|$*

$$\Pr(x_{\bar{t}} \in \mathcal{Z} | x_0 = i, \hat{\pi}, \hat{P}) > 0 \quad \forall i \in \mathcal{X},$$

and all states have a positive probability of being visited. In this case we let

$$d_j = \sum_{t=0}^{\infty} \Pr(x_t = j | \hat{\pi}, \hat{P}, \hat{\xi}) \quad \forall j \in \mathcal{X}.$$

Put simply, Assumption 1 requires that every state has a positive probability of being visited, and defines d_j as a suitable occupation measure of state j.

The following assumption relates the transitions of the exploration policy and the (uncertain) transitions of the policy under evaluation.

Assumption 2. *There exists* $\beta \in (0,1)$ *such that* $\gamma P(x'|x,\pi(x)) \leq \beta \hat{P}(x'|x,\hat{\pi}(x)), \quad \forall P \in \mathcal{P}, x \in \mathcal{X}, x' \in \mathcal{X}$.

Assumption 2 may appear restrictive, especially when the discount factor γ approaches 1. Unfortunately, it is necessary in the sense that without it ΠT^{π} is not necessarily a contraction (see supplementary material). We note that a similar difficulty arises in off-policy RL (Bertsekas & Yu, 2009; Sutton et al., 2009), and our Assumption 2 is in fact similar to an assumption of Bertsekas & Yu 2009. Nevertheless, although our algorithms in the sequel are motivated by the contraction property of ΠT^{π}, we show empirically that our approach works in cases where Assumption 2 is *severely violated*, therefore in practice it is not a serious limitation.

Let $\|\cdot\|_d$ denote the d-weighted Euclidean norm, which is well-defined due to Assumption 1. Our key insight is the following proposition, which shows that under Assumption 2, the robust Bellman operator is a β-contraction in $\|\cdot\|_d$.

Proposition 3. *Let Assumptions 1 and 2 hold. Then* $\|T^{\pi}y - T^{\pi}z\|_d \leq \beta\|y - z\|_d$ *for all* $y, z \in \mathbb{R}^{|\mathcal{X}|}$

Proof. Fix $x \in \mathcal{X}$, and assume that $T^{\pi}y(x) \geq T^{\pi}z(x)$. Choose some $\epsilon > 0$, and $P_x \in \mathcal{P}$ such that

$$\mathbb{E}^{P_x}\left[z(x')|\, x, \pi(x)\right] \leq \inf_{P\in\mathcal{P}} \mathbb{E}^{P}\left[z(x')|\, x, \pi(x)\right] + \epsilon. \quad (7)$$

Also, note that by definition

$$\inf_{P\in\mathcal{P}} \mathbb{E}^{P}\left[y(x')|\, x, \pi(x)\right] \leq \mathbb{E}^{P_x}\left[y(x')|\, x, \pi(x)\right]. \quad (8)$$

Now, we have

$$\begin{aligned}
&0 \leq T^{\pi}y(x) - T^{\pi}z(x) \\
&\quad\leq \left(\gamma\mathbb{E}^{P_x}\left[y(x')|\, x, \pi(x)\right]\right) - \left(\gamma\mathbb{E}^{P_x}\left[z(x')|\, x, \pi(x)\right] - \gamma\epsilon\right) \\
&\quad= \gamma\mathbb{E}^{P_x}\left[y(x') - z(x')|\, x, \pi(x)\right] + \gamma\epsilon \\
&\quad\leq \beta\mathbb{E}^{\hat{P}}\left[|y(x') - z(x')|\,|\, x, \hat{\pi}(x)\right] + \gamma\epsilon,
\end{aligned}$$

where the second inequality is by (7) and (8), and the last inequality is by Assumption 2. Conversely, if $T^{\pi}z(x) \geq T^{\pi}y(x)$, following the same procedure we obtain $0 \leq T^{\pi}z(x) - T^{\pi}y(x) \leq \beta\mathbb{E}^{\hat{P}}\left[|y(x') - z(x')|\,|\, x, \hat{\pi}(x)\right] + \gamma\epsilon$, and we therefore conclude that $|T^{\pi}y(x) - T^{\pi}z(x)| \leq \beta\mathbb{E}^{\hat{P}}\left[|y(x') - z(x')|\,|\, x, \hat{\pi}(x)\right] + \gamma\epsilon$. Since ϵ was arbitrary, we have that $|T^{\pi}y(x) - T^{\pi}z(x)| \leq \beta\mathbb{E}^{\hat{P}}\left[|y(x') - z(x')|\,|\, x, \hat{\pi}(x)\right]$ for all x, and therefore

$$\|T^{\pi}y - T^{\pi}z\|_d \leq \beta\left\|\hat{P}^{\hat{\pi}}\,|y - z|\right\|_d \leq \beta\,\|y - z\|_d\,,$$

where in last equality we used the well-known result that the state transition matrix $\hat{P}^{\hat{\pi}}$ is contracting in the d-weighted Euclidean norm (Bertsekas, 2012). □

The projection operator Π is known to be non-expansive in the d-weighted norm (Bertsekas, 2012). This fact, and Lemma 6.9 of Bertsekas & Tsitsiklis (1996) lead to the following contraction property and error bound for the approximate robust value function $\tilde{V}^{\pi}$:

Corollary 4. *Let Assumptions 1 and 2 hold. Then the projected robust Bellman operator* ΠT^{π} *is a* β*-contraction in the* d*-weighted Euclidean norm. Furthermore, Eq. (6) has a unique solution, and*

$$\left\|\tilde{V}^{\pi} - V^{\pi}\right\|_d \leq \frac{1}{1-\beta}\,\|\Pi V^{\pi} - V^{\pi}\|_d\,.$$

The contraction property in Corollary 4 also suggests a straightforward procedure for solving Equation (6) which we describe next.

3.2. Robust Projected Value Iteration

Consider the robust equivalent of PVI for solving Eq. (6):

$$\Phi w_{k+1} = \Pi T^{\pi}\left(\Phi w_k\right). \quad (9)$$

The algorithm (9) may be written explicitly in matrix form (see Bertsekas 2012) as

$$w_{k+1} = \left(\Phi^{\top}D\Phi\right)^{-1}\left(\Phi^{\top}Dr + \gamma\Phi^{\top}D\sigma_{\pi}(\Phi w_k)\right). \quad (10)$$

We refer to the algorithm in (10) as *robust projected value iteration* (RPVI). Note that a matrix inversion approach would not be applicable here, as (10) is not linear due to non-linearity of $\sigma_{\pi}(\cdot)$.

Corollary 4 guarantees that under Assumptions 1 and 2, the iterates of (9) converge to the fixed point of ΠT^{π}, and the RPVI algorithm converges to the corresponding weights. We emphasize that Assumption 2 is only a *sufficient* condition for convergence. As we show empirically in Section 5, the algorithm works in cases where Assumption 2 is severely violated, and in fact, we have not encountered convergence issues in any of our experiments. Nevertheless, Assumption 2 does point out where things may go wrong. This is important in practice, especially if the uncertainty set may be controlled to satisfy it. Finally, note that for averager type function approximations (Gordon, 1995), such as non-overlapping grid tiles, kernel smoothing, and k-nearest-neighbor, Π contracts in the sup-norm.

Since T^π also contracts in the sup-norm (Iyengar, 2005), ΠT^π contracts regardless of Assumption 2, and convergence of RPVI is guaranteed.

For a large state space, computing the terms in (10) exactly is intractable. For this case we propose a sampling procedure for estimating these terms, as described next.

3.3. A Sampling Based Approach

When the state space is too large for the terms in Equation (6) to be computed exactly, one may resort to a sampling based procedure. This approach is popular in the RL and ADP literature, and has been used successfully on problems with very large state spaces (Powell, 2011). Here, we describe how it may be applied for the robust MDP setting.

Assume that we have obtained a long trajectory from an MDP with transition probabilities $\hat{P}$, while following policy π. We denote this trajectory by $x_0, u_0, r_0, x_1, u_1, r_1, \ldots, x_N, u_N, r_N$. The terms in (10) may be estimated from the data by[1]

$$\Phi^\top D\Phi \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)\phi(x_t)^\top,\ \ \Phi^\top Dr \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)r(x_t,u_t),$$

and

$$\Phi^\top D\sigma_\pi(\Phi w_k) \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)\sigma_{\mathcal{P}(x_t,u_t)}(\Phi w_k). \quad (11)$$

Using the law of large numbers, it may be proved[2] that these estimates converge with probability 1 to their respective terms in (10) as $N \to \infty$. Together with Corollary 4 we have the following convergence result. The straightforward proof is omitted.

Proposition 5. *Let Assumptions 1 and 2 hold. Consider the RPVI algorithm with the terms in (10) replaced by their sampled counterparts (11). Then as $N \to \infty$ and $k \to \infty$, w_k converges with probability 1 to w^*, and Φw^* is the unique solution of (6).*

3.4. Solving the Inner Problem

In Eq. (11), the calculation of each $\sigma_{\mathcal{P}(x_t,u_t)}(\Phi w_k)$ in the sum requires the solution of the *inner problem*:

$$\inf_{p\in\mathcal{P}(x,u)} \sum_{x\in\mathcal{X}_r(x,u)} p(x)\phi(x)^\top w_k, \quad (12)$$

where $\mathcal{X}_r(x,u)$ denotes the set of reachable states from (x,u) under *all* transitions in the set $\mathcal{P}(x,u)$. Solving Eq. (12) clearly requires a model – i.e., access to the state transitions in $\mathcal{P}(x,u)$. Also, depending on the uncertainty set, it may be computationally demanding. We now discuss *specific* uncertainty sets for which Eq. (12) is tractable.

A natural class of models is constructed from empirical state transitions $x_t \to x_{t+1}$. Let $\hat{p}$ denote the empirical transition frequencies from state x and action u (obtained by, e.g., historical observations of the system), and consider sets on the support of $\hat{p}$ of the form $\mathcal{P}(x,u) = \left\{p : \mathrm{Dist}(p,\hat{p}) \le \epsilon, p^\top \mathbb{1} = 1, p \ge 0\right\}$, where $\mathrm{Dist}(\cdot,\cdot)$ is some distance function and $\epsilon > 0$. The distance function and confidence parameter ϵ are typically related to statistical confidence regions about $\hat{p}$ (Nilim & El Ghaoui, 2005). For the case of the L_1 distance, Strehl & Littman (2005) solve Eq. (12) with complexity $\mathcal{O}(|\hat{p}|\log|\hat{p}|)$. Iyengar (2005) and Nilim & El Ghaoui (2005) propose efficient solutions for the Kullback-Liebler distance, and also for interval and ellipsoidal models. All of these methods scale at least linearly with the number of elements in $\hat{p}$, which in most practical scenarios is small compared to the cardinality of the state space, as it is bounded by the sample size used to create $\hat{p}$. In the case of binary transitions, as in our option pricing example of Section 5, performing the minimization in (12) is trivial.

Nonetheless, some problems may involve very large, or even continuous sets of reachable states. A natural model for these cases is a set of *parametric* distributions. Let $p_\theta(x)$ denote a distribution on $\mathcal{X}$ parameterized by θ. We consider uncertainty sets of the form $\mathcal{P}(x,u) = \{p_\theta : \theta \in \Theta\}$, where Θ is some convex set[3], and our goal is solving

$$\inf_{\theta\in\Theta} \mathbb{E}_{p_\theta}\left[\phi(x)^\top w_k\right]. \quad (13)$$

We assume that we have access to a distribution $\tilde{p}(x)$ such that $p_\theta(x)/\tilde{p}(x)$ is well defined for all $x \in \mathcal{X}$ and $\theta \in \Theta$. Now, observe that (13) may be written as a Stochastic Program (SP): $\inf_{\theta\in\Theta}\mathbb{E}_{\tilde{p}}\left[\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k\right]$. A standard solution to this SP is via the Sample Average Approximation (SAA; Shapiro & Nemirovski 2005), where N_s i.i.d. samples $x_i \sim \tilde{p}$ are drawn, and the following *deterministic* problem is solved: $\inf_{\theta\in\Theta}\frac{1}{N_s}\sum_{i=1}^{N_s}\frac{p_\theta(x_i)}{\tilde{p}(x_i)}\phi(x_i)^\top w_k$. When the objective of the SP is convex, and under additional technical conditions on $\tilde{p}$, p_θ, and ϕ, efficient solution of (13) is guaranteed[4] (Shapiro & Nemirovski, 2005). An alternative to the SAA is to optimize (13) directly using stochastic mirror descent (Nemirovski et al., 2009), by noting that an unbiased estimate of the gradient may be ob-

[1] These estimates are for the case $\mathcal{Z} = \emptyset$ in Assumption 1. Modifying these estimates for the case $\mathcal{Z} \neq \emptyset$ is straightforward, along the lines of Chapter 7.1 of Bertsekas (2012).

[2] The proof is similar to the case without uncertainty, detailed by Bertsekas (2012).

[3] As a concrete example, consider a Gaussian distribution $p_\theta = \mathcal{N}(\theta, 1)$, where $\Theta = [\theta^-, \theta^+]$, is a confidence interval for the maximum likelihood estimate of θ from historical data.

[4] See the supplementary material for an explicit result.

tained by sampling, using the likelihood ratio trick:

$$\nabla_\theta \mathbb{E}_{p_\theta}\left[\phi(x)^\top w_k\right] = \mathbb{E}_{p_\theta}\left[\nabla_\theta \log p_\theta(x)\phi(x)^\top w_k\right].$$

An in-depth analysis of this approach is deferred to the full version of this paper. In the supplementary material we present a successful application of our method to a domain with continuous state transitions, using the SAA method described above.

4. Robust Approximate Policy Iteration

In this section we propose a policy improvement algorithm, driven by the RPVI method of the previous section.

First, let us introduce the state-action value function $Q^\pi(x,u) = \inf_{P\in\mathcal{P}} \mathbb{E}^{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,u_t)\middle|\, x_0 = x, u_0 = u\right]$, which is more convenient for applying the optimization step of policy iteration than $V^\pi(x)$. We assume linear function approximation of the form $\tilde{Q}^\pi(x,u) = \phi(x,u)^\top w$, where $\phi(x,u)\in\mathbb{R}^k$ is a state-action feature vector and $w\in\mathbb{R}^k$ is a parameter vector. Note that $Q^\pi(x,u)$ may be seen as the value function of an equivalent RMDP with states in $\mathcal{X}\times\mathcal{U}$, therefore the policy evaluation algorithm of Section 3 applies. Also, note that given some w, a greedy policy $\pi^*_w(x)$ at state x with respect to that approximation may be computed by

$$\pi^*_w(x) = \arg\max_u \phi(x,u)^\top w, \tag{14}$$

and we write $\phi^*_w(x) = \phi(x,\pi^*_w(x))$, and let Φ^*_w denote a matrix with $\phi^*_w(x)$ in its rows.

The Approximate Robust Policy Iteration (ARPI) algorithm is initialized with an arbitrary parameter vector w_0. At iteration $i+1$, we estimate the parameter w_{i+1} of the *greedy* policy with respect to w_i as follows. We initialize $\theta_0\in\mathbb{R}^k$ to some arbitrary value, and then iterate on θ:

$$\theta_{j+1} = \left(\Phi^\top D\Phi\right)^{-1}\left(\Phi^\top Dr + \gamma\Phi^\top D\sigma_\pi(\Phi^*_{w_i}\theta_j)\right), \tag{15}$$

where the terms in (15) are estimated from data (cf. Eq. 11) according to $\Phi^\top D\Phi \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)\phi(x_t,u_t)^\top$, $\Phi^\top Dr \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)r(x_t,u_t)^\top$, and $\Phi^\top D\sigma_\pi(\Phi^*_{w_i}\theta_j) \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)\sigma_{\mathcal{P}(x_t,u_t)}(\Phi^*_{w_i}\theta_j)$. Note that, similarly to Eq. (11), each term in the last sum requires the solution of the following problem $\inf_{p\in\mathcal{P}(x,u)}\sum_{x\in\mathcal{X}_r(x,u)} p(x)\phi(x,\pi^*_{w_i}(x))^\top\theta_j$, which may be solved efficiently for the uncertainty sets discussed above. After θ has converged, we set w_{i+1} to its final value. In practice, we only iterate (15) for a few iterations[5] and set w_{i+1} to the last value of θ.

For comparison, in standard LSPI (Lagoudakis & Parr, 2003) the iteration on θ is not needed, as the policy evaluation equation (3) is linear, and may be solved using a least squares approach (LSTD; Boyan 2002). Computationally, the contraction property of Corollary 4 guarantees a linear convergence rate for the θ iteration, therefore the addition of this step should not impact performance significantly. Also, note that the computation of $\Phi^\top D\Phi$ and $\Phi^\top Dr$ only needs to be done once.

For standard approximate policy iteration, a classical result (Bertsekas, 2012) bounds the error (closeness to optimality) of the resulting policy by errors in policy evaluation and policy improvement. We now extend this result to robust approximate policy iteration.

Consider a general approximate robust policy iteration method that generates a sequence of policies $\{\pi_i\}$ and corresponding robust value functions $\{V_i\}$ that satisfy

$$\|V_i - V^{\pi_i}\|_\infty \le \delta, \quad \|T^{\pi_{i+1}}V_i - TV_i\|_\infty \le \epsilon. \tag{16}$$

The following extension of Proposition 2.5.8 of Bertsekas (2012) bounds the error $\|V^{\pi_i} - V^*\|_\infty$. The proof is based on the contraction and monotonicity properties of T^π and T, and detailed in the supplementary material.

Proposition 6. *The sequence $\{\pi_i\}$ generated by the general approximate robust policy iteration algorithm (16) satisfies*

$$\limsup_{i\to\infty}\|V^{\pi_i} - V^*\|_\infty \le \frac{\epsilon + 2\gamma\delta}{(1-\gamma)^2}.$$

Note that in the ARPI algorithm, since we are working with state-action values, and solve the maximization in (14) explicitly, there are no errors in the policy improvement step. We therefore have the following corollary

Corollary 7. *Consider the ARPI algorithm (15), and denote $Q_i(x,u) = \phi(x,u)^\top w_i$ and $\pi_i = \pi^*_{w_{i-1}}$. If the sequence of value functions satisfy $\|Q_i - Q^{\pi_i}\|_\infty \le \delta$ for all i, then $\limsup_{i\to\infty}\|Q^{\pi_i} - Q^*\|_\infty \le \frac{2\gamma\delta}{(1-\gamma)^2}$.*

Corollary 7 suggests that the ARPI algorithm is fundamentally sound. We note that more general L_2-norm bounds for approximate policy iteration were proposed by Munos (2003), and extending them to the robust case requires further work. In addition, Kaufman & Schaefer (2012) provide bounds for robust policy iteration without function approximation, but with errors in the calculation of the $\sigma_{\mathcal{P}(x,u)}$ operator.

5. Applications

In this section we discuss applications of robust ADP. We start with a discussion of optimal stopping problems. Then, we present an empirical evaluation on an option trading domain – a finite horizon continuous state space optimal stopping problem, for which an exact solution is intractable.

[5] Due to the fast convergence of (15) in practice, we didn't employ more sophisticated stopping conditions.

An optimal stopping problem is an RMDP where the only choice is when to terminate the process. Formally, the action set is binary $\mathcal{U} = \{0, 1\}$, and executing $u = 1$ from any state always transitions to a terminal state with probability 1 (and no uncertainty). Let $\hat{\pi}$ denote a policy that never chooses to terminate, i.e., $\hat{\pi}(x) = 0, \forall x$. In the supplementary material we show that if Assumption 2 is satisfied for $\pi = \hat{\pi}$, then it is immediately satisfied for all other policies. While this does not ease the conditions that Assumptions 2 places on the uncertainty set and discount factor, it simplifies the design of a suitable exploration policy.

5.1. Option Trading

In this section we apply ARPI to the problem of trading American-style options. An American-style put (call) option (Hull, 2006) is a contract which gives the owner the right, but not the obligation, to sell (buy) an asset at a specified strike price K on or before some maturity time T. Letting the state x_t represent the price of the asset at time $t \leq T$, the immediate payoff of executing a put option at that time is $g_{put}(x_t)$, where $g_{put}(x) \doteq \max(0, K - x)$, whereas for a call option we have $g_{call}(x) \doteq \max(0, x - K)$. Assuming Markov state transitions, an optimal execution policy may be found by solving a finite horizon optimal stopping problem; however, since the state space is typically continuous, an exact solution is infeasible. Even calculating the value of a given policy, an important goal by itself, is challenging. Previous studies (Tsitsiklis & Van Roy, 2001; Li et al., 2009) have proposed RL solutions for these tasks, and shown their utility. Here we extend this approach.

One challenge of option investments is that the underlying model is never truly known, but only accessed through historical data, in the form of state trajectories (e.g., stock prices over time). Catering for risk-averse traders, we plan policies based on the worst-case model that fits the data.

In the following we show that option trading may be formulated as an RMDP, and then present our results of applying the ARPI algorithm to the problem. We consider three different scenarios: a simple put option, a combination of a put and a call, and a case of model misspecification.

5.1.1. An RMDP Formulation

The option pricing problem may be formulated as an RMDP as follows. To account for the finite horizon, we include time explicitly in the state, thus, the state at time t is $\{x_t, t\}$. The action is binary, where 1 stands for executing the option and 0 for continuing to hold it. Once an option is executed, or when $t = T$, a transition to a terminal state takes place. Otherwise, the state transitions to $\{x_{t+1}, t+1\}$ where x_{t+1} is determined by a stochastic kernel $\hat{P}(x'|x, t)$. The reward for executing $u = 1$ at state x is $g(x)$ and zero otherwise. We have $g(x) = g_{put}(x)$ for a put option, $g(x) = g_{call}(x)$ for a call option, or some combination of them for a mixed investment.

Note that the state-action values for execution is known in advance, for we have $Q(\{x,t\}, u = 1) = g(x)$ by definition. Therefore, we only need to estimate the value of not exercising the option. We use linear function approximation $\tilde{Q}^{\pi}(\{x,t\}, u = 0) = \phi(\{x,t\})^\top w$, and the ARPI update equation (15) in this case may be written as $\theta_{j+1} = \left(\Phi^\top D \Phi\right)^{-1}\left(\gamma \Phi^\top D \sigma_\pi(\nu)\right)$, where $\nu(x,t)$ equals $g(x)$ if $g(x) > \phi(\{x,t\})^\top w_i$, and equals $\phi(\{x,t\})^\top \theta_j$ otherwise. As our features we chose 2-dimensional (for x and t) radial basis functions (RBF).[6]

The parameters for the experiments are provided in the supplementary material, and were chosen to balance the different factors in the problem. Most importantly, we chose $\gamma = 0.98$ and a large uncertainty set such that Assumption 2 is *severely violated*. We did not, however, encounter any convergence problems, indicating that our method works well beyond the limits of Assumption 2. The Matlab code for these results is provided in the supplementary material.

5.1.2. Trading with a Put Option

Here we consider a simple put option, where K is equal to the initial price x_0. Our price fluctuation model M follows a Bernoulli distribution[7] (Cox et al., 1979), $x_{t+1} = \begin{cases} f_u x_t, & \text{w.p. } p \\ f_d x_t, & \text{w.p. } 1-p \end{cases}$, where the up and down factors, f_u and f_d, are constant. Our empirical evaluation proceeds as follows. In each experiment, we generate N_{data} trajectories of length T from the true model M. From these trajectories we form the maximum likelihood estimate of the up probability $\hat{p}$, and the 95% confidence intervals $\hat{p}_-$ and $\hat{p}_+$ using the Clopper-Pearson method (Clopper & Pearson, 1934), which constructs our uncertain model M_{robust}. We also build a model without uncertainty $M_{nominal}$ by setting $\hat{p}_- = \hat{p}_+ = \hat{p}$. Using $\hat{p}$, we then simulate N_{sim} trajectories of length T (this corresponds to a policy that never executes the option), where $x_0 = K + \epsilon$, and ϵ is uniformly distributed in $[-\delta, \delta]$. These trajectories are used as input data for the ARPI algorithm of Section 4.

Let π_{robust} and $\pi_{nominal}$ denote the policies found by ARPI using M_{robust} and $M_{nominal}$, respectively. We evaluate the performance of π_{robust} and $\pi_{nominal}$ using N_{test}

[6] In comparison, Li et al. (2009) used Laguerre polynomials for x and several monotone functions for t. We observed significantly better performance with the RBFs. We attribute this to the non-separable (in x and t) nature of the value function, a property that is not captured by the representation of Li et al. (2009).

[7] Similar results were obtained with a geometric Brownian motion model, using the SAA method for solving the inner problem. These results are provided in the supplementary material.

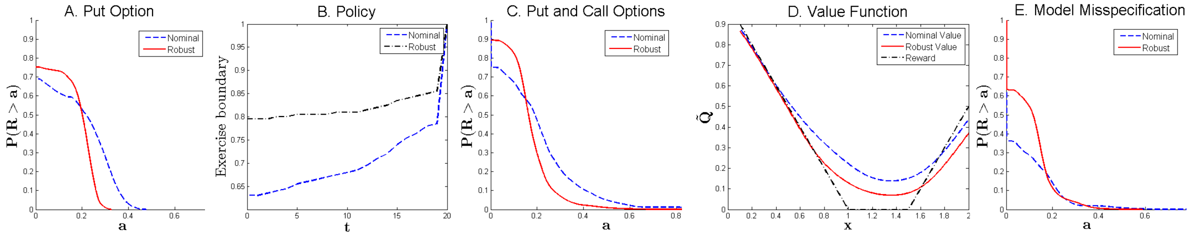


Figure 1. Performance of robust vs. nominal policies. A,C,E: The tail distribution (complementary cumulative distribution function) of the total reward R for the put option (A), put and call (C) and model misspecification (E) scenarios. Note that a higher value for some a indicates a higher chance of guaranteeing a reward of at least a, therefore the plots (A) and (C) display a risk-sensitive behavior of the robust policies. The results were obtained from 100 independent experiments. B: The nominal and robust policies for the put option scenario, represented by the exercise boundary for each t. D: The reward $g(x)$ and value function $\tilde{Q}(x, t = 5)$ from a typical experiment of the put and call option scenario.

trajectories obtained from the *true* model M. Recall that we seek risk-averse policies; thus, the advantage of π_{robust} should reflect in the least favorable outcomes. In Figure 1A we plot the tail distribution of the total reward R (from 100 experiments) obtained by π_{robust} and $\pi_{nominal}$. It may be seen that π_{robust} has a lower probability of obtaining a low payoff (or losing the investment). This, however, comes at a cost of a smaller probability for a high payoff. To the risk-sensitive investor, such results are important. In Figure 1B we further illustrate the policies π_{robust} and $\pi_{nominal}$ by plotting the exercise boundary (the lowest price for which the policy decides to exercise) for each t. The conservative behavior of π_{robust} is evident.

5.1.3. Trading with a Put and a Call

We now consider a more complicated scenario, where the trader has bought both a put option, with strike price $K_{put} < x_0$, and a call option, with strike $K_{call} > x_0$. The reward is given by $g(x) = g_{put}(x) + g_{call}(x)$, and the models and experimental procedure are the same as in the previous scenario. In Figure 1C we plot the tail distribution of the total reward (from 100 independent experiments) obtained by π_{robust} and $\pi_{nominal}$. Notice that the risk-averse policy has a significantly smaller chance of losing the investment. In Figure 1D we display the reward $g(x)$ and the (approximate) value functions $\tilde{Q}^{\pi_{robust}}$ and $\tilde{Q}^{\pi_{nominal}}$ from a typical experiment, for $t = 5$. The robust value function is important by itself, as it holds valuable information about the expected future profit.

5.1.4. Robustness to Model Misspecification

In the previous scenarios we assumed that our estimated models, M_{robust} and $M_{nominal}$, are the same as the true model M. In practice, this is rarely the case, and one has to consider the possibility of model misspecification. An RMDP model provides some robustness against model misspecification, as we now demonstrate. Let the probability p in the *true* model M depend on the state according to $p(x) = p_1 \mathbb{1}\{x \le \alpha\} + p_2 \mathbb{1}\{x > \alpha\}$, where the threshold α is $(K_{put} + K_{call})/2$. However, let the estimated models M_{robust} and $M_{nominal}$, and the experimental procedure remain as before. We consider again the case of both a put and a call option, as in Section 5.1.3. In Figure 1E we plot the tail distribution of the total reward (from 100 independent experiments) obtained by π_{robust} and $\pi_{nominal}$. Observe that in this case, the misspecification of the nominal model led to a policy that is dominated by the robust policy, which was less affected by this problem.

6. Conclusion and Future Work

We presented a novel framework for solving *large-scale* uncertain Markov decision processes. To the best of our knowledge, such problems are beyond the capabilities of previous studies, which focused on exact solutions and hence suffer from the "curse of dimensionality". We presented both formal guarantees and empirical evidence to the usefulness of our approach. As we demonstrated, uncertain MDPs are suitable for both risk-averseness and mitigation of model misspecification, indicating their importance for decision making under uncertainty.

Interestingly, as was recognized by Iyengar (2005), results on robust MDPs may also be extended to their 'best-case' counterpart, known as optimistic MDPs[8]. Such are useful for efficient exploration, as in the UCRL2 algorithm (Jaksch et al., 2010), suggesting a future extension of our work.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement n. 306638. H. Xu is partially supported by the Ministry of Education of Singapore through AcRF Tier Two grant R-265-000-443-112.

[8] See the supplementary material for more details.

References

Bagnell, A., Ng, A., and Schneider, J. Solving uncertain Markov decision problems. Technical Report CMU-RI-TR-01-25, Carnegie Mellon University, August 2001.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, fourth edition, 2012.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Bertsekas, D. P. and Yu, H. Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227(1):2750, 2009.

Boyan, J. A. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Cox, J. C., Ross, S. A., and Rubinstein, M. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.

Gordon, G. J. Stable function approximation in dynamic programming. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

Hull, J. C. *Options, Futures, and Other Derivatives (6th edition)*. Prentice Hall, 2006.

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Kaufman, D. L. and Schaefer, A. J. Robust modified policy iteration. *INFORMS Journal on Computing*, 2012.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4: 1107–1149, 2003.

Li, Y., Szepesvari, C., and Schuurmans, D. Learning exercise policies for American options. In *Proc. of the 12th International Conference on Artificial Intelligence and Statistics, JMLR: W&CP*, volume 5, pp. 352–359, 2009.

Mannor, S., Simester, D., Sun, P., and Tsitsiklis, J. N. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Munos, R. Error bounds for approximate policy iteration. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 560–567, 2003.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.

Nilim, A. and El Ghaoui, L. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Powell, W. B. *Approximate Dynamic Programming*. John Wiley and Sons, 2011.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.

Shapiro, A. and Nemirovski, A. On complexity of stochastic programming problems. In *Continuous optimization*, pp. 111–146. Springer, 2005.

Strehl, A. L. and Littman, M. L. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*, pp. 856–863. ACM, 2005.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvari, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

Tsitsiklis, J. N. and Van Roy, B. Regression methods for pricing complex American-style options. *Neural Networks, IEEE Transactions on*, 12(4):694–703, 2001.

Chapter 5

Optimizing the CVaR via Sampling

This chapter was published as:

A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, pages 2993–2999. AAAI Press, 2015.

Abbreviations

VaR	—	Value-at-risk
$CVaR$	—	Conditional value-at-risk
LR	—	Likelihood ratio
RL	—	Reinforcement learning
MC	—	Monte carlo
CDF	—	Cumulative distribution function
$GCVaR$	—	CVaR gradient estimation algorithm
SGD	—	Stochastic gradient descent
$CVaRSGD$	—	CVaR Stochastic gradient descent algorithm
MDP	—	Markov decision process
IS	—	Importance sampling

Notations

Φ	—	CVaR
ν_α	—	An α-quantile
θ	—	Tunable parameters
F	—	Cumulative distribution function
f	—	Probability density function
$\mathbf{X}$	—	Continuous random variable
Y	—	Discrete random variable
r	—	Reward function
R	—	Reward random variable
$\mathcal{D}_{y;\theta}$	—	Level set
ν	—	Empirical quantile
$\hat{F}$	—	Empirical CDF
$\Delta_{j;N}$	—	MC gradient estimate
Γ	—	Projection
$\hat{\Gamma}_\theta$	—	Infinitesimal projection
κ	—	Asymptotically stable equilibria
ϵ	—	Step size

Optimizing the CVaR via Sampling

Aviv Tamar, Yonatan Glassner, and **Shie Mannor**
Electrical Engineering Department
The Technion - Israel Institute of Technology
Haifa, Israel 32000
{avivt, yglasner}@tx.technion.ac.il, shie@ee.technion.ac.il

Abstract

Conditional Value at Risk (CVaR) is a prominent risk measure that is being used extensively in various domains. We develop a new formula for the gradient of the CVaR in the form of a conditional expectation. Based on this formula, we propose a novel sampling-based estimator for the gradient of the CVaR, in the spirit of the likelihood-ratio method. We analyze the bias of the estimator, and prove the convergence of a corresponding stochastic gradient descent algorithm to a local CVaR optimum. Our method allows to consider CVaR optimization in new domains. As an example, we consider a reinforcement learning application, and learn a risk-sensitive controller for the game of Tetris.

1 Introduction

Conditional Value at Risk (CVaR; Rockafellar and Uryasev, 2000) is an established risk measure that has found extensive use in finance among other fields. For a random payoff R, whose distribution is parameterized by a controllable parameter θ, the α-CVaR is defined as the expected payoff over the $\alpha\%$ worst outcomes of Z:

$$\Phi(\theta) = \mathbb{E}^{\theta}\left[R \middle| R \leq \nu_{\alpha}(\theta)\right],$$

where $\nu_{\alpha}(\theta)$ is the α-quantile of R. CVaR optimization aims to find a parameter θ that maximizes $\Phi(\theta)$.

When the payoff is of the structure $R = f_{\theta}(X)$, where f_{θ} is a deterministic function, and X is random but does not depend on θ, CVaR optimization may be formulated as a stochastic program, and solved using various approaches (Rockafellar and Uryasev 2000; Hong and Liu 2009; Iyengar and Ma 2013). Such a payoff structure is appropriate for certain domains, such as portfolio optimization, in which the investment strategy generally does not affect the asset prices. However, in many important domains, for example queueing systems, resource allocation, and reinforcement learning, the tunable parameters also control the *distribution* of the random outcomes. Since existing CVaR optimization methods are not suitable for such cases, and due to increased interest in risk-sensitive optimization recently in these domains (Tamar, Di Castro, and Mannor 2012; Prashanth and Ghavamzadeh 2013), there is a strong incentive to develop more general CVaR optimization algorithms.

In this work, we propose a CVaR optimization approach that is applicable when θ also controls the distribution of X. The basis of our approach is a new formula that we derive for the CVaR gradient $\frac{\partial \Phi(\theta)}{\partial \theta}$ in the form of a conditional expectation. Based on this formula, we propose a sampling-based estimator for the CVaR gradient, and use it to optimize the CVaR by stochastic gradient descent.

In addition, we analyze the bias of our estimator, and use the result to prove convergence of the stochastic gradient descent algorithm to a local CVaR optimum. Our method allows us to consider CVaR optimization in new domains. As an example, we consider a reinforcement learning application, and learn a risk-sensitive controller for the game of Tetris. To our knowledge, CVaR optimization for such a domain is beyond the reach of existing approaches. Considering Tetris also allows us to easily interpret our results, and show that we indeed learn sensible policies.

We remark that in certain domains, CVaR is often not maximized directly, but used as a constraint in an optimization problem of the form $\max_{\theta} \mathbb{E}^{\theta}[R]$ s.t. $\Phi(\theta) \geq b$. Extending our approach to such problems is straightforward, using standard penalty method techniques (see, e.g., Tamar, Di Castro, and Mannor, 2012, and Prashanth and Ghavamzadeh, 2013, for a such an approach with a variance-constrained objective), since the key component for these methods is the CVaR gradient estimator we provide here. Another appealing property of our estimator is that it naturally incorporates importance sampling, which is important when α is small, and the CVaR captures *rare* events.

Related Work Our approach is similar in spirit to the *likelihood-ratio* method (LR; Glynn, 1990), that estimates the gradient of the *expected* payoff. The LR method has been successfully applied in diverse domains such as queueing systems, inventory management, and financial engineering (Fu 2006), and also in reinforcement learning (RL; Sutton and Barto, 1998), where it is commonly known as the *policy gradient* method (Baxter and Bartlett 2001; Peters and Schaal 2008). Our work extends the LR method to estimating the gradient of the CVaR of the payoff.

Closely related to our work are the studies of Hong and Liu (2009) and Scaillet (2004), who proposed perturbation analysis style estimators for the gradient of the CVaR, for the setting mentioned above, in which θ *does not affect* the distribution of X. Indeed, their gradient formulae are different than ours, and do not apply in our setting.

LR gradient estimators for other risk measures have been proposed by Borkar (2001) for exponential utility functions, and by Tamar, Di Castro, and Mannor (2012) for mean–variance. These measures, however, consider a very different notion of risk than the CVaR. For example, the mean–variance measure is known to underestimate the risk of rare, but catastrophic events (Agarwal and Naik 2004).

Risk-sensitive optimization in RL is receiving increased interest recently. A mean-variance criterion was considered by Tamar, Di Castro, and Mannor (2012) and Prashanth and Ghavamzadeh (2013). Morimura et al. (2010) consider the expected return, with a CVaR based risk-sensitive policy for guiding the exploration while learning. Their method, however, does not scale to large problems. Borkar and Jain (2014) optimize a CVaR constrained objective using dynamic programming, by augmenting the state space with the accumulated reward. As such, that method is only suitable for a finite horizon and a small state-space, and *does not scale-up* to problems such as the Tetris domain we consider. A function approximation extension of (Borkar and Jain 2014) is mentioned, using a three time scales stochastic approximation algorithm. In that work, three different learning rates are decreased to 0, and convergence is determined by the slowest one, leading to an overall slow convergence. In contrast, our approach requires only a single learning rate. Recently, Prashanth (2014) used our gradient formula of Proposition 2 (from a preliminary version of this paper) in a two time-scale stochastic approximation scheme to show convergence of CVaR optimization. Besides providing the theoretical basis for that work, our current convergence result (Theorem 5) obviates the need for the extra time-scale, and results in a simpler and faster algorithm.

2 A CVaR Gradient Formula

In this section we present a new LR-style formula for the gradient of the CVaR. This gradient will be used in subsequent sections to optimize the CVaR with respect to some parametric family. We start with a formal definition of the CVaR, and then present a CVaR gradient formula for 1-dimensional random variables. We then extend our result to the multi-dimensional case.

Let Z denote a random variable with a cumulative distribution function (C.D.F.) $F_Z(z) = \Pr(Z \leq z)$. For convenience, we assume that Z is a continuous random variable, meaning that $F_Z(z)$ is everywhere continuous. We also assume that Z is bounded. Given a confidence level $\alpha \in (0,1)$, the α-Value-at-Risk, (VaR; or α-quantile) of Z is denoted $\nu_\alpha(Z)$, and given by

$$\nu_\alpha(Z) = F_Z^{-1}(\alpha) \doteq \inf\left\{z : F_Z(z) \geq \alpha\right\}. \qquad (1)$$

The α-Conditional-Value-at-Risk of Z is denoted by $\Phi_\alpha(Z)$ and defined as the expectation of the α fraction of the worst outcomes of Z

$$\Phi_\alpha(Z) = \mathbb{E}\left[Z \middle| Z \leq \nu_\alpha(Z)\right]. \qquad (2)$$

We next present a formula for the sensitivity of $\Phi_\alpha(Z)$ to changes in $F_Z(z)$.

2.1 CVaR Gradient of a 1-Dimensional Variable

Consider again a random variable Z, but now let its probability density function (P.D.F.) $f_Z(z;\theta)$ be parameterized by a vector $\theta \in \mathbb{R}^k$. We let $\nu_\alpha(Z;\theta)$ and $\Phi_\alpha(Z;\theta)$ denote the VaR and CVaR of Z as defined in Eq. (1) and (2), when the parameter is θ, respectively.

We are interested in the sensitivity of the CVaR to the parameter vector, as expressed by the gradient $\frac{\partial}{\partial\theta_j}\Phi_\alpha(Z;\theta)$. In all but the most simple cases, calculating the gradient analytically is intractable. Therefore, we derive a formula in which $\frac{\partial}{\partial\theta_j}\Phi_\alpha(Z;\theta)$ is expressed as a conditional expectation, and use it to calculate the gradient by *sampling*. For technical convenience, we make the following assumption:

Assumption 1. *Z is a continuous random variable, and bounded in $[-b,b]$ for all θ.*

We also make the following smoothness assumption on $\nu_\alpha(Z;\theta)$ and $\Phi_\alpha(Z;\theta)$

Assumption 2. *For all θ and $1 \leq j \leq k$, the gradients $\frac{\partial\nu_\alpha(Z;\theta)}{\partial\theta_j}$ and $\frac{\partial\Phi_\alpha(Z;\theta)}{\partial\theta_j}$ exist and are bounded.*

Note that since Z is continuous, Assumption 2 is satisfied whenever $\frac{\partial}{\partial\theta_j}f_Z(z;\theta)$ is bounded. Relaxing Assumptions 1 and 2 is possible, but involves technical details that would complicate the presentation, and is left to future work. The next assumption is standard in LR gradient estimates

Assumption 3. *For all θ, z, and $1 \leq j \leq k$, we have that $\frac{\partial f_Z(z;\theta)}{\partial\theta_j}/f_Z(z;\theta)$ exists and is bounded.*

In the next proposition we present a LR-style sensitivity formula for $\Phi_\alpha(Z;\theta)$, in which the gradient is expressed as a conditional expectation. In Section 3 we shall use this formula to suggest a sampling algorithm for the gradient.

Proposition 1. *Let Assumptions 1, 2, and 3 hold. Then*

$$\frac{\partial\Phi_\alpha(Z;\theta)}{\partial\theta_j} = \mathbb{E}^\theta\left[\frac{\partial \log f_Z(Z;\theta)}{\partial\theta_j}(Z - \nu_\alpha(Z;\theta))\middle| Z \leq \nu_\alpha(Z;\theta)\right].$$

Proof. Define the level-set $D_\theta = \left\{z \in [-b,b] : z \leq \nu_\alpha(Z;\theta)\right\}$. By definition, $D_\theta \equiv [-b, \nu_\alpha(Z;\theta)]$, and $\int_{z\in D_\theta} f_Z(z;\theta)\,dz = \alpha$. Taking a derivative and using the Leibniz rule we obtain

$$\begin{aligned} 0 &= \frac{\partial}{\partial\theta_j}\int_{-b}^{\nu_\alpha(Z;\theta)} f_Z(z;\theta)\,dz \qquad (3)\\ &= \int_{-b}^{\nu_\alpha(Z;\theta)} \frac{\partial f_Z(z;\theta)}{\partial\theta_j}dz + \frac{\partial\nu_\alpha(Z;\theta)}{\partial\theta_j} f_Z(\nu_\alpha(Z;\theta);\theta). \end{aligned}$$

By definition (2) we have $\Phi_\alpha(Z;\theta) = \int_{z\in D_\theta}\frac{f_Z(z;\theta)z}{\alpha}dz = \alpha^{-1}\int_{-b}^{\nu_\alpha(Z;\theta)} f_Z(z;\theta)\,zdz$. Now, taking a derivative and using the Leibniz rule we obtain

$$\begin{aligned} \frac{\partial}{\partial\theta_j}\Phi_\alpha(Z;\theta) =& \alpha^{-1}\int_{-b}^{\nu_\alpha(Z;\theta)}\frac{\partial f_Z(z;\theta)}{\partial\theta_j}zdz \qquad (4)\\ &+\alpha^{-1}\frac{\partial\nu_\alpha(Z;\theta)}{\partial\theta_j}f_Z(\nu_\alpha(Z;\theta);\theta)\,\nu_\alpha(Z;\theta). \end{aligned}$$

Rearranging, and plugging (3) in (4) we obtain $\frac{\partial}{\partial\theta_j}\Phi_\alpha(Z;\theta) = \alpha^{-1}\int_{-b}^{\nu_\alpha(Z;\theta)} \frac{\partial f_Z(z;\theta)}{\partial\theta_j}\left(z-\nu_\alpha(Z;\theta)\right)dz$. Finally, using the likelihood ratio trick – multiplying and dividing by $f_Z\left(z;\theta\right)$ inside the integral, which is justified due to Assumption 3, we obtain the required expectation. □

Let us contrast the CVaR LR formula of Proposition 1 with the standard LR formula for the expectation (Glynn 1990) $\frac{\partial}{\partial\theta_j}\mathbb{E}^\theta[Z] = \mathbb{E}^\theta\left[\frac{\partial \log f_Z(Z;\theta)}{\partial\theta_j}(Z-b)\right]$, where the baseline b could be any arbitrary constant. Note that in the CVaR case the baseline is *specific*, and, as seen in the proof, accounts for the sensitivity of the level-set D_θ. Quite surprisingly, this specific baseline turns out to be exactly the VaR, $\nu_\alpha(Z;\theta)$, which, as we shall see later, also leads to an elegant sampling based estimator.

In a typical application, Z would correspond to the performance of some system, such as the profit in portfolio optimization, or the total reward in RL. Note that in order to use Proposition 1 in a gradient estimation algorithm, one needs access to $\frac{\partial}{\partial\theta_j}\log f_Z(Z;\theta)$: the sensitivity of the performance distribution to the parameters. Typically, the system performance is a complicated function of a high-dimensional random variable. For example, in RL and queueing systems, the performance is a function of a trajectory from a stochastic dynamical system, and calculating its probability distribution is usually intractable. The sensitivity of the trajectory distribution to the parameters, however, is often easy to calculate, since the parameters typically control how the trajectory is generated. We shall now generalize Proposition 1 to such cases. The utility of this generalization is further exemplified in Section 5, for the RL domain.

2.2 CVaR Gradient Formula – General Case

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ denote an $n-$dimensional random variable with a finite support $[-b,b]^n$, and let Y denote a discrete random variable taking values in some countable set $\mathcal{Y}$. Let $f_Y(y;\theta)$ denote the probability mass function of Y, and let $f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)$ denote the probability density function of $\mathbf{X}$ given Y. Let the reward function r be a bounded mapping from $[-b,b]^n \times \mathcal{Y}$ to $\mathbb{R}$, and consider the random variable $R \doteq r(\mathbf{X}, Y)$. We are interested in a formula for $\frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$.

We make the following assumption, similar to Assumptions 1, 2, and 3.

Assumption 4. *The reward R is a continuous random variable for all θ. Furthermore, for all θ and $1 \le j \le k$, the gradients $\frac{\partial}{\partial\theta_j}\nu_\alpha(R;\theta)$ and $\frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$ are well defined and bounded. In addition $\frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)}{\partial\theta_j}$ and $\frac{\partial \log f_Y(y;\theta)}{\partial\theta_j}$ exist and are bounded for all $\mathbf{x}$, y, and θ.*

Define the level-set $\mathcal{D}_{y;\theta} = \left\{\mathbf{x} \in [-b,b]^n : r(\mathbf{x},y) \le \nu_\alpha(R;\theta)\right\}$. We require some smoothness of the function r, that is captured by the following assumption on $\mathcal{D}_{y;\theta}$.

Assumption 5. *For all y and θ, the set $\mathcal{D}_{y;\theta}$ may be written as a finite sum of $L_{y;\theta}$ disjoint, closed, and connected components $D^i_{y;\theta}$, each with positive measure: $\mathcal{D}_{y;\theta} = \sum_{i=1}^{L_{y;\theta}} D^i_{y;\theta}$.*

Assumption 5 may satisfied, for example, when $r(\mathbf{x},y)$ is Lipschitz in $\mathbf{x}$ for all $y \in \mathcal{Y}$. We now present a sensitivity formula for $\Phi_\alpha(R;\theta)$.

Proposition 2. *Let Assumption 4 and 5 hold. Then*

$$\frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta) = \mathbb{E}^\theta\left[\left(\frac{\partial \log f_Y(Y;\theta)}{\partial\theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{X}|Y;\theta)}{\partial\theta_j}\right)\left(R - \nu_\alpha(R;\theta)\right)\middle| R \le \nu_\alpha(R;\theta)\right].$$

The proof of Proposition 2 is similar in spirit to the proof of Proposition 1, but involves some additional difficulties of applying the Leibnitz rule in a multidimensional setting. It is given in (Tamar, Glassner, and Mannor 2014). We reiterate that relaxing Assumptions 4 and 5 is possible, but is technically involved, and left for future work. In the next section we show that the formula in Proposition 2 leads to an effective algorithm for estimating $\frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$ by sampling.

3 A CVaR Gradient Estimation Algorithm

The sensitivity formula in Proposition 2 suggests a natural Monte–Carlo (MC) estimation algorithm. The method, which we label GCVaR (Gradient estimator for CVaR), is described as follows. Let $\mathbf{x}_1, y_1 \ldots, \mathbf{x}_N, y_N$ be N samples drawn i.i.d. from $f_{\mathbf{X},Y}(\mathbf{x},y;\theta)$, the joint distribution of $\mathbf{X}$ and Y. We first estimate $\nu_\alpha(R;\theta)$ using the empirical α-quantile[1] $\tilde{v}$

$$\tilde{v} = \inf_z \hat{F}(z) \ge \alpha, \tag{5}$$

where $\hat{F}(z)$ is the empirical C.D.F. of R: $\hat{F}(z) \doteq \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{r(\mathbf{x}_i,y_i) \le z}$. The MC estimate of the gradient $\Delta_{j;N} \approx \frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$ is given by

$$\Delta_{j;N} = \frac{1}{\alpha N}\sum_{i=1}^N\left(\frac{\partial \log f_Y(y_i;\theta)}{\partial\theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}_i|y_i;\theta)}{\partial\theta_j}\right)\times \times\left(r(\mathbf{x}_i,y_i) - \tilde{v}\right)\mathbf{1}_{r(\mathbf{x}_i,y_i) \le \tilde{v}}. \tag{6}$$

It is known that the empirical α-quantile is a biased estimator of $\nu_\alpha(R;\theta)$. Therefore, $\Delta_{j;N}$ is also a biased estimator of $\frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$. In the following we analyze and bound this bias. We first show that $\Delta_{j;N}$ is a consistent estimator. The proof is similar to the proof of Theorem 4.1 in (Hong and Liu 2009), and given in (Tamar, Glassner, and Mannor 2014).

Theorem 3. *Let Assumption 4 and 5 hold. Then $\Delta_{j;N} \to \frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta)$ w.p. 1 as $N \to \infty$.*

With an additional smoothness assumption we can explicitly bound the bias. Let $f_R(\cdot;\theta)$ denote the P.D.F. of R, and define the function $g(\beta;\theta) \doteq \mathbb{E}^\theta\left[\left(\frac{\partial \log f_Y(Y;\theta)}{\partial\theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{X}|Y;\theta)}{\partial\theta_j}\right)(R - \nu_\alpha(R;\theta))\middle| R = \beta\right]$.

[1] Algorithmically, this is equivalent to first sorting the $r(\mathbf{x}_i, y_i)$'s in ascending order, and then selecting $\tilde{v}$ as the $\lceil \alpha N \rceil$ term in the sorted list.

Algorithm 1 GCVaR

```
Given:
  • CVaR level $\alpha$
  • A reward function $r(\mathbf{x}, y) : \mathbb{R}^n \times \mathcal{Y} \to \mathbb{R}$
  • Derivatives $\frac{\partial}{\partial \theta_j}$ of the probability mass function $f_Y(y;\theta)$
    and probability density function $f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)$
  • An i.i.d. sequence $\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N \sim f_{\mathbf{X},Y}(\mathbf{x}, y;\theta)$.
Set $r_1^s, \ldots, r_N^s = \text{Sort}\left(r(\mathbf{x}_1, y_1), \ldots, r(\mathbf{x}_N, y_N)\right)$
Set $\tilde{v} = r^s_{\lceil \alpha N \rceil}$
For $j = 1, \ldots, k$ do
    $\Delta_{j;N} = \frac{1}{\alpha N}\sum_{i=1}^{N}\left(\frac{\partial \log f_Y(y_i;\theta)}{\partial \theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}_i|y_i;\theta)}{\partial \theta_j}\right)\times$
        $\times \left(r(\mathbf{x}_i, y_i) - \tilde{v}\right)\mathbf{1}_{r(\mathbf{x}_i, y_i)\le \tilde{v}}$
Return: $\Delta_{1;N}, \ldots, \Delta_{k;N}$
```

Assumption 6. *For all θ, $f_R(\cdot;\theta)$ and $g(\cdot;\theta)$ are continuous at $\nu_\alpha(R;\theta)$, and $f_R(\nu_\alpha(R;\theta);\theta) > 0$.*

Assumption 6 is similar to Assumption 4 of (Hong and Liu 2009), and may be satisfied, for example, when $\frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)}{\partial \theta_j}$ is continuous and $r(\mathbf{x}, y)$ is Lipschitz in $\mathbf{x}$. The next theorem shows that the bias is $\mathcal{O}(N^{-1/2})$. The proof, given in (Tamar, Glassner, and Mannor 2014), is based on separating the bias to a term that is bounded using a result of Hong and Liu (2009), and an additional term that is bounded using well-known results for the bias of empirical quantiles.

Theorem 4. *Let Assumptions 4, 5, and 6 hold. Then $\mathbb{E}\left[\Delta_{j;N}\right] - \frac{\partial}{\partial \theta_j}\Phi_\alpha(R;\theta)$ is $O(N^{-1/2})$.*

At this point, let us again contrast GCVaR with the standard LR method. One may naively presume that applying a standard LR gradient estimator to the $\alpha\%$ worst samples would work as a CVaR gradient estimator. This corresponds to applying the GCVaR algorithm without subtracting the $\tilde{v}$ baseline from the reward in (6). Theorems 3 and 4 show that such an estimator *would not be consistent*. In fact, in (Tamar, Glassner, and Mannor 2014) we give an example where the gradient error of such an approach may be arbitrarily large.

In the sequel, we use GCVaR as part of a stochastic gradient descent algorithm for CVaR optimization. An asymptotically decreasing gradient bias, as may be established from Theorem 3, is necessary to guarantee convergence of such a procedure. Furthermore, the bound of Theorem 4 will allow us to quantify how many samples are needed at each iteration for such convergence to hold.

Variance Reduction by Importance Sampling

For very low quantiles, i.e., α close to 0, the GCVaR estimator would suffer from a high variance, since the averaging is effectively only over αN samples. This is a well-known issue in sampling based approaches to VaR and CVaR estimation, and is often mitigated using variance reduction techniques such as Importance Sampling (IS; Rubinstein and Kroese, 2011; Bardou, Frikha, and Pagès, 2009). In IS, the variance of a MC estimator is reduced by using samples from a *different* sampling distribution, and suitably modifying the estimator to keep it unbiased. It is straightforward to incorporate IS into LR gradient estimators in general, and to our GCVaR estimator in particular. Due to space constraints, and since this is fairly standard textbook material (e.g., Rubinstein and Kroese, 2011), we provide the full technical details in (Tamar, Glassner, and Mannor 2014). In our experiments we show that IS indeed improves performance significantly.

4 CVaR Optimization

In this section, we consider the setting of Section 2.2, and aim to solve the CVaR optimization problem:

$$\max_{\theta \in \mathbb{R}^k} \Phi_\alpha(R;\theta). \tag{7}$$

For this goal we propose CVaRSGD: a stochastic gradient descent algorithm, based on the GCVaR gradient estimator. We now describe the CVaRSGD algorithm in detail, and show that it converges to a local optimum of (7).

In CVaRSGD, we start with an arbitrary initial parameter $\theta^0 \in \mathbb{R}^k$. The algorithm proceeds iteratively as follows. At each iteration i of the algorithm, we first sample n_i i.i.d. realizations $x_1, y_1, \ldots, x_{n_i}, y_{n_i}$ of the random variables $\mathbf{X}$ and Y, from the distribution $f_{\mathbf{X},Y}(\mathbf{x}, y;\theta^i)$. We then apply the GCVaR algorithm to obtain an estimate $\Delta_{j;n_i}$ of $\frac{\partial}{\partial \theta_j}\Phi_\alpha(R;\theta^i)$, using the samples $x_1, y_1, \ldots, x_{n_i}, y_{n_i}$. Finally, we update the parameter according to

$$\theta_j^{i+1} = \Gamma\left(\theta_j^i + \epsilon_i \Delta_{j;n_i}\right), \tag{8}$$

where ϵ_i is a positive step size, and $\Gamma : \mathbb{R}^k \to \mathbb{R}^k$ is a projection to some compact set Θ with a smooth boundary. The purpose of the projection is to facilitate convergence of the algorithm, by guaranteeing that the iterates remain bounded (this is a common stochastic approximation technique; Kushner and Yin, 2003). In practice, if Θ is chosen large enough so that it contains the local optima of $\Phi_\alpha(R;\theta)$, the projection would rarely occur, and would have a negligible effect on the algorithm. Let $\hat{\Gamma}_\theta(\nu) \doteq \lim_{\delta\to 0}\frac{\Gamma(\theta+\delta\nu)-\theta}{\delta}$ denote an operator that, given a direction of change ν to the parameter θ, returns a modified direction that keeps θ within Θ. Consider the following ordinary differential equation:

$$\dot{\theta} = \hat{\Gamma}_\theta\left(\nabla \Phi_\alpha(R;\theta)\right), \quad \theta(0) \in \Theta. \tag{9}$$

Let $\mathcal{K}$ denote the set of all asymptotically stable equilibria of (9). The next theorem shows that under suitable technical conditions, the CVaRSGD algorithm converges to $\mathcal{K}$ almost surely. The theorem is a direct application of Theorem 5.2.1 of Kushner and Yin (2003), and given here without proof.

Theorem 5. *Consider the CVaRSGD algorithm (8). Let Assumptions 4, 5, and 6 hold, and assume that $\Phi_\alpha(R;\theta)$ is continuously differentiable in θ. Also, assume that $\sum_{i=1}^{\infty}\epsilon_i = \infty$, $\sum_{i=1}^{\infty}\epsilon_i^2 < \infty$, and that $\sum_{i=1}^{\infty}\epsilon_i\left|\mathbb{E}\left[\Delta_{j;n_i}\right] - \frac{\partial}{\partial \theta_j}\Phi_\alpha(R;\theta^i)\right| < \infty$ w.p. 1 for all j. Then $\theta^i \to \mathcal{K}$ almost surely.*

Note that from the discussion in Section 3, the requirement $\sum_{i=1}^{\infty} \epsilon_i \left| \mathbb{E}\left[\Delta_{j;n_i}\right] - \frac{\partial}{\partial \theta_j} \Phi_\alpha(R;\theta^i) \right| < \infty$ implies that we must have $\lim_{i \to \infty} n_i = \infty$. However, the rate of n_i could be very slow, for example, using the bound of Theorem 4 the requirement may be satisfied by choosing $\epsilon_i = 1/i$ and $n_i = (\log i)^4$.

5 Application to Reinforcement Learning

In this section we show that the `CVaRSGD` algorithm may be used in an RL policy-gradient type scheme, for optimizing performance criteria that involve the CVaR of the total return. We first describe some preliminaries and our RL setting, and then describe our algorithm.

We consider an episodic[2] Markov Decision Problem (MDP) in discrete time with a finite state space $\mathcal{S}$ and a finite action space $\mathcal{A}$. At time $t \in \{0, 1, 2, \dots\}$ the state is s_t, and an action a_t is chosen according to a parameterized policy π_θ, which assigns a distribution over actions $f_{a|h}(a|h;\theta)$ according to the observed history of states $h_t = s_0, \dots, s_t$. Then, an immediate random reward $\rho_t \sim f_{\rho|s,a}(\rho|s,a)$ is received, and the state transitions to s_{t+1} according to the MDP transition probability $f_{s'|s,a}(s'|s,a)$. We denote by ζ_0 the initial state distribution and by s^* a terminal state, and we assume that for all θ, s^* is reached w.p. 1.

For some policy π_θ, let $s_0, a_0, \rho_0, s_1, a_1, \rho_1, \dots, s_\tau$ denote a state-action-reward trajectory from the MDP under that policy, that terminates at time τ, i.e., $s_\tau = s^*$. The trajectory is a random variable, and we decompose[3] it into a discrete part $Y \doteq s_0, a_0, s_1, a_1, \dots, s^*$ and a continuous part $X \doteq \rho_0, \rho_1, \dots, \rho_{\tau-1}$. Our quantity of interest is the total reward along the trajectory $R \doteq \sum_{t=0}^{\tau} \rho_t$. In standard RL, the objective is to find the parameter θ that maximizes the expected return $V(\theta) = \mathbb{E}^\theta[R]$. Policy gradient methods (Baxter and Bartlett 2001; Marbach and Tsitsiklis 1998; Peters and Schaal 2008) use simulation to estimate $\partial V(\theta)/\partial \theta_j$, and then perform stochastic gradient ascent on the parameters θ. In this work we are risk-sensitive, and our goal is to *maximize the CVaR of the total return* $J(\theta) \doteq \Phi_\alpha(R;\theta)$. In the spirit of policy gradient methods, we estimate $\partial J(\theta)/\partial \theta_j$ from simulation, using `GCVaR`, and optimize θ using `CVaRSGD`. We now detail our approach.

First, it is well known (Marbach and Tsitsiklis 1998) that by the Markov property of the state transitions:

$$\partial \log f_Y\left(Y;\theta\right)/\partial \theta = \sum_{t=0}^{\tau-1} \partial \log f_{a|h}(a_t|h_t;\theta)/\partial \theta. \qquad (10)$$

Also, note that in our formulation we have

$$\partial \log f_{\mathbf{X}|Y}\left(x_i|y_i;\theta\right)/\partial \theta = 0, \qquad (11)$$

since the reward does not depend on θ directly.

To apply `CVaRSGD` in the RL setting, at each iteration i of the algorithm we simulate n_i trajectories $x_1, y_1, \dots, x_{n_i}, y_{n_i}$ of the MDP using policy π_{θ^i} (each x_k and y_k here together correspond to a single trajectory, as realizations of the random variables X and Y defined above). We then apply the `GCVaR` algorithm to obtain an estimate $\Delta_{j;n_i}$ of $\partial J(\theta)/\partial \theta_j$, using the simulated trajectories $x_1, y_1, \dots, x_{n_i}, y_{n_i}$, Eq. (10), and Eq. (11). Finally, we update the policy parameter according to Eq. (8). Note that due to Eq. (10), the transition probabilities of the MDP, which are generally not known to the decision maker, are not required for estimating the gradient using `GCVaR`. Only policy-dependent terms are required.

We should remark that for the standard RL criterion $V(\theta)$, a Markov policy that depends only on the current state suffices to achieve optimality (Bertsekas 2012). For the CVaR criterion this is not necessarily the case. Bäuerle and Ott (2011) show that under certain conditions, an augmentation of the current state with a function of the accumulated reward suffices for optimality. In our simulations, we used a Markov policy, and still obtained useful and sensible results.

Assumptions 4, 5, and 6, that are required for convergence of the algorithm, are reasonable for the RL setting, and may be satisfied, for example, when $f_{\rho|s,a}(\rho|s,a)$ is smooth, and $\partial \log f_{a|h}(a|h;\theta)/\partial \theta_j$ is well defined and bounded. This last condition is standard in policy gradient literature, and a popular policy representation that satisfies it is softmax action selection (Sutton et al. 2000; Marbach and Tsitsiklis 1998), given by $f_{a|h}(a|h;\theta) = \frac{\exp(\phi(h,a)^\top \theta)}{\sum_{a'} \exp(\phi(h,a')^\top \theta)}$, where $\phi(h,a) \in \mathbb{R}^k$ are a set of k features that depend on the history and action.

In some RL domains, the reward takes only discrete values. While this case is not specifically covered by the theory in this paper, one may add an arbitrarily small smooth noise to the total reward for our results to hold. Since such a modification has negligible impact on performance, this issue is of little importance in practice. In our experiments the reward was discrete, and we did not observe any problem.

5.1 Experimental Results

We examine Tetris as a test case for our algorithms. Tetris is a popular RL benchmark that has been studied extensively. The main challenge in Tetris is its large state space, which necessitates some form of approximation in the solution technique. Many approaches to learning controllers for Tetris are described in the literature, among them are approximate value iteration (Tsitsiklis and Van Roy 1996), policy gradients (Kakade 2001; Furmston and Barber 2012), and modified policy iteration (Gabillon, Ghavamzadeh, and Scherrer 2013). The standard performance measure in Tetris is the expected number of cleared lines in the game. Here, we are interested in a risk-averse performance measure, captured by the CVaR of the total game score. Our goal in this section is to compare the performance of a policy optimized for the CVaR criterion versus a policy obtained using the standard policy gradient method. As we will show, optimizing the CVaR indeed produces a different policy, characterized by a risk-averse behavior. We note that at present, the best results in the literature (for the standard performance measure) were obtained using a modified policy iteration

[2] Also known as a stochastic shortest path (Bertsekas 2012).

[3] This decomposition is not restrictive, and used only to illustrate the definitions of Section 2. One may alternatively consider a continuous state space, or discrete rewards, so long as Assumptions 4, 5, and 6 hold.

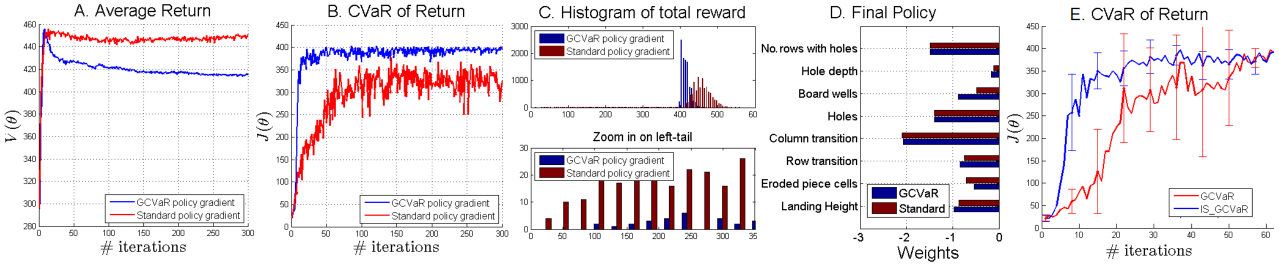


Figure 1: **GCVaR vs. policy gradient.** (A,B) Average return (A) and CVaR ($\alpha = 0.05$) of the return (B) for CVaRSGD and standard policy-gradient vs. iteration. (C) Histogram (counts from 10,000 independent runs) of the total return of the final policies. The lower plot is a zoom-in on the left-tail, and clearly shows the risk-averse behavior of the CVaRSGD policy. (D) Final policy parameters. Note the difference in the Board Well feature, which encourages risk taking. (E) CVaR ($\alpha = 0.01$) of the return for CVaRSGD vs. iteration, with and without importance sampling.

approach (Gabillon, Ghavamzadeh, and Scherrer 2013), and not using policy gradients. We emphasize that our goal here is not to compete with those results, but rather to illustrate the application of CVaRSGD. We do point out, however, that whether the approach of Gabillon, Ghavamzadeh, and Scherrer (2013) could be extended to handle a CVaR objective is currently not known.

We used the regular 10×20 Tetris board with the 7 standard shapes (a.k.a. *tetrominos*). In order to induce risk-sensitive behavior, we modified the reward function of the game as follows. The score for clearing 1,2,3 and 4 lines is 1,4,8 and 16 respectively. In addition, we limited the maximum number of steps in the game to 1000. These modifications strengthened the difference between the risk-sensitive and nominal policies, as they induce a tradeoff between clearing many 'single' lines with a low profit, or waiting for the more profitable, but less frequent, 'batches'.

We used the softmax policy, with the feature set of Thiery and Scherrer (2009). Starting from a fixed policy parameter θ_0, which was obtained by running several iterations of standard policy gradient (giving both methods a 'warm start'), we ran both CVaRSGD and standard policy gradient[4] for enough iterations such that both algorithms (approximately) converged. We set $\alpha = 0.05$ and $N = 1000$.

In Fig. 1A and Fig. 1B we present the average return $V(\theta)$ and CVaR of the return $J(\theta)$ for the policies of both algorithms at each iteration (evaluated by MC on independent trajectories). Observe that for CVaRSGD, the average return has been compromised for a higher CVaR value.

This compromise is further explained in Fig. 1C, where we display the reward distribution of the final policies. It may be observed that the left-tail distribution of the CVaR policy is significantly lower than the standard policy. For the risk-sensitive decision maker, such results are very important, especially if the left-tail contains catastrophic outcomes, as is common in many real-world domains, such as finance. To better understand the differences between the policies, we compare the final policy parameters θ in Fig. 1D. The most significant difference is in the parameter that corresponds to the Board Well feature. A *well* is a succession of unoccupied cells in a column, such that their left and right cells are both occupied. The controller trained by CVaRSGD has a smaller negative weight for this feature, compared to the standard controller, indicating that actions which create deep-wells are repressed. Such wells may lead to a high reward when they get filled, but are risky as they heighten the board.

To demonstrate the importance of IS in optimizing the CVaR when α is small, we chose $\alpha = 0.01$, and $N = 200$, and compared CVaRSGD against its IS version, IS_CVaRSGD, described in (Tamar, Glassner, and Mannor 2014). As Fig. 1E shows, IS_GCVaRSGD converged significantly faster, improving the convergence rate by more than a factor of 2. The full details are provided in (Tamar, Glassner, and Mannor 2014).

6 Conclusion and Future Work

We presented a novel LR-style formula for the gradient of the CVaR performance criterion. Based on this formula, we proposed a sampling-based gradient estimator, and a stochastic gradient descent procedure for CVaR optimization that is guaranteed to converge to a local optimum. To our knowledge, this is the first extension of the LR method to the CVaR performance criterion, and our results extend CVaR optimization to new domains.

We evaluated our approach empirically in an RL domain: learning a risk-sensitive policy for Tetris. To our knowledge, such a domain is beyond the reach of existing CVaR optimization approaches. Moreover, our empirical results show that optimizing the CVaR indeed results in useful risk-sensitive policies, and motivates the use of simulation-based optimization for risk-sensitive decision making.

Acknowledgments

The authors thank Odalric-Ambrym Maillard for many helpful discussions. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement n. 306638.

[4] Standard policy gradient is similar to CVaRSGD when $\alpha = 1$. However, it is common to subtract a baseline from the reward in order to reduce the variance of the gradient estimate. In our experiments, we used the average return $< r >$ as a baseline, and our gradient estimate was $\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log f_Y(y_i;\theta)}{\partial \theta_j}(r(x_i,y_i) - < r >)$.

References

Agarwal, V., and Naik, N. Y. 2004. Risks and portfolio decisions involving hedge funds. *Review of Financial Studies* 17(1):63–98.

Bardou, O.; Frikha, N.; and Pagès, G. 2009. Computing VaR and CVaR using stochastic approximation and adaptive unconstrained importance sampling. *Monte Carlo Methods and Applications* 15(3):173–210.

Bäuerle, N., and Ott, J. 2011. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research* 74(3):361–379.

Baxter, J., and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *JAIR* 15:319–350.

Bertsekas, D. P. 2012. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, 4th edition.

Borkar, V., and Jain, R. 2014. Risk-constrained Markov decision processes. *IEEE TAC* PP(99):1–1.

Borkar, V. S. 2001. A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. *Systems & Control Letters* 44(5):339–346.

Fu, M. C. 2006. Gradient estimation. In Henderson, S. G., and Nelson, B. L., eds., *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*. Elsevier. 575 – 616.

Furmston, T., and Barber, D. 2012. A unifying perspective of parametric policy search methods for Markov decision processes. In *Advances in Neural Information Processing Systems 25*.

Gabillon, V.; Ghavamzadeh, M.; and Scherrer, B. 2013. Approximate dynamic programming finally performs well in the game of tetris. In *Advances in Neural Information Processing Systems 26*.

Glynn, P. W. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33(10):75–84.

Hong, L. J., and Liu, G. 2009. Simulating sensitivities of conditional value at risk. *Management Science*.

Iyengar, G., and Ma, A. 2013. Fast gradient descent method for mean-CVaR optimization. *Annals of Operations Research* 205(1):203–212.

Kakade, S. 2001. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*.

Kushner, H., and Yin, G. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer Verlag.

Marbach, P., and Tsitsiklis, J. N. 1998. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control* 46(2):191–209.

Morimura, T.; Sugiyama, M.; Kashima, H.; Hachiya, H.; and Tanaka, T. 2010. Nonparametric return distribution approximation for reinforcement learning. In *International Conference on Machine Learning*, 799–806.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21(4):682–697.

Prashanth, L., and Ghavamzadeh, M. 2013. Actor-critic algorithms for risk-sensitive mdps. In *Advances in Neural Information Processing Systems 26*.

Prashanth, L. 2014. Policy gradients for CVaR-constrained MDPs. In *International Conference on Algorithmic Learning Theory*.

Rockafellar, R. T., and Uryasev, S. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2:21–42.

Rubinstein, R. Y., and Kroese, D. P. 2011. *Simulation and the Monte Carlo method*. John Wiley & Sons.

Scaillet, O. 2004. Nonparametric estimation and sensitivity analysis of expected shortfall. *Mathematical Finance*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. Cambridge Univ Press.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 13*.

Tamar, A.; Di Castro, D.; and Mannor, S. 2012. Policy gradients with variance related risk criteria. In *International Conference on Machine Learning*.

Tamar, A.; Glassner, Y.; and Mannor, S. 2014. Optimizing the CVaR via sampling. *arXiv:1404.3862*.

Thiery, C., and Scherrer, B. 2009. Improvements on learning tetris with cross entropy. *International Computer Games Association Journal* 32.

Tsitsiklis, J. N., and Van Roy, B. 1996. Feature-based methods for large scale dynamic programming. *Machine Learning* 22(1-3):59–94.

Chapter 6

Policy Gradient for Coherent Risk Measures

The following chapter is joint work with Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. This work was sent for publication, but not published yet.

Abbreviations

RL	—	Reinforcement learning
MDP	—	Markov decision process
SRP	—	Static risk problem
DRP	—	Dynamic risk problem
SGD	—	Stochastic gradient descent
$CVaR$	—	Conditional value-at-risk

Notations

$\mathcal{M}$	—	Markov decision process (MDP)
$\mathcal{X}$	—	State space
x	—	State
$\mathcal{U}$	—	Action space
a	—	Action

C	—	Cost
$C_{\max}$	—	Maximal cost
P	—	Transition probability
x_0	—	Initial state
γ	—	Discount factor
θ	—	Policy parameters
μ_θ	—	Policy
T	—	Time horizon
$(\Omega, \mathcal{F}, P_\theta)$	—	Probability space
$\mathcal{Z}$	—	Space of random variables $Z : \Omega \mapsto (-\infty, \infty)$
ρ	—	Static risk-measure
$\mathcal{B}$	—	Set of probability distributions
$\mathbb{E}_\xi[\cdot]$	—	A ξ-weighted expectation
$\mathcal{U}$	—	Risk envelope
ξ	—	Perturbation weights in coherent risk representation
$\mathcal{E}$	—	Set of equality constraints
g_e	—	Affine equality constraint
$\mathcal{I}$	—	Set of inequality constraints
f_i	—	Convex inequality constraint
M	—	Bound on constraint derivative
ρ_T	—	Dynamic-risk objective for horizon of length T
ρ_∞	—	Dynamic-risk objective for infinite horizon
L	—	Lagrangian
$\lambda^{\mathcal{P}}$	—	Lagrange multiplier for normalization constraint
$\lambda^{\mathcal{E}}$	—	Lagrange multipliers for equality constraints
$\lambda^{\mathcal{I}}$	—	Lagrange multipliers for inequality constraints
$(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta)$	—	Lagrangian saddle point
q_α	—	A $(1-\alpha)$-quantile
$\mathbb{SD}$	—	Semi-deviation
ρ_{MSD}	—	Mean-semideviation
N	—	Number of samples
$P_{\theta;N}$	—	Empirical distribution
ρ_N	—	Empirical risk
V_θ	—	Risk-sensitive value function for policy μ_θ
h_θ	—	Stage-wise cost function for policy gradient theorem

6.1 Abstract

Several authors have recently developed risk-sensitive policy gradient methods that augment the standard expected cost minimization problem with a measure of *variability* in cost. These studies have focused on *specific* risk-measures, such as the variance or conditional value at risk (CVaR). In this work, we extend the policy gradient method to *the whole class* of coherent risk measures, which is widely accepted in finance and operations research, among other fields. We consider both static and time-consistent dynamic risk measures. For static risk measures, our approach is in the spirit of *policy gradient* algorithms and combines a standard sampling approach with convex programming. For dynamic risk measures, our approach is *actor-critic* style and involves explicit approximation of value function. Most importantly, our contribution presents a *unified* approach to risk-sensitive reinforcement learning that generalizes and extends previous results.

6.2 Introduction

Risk-sensitive optimization considers problems in which the objective involves a *risk measure* of the random cost, in contrast to the typical *expected* cost objective. Such problems are important when the decision-maker wishes to manage the *variability* of the cost, in addition to its expected outcome, and are standard in various applications of finance and operations research. In reinforcement learning (RL) [100], risk-sensitive objectives have gained popularity as a means to regularize the variability of the total (discounted) cost/reward in a Markov decision process (MDP).

Many risk objectives have been investigated in the literature and applied to RL, such as the celebrated Markowitz mean-variance model [63], Value-at-Risk (VaR) and Conditional Value at Risk (CVaR) [68, 105, 83, 27, 20, 106]. The view taken in this paper is that the preference of one risk measure over another is *problem-dependent* and depends on factors such as the cost distribution, sensitivity to rare events, ease of estimation from data, and computational tractability of the optimization problem. However, the highly influential paper of Artzner et al. [2] identified a set of natural properties that are desirable for a risk measure to satisfy. Risk measures that satisfy these properties are termed *coherent* and have obtained widespread acceptance in financial applications, among others. We focus on such coherent measures of risk in this work.

For sequential decision problems, such as MDPs, another desirable property of a risk measure is *time consistency.* A time-consistent risk measure satisfies a "dynamic programming" style property: if a strategy is risk-optimal for an n-stage problem, then the component of the policy from the t-th time until the end (where $t < n$) is also risk-optimal (see principle of optimality in [11]). The recently proposed class of dynamic Markov coherent risk measures [89] satisfies both the coherence and time consistency properties.

In this work, we present policy gradient algorithms for RL with a coherent risk objective. Our approach applies to *the whole class* of coherent risk measures, thereby generalizing and unifying previous approaches that have focused on individual risk measures. We consider both *static* coherent risk of the total discounted return from an MDP and time-consistent *dynamic* Markov coherent risk. Our main contribution is formulating the risk-sensitive policy-gradient under the coherent-risk framework. More specifically, we provide:

- A new formula for the gradient of static coherent risk that is convenient for approximation using sampling.

- An algorithm for the gradient of general static coherent risk that involves sampling with convex programming and a corresponding consistency result.

- A new policy gradient theorem for Markov coherent risk, relating the gradient to a suitable *value function* and a corresponding actor-critic algorithm.

Several previous results are special cases of the results presented here; our approach allows to re-derive them in greater generality and simplicity.

Related Work Risk-sensitive optimization in RL for specific risk functions has been studied recently by several authors. [15] studied exponential utility functions, [68], [105], [83] studied mean-variance models, [20], [106] studied CVaR in the static setting, and [79], [21] studied dynamic coherent risk for systems with linear dynamics. Our paper presents a general method *for the whole class* of coherent risk measures (both static and dynamic) and is not limited to a specific choice within that class, nor to particular system dynamics.

Reference [75] showed that an MDP with a dynamic coherent risk objective is essentially a robust MDP. The planning for large scale MDPs was considered in [108], using an approximation of the value function. For many problems, approximation in the policy space is more suitable (see, e.g., [62]). Our sampling-based RL-style approach is suitable for approximations both in the policy and value function, and scales-up to large or continuous MDPs. We do, however, make use of a technique of [108] in a part of our method.

Optimization of coherent risk measures was thoroughly investigated by Ruszczynski and Shapiro [90] (see also [95]) for the stochastic programming case in which the policy parameters do not affect the distribution of the stochastic system (i.e., the MDP trajectory), but only the reward function, and thus, this approach is not suitable for most RL problems. For the case of MDPs and dynamic risk, [89] proposed a dynamic programming approach. This approach does not scale-up to large MDPs, due to the "curse of dimensionality". For further motivation of risk-sensitive policy gradient methods, we refer the reader to [68, 105, 83, 20, 106].

6.3 Preliminaries

Consider a probability space $(\Omega, \mathcal{F}, P_\theta)$, where Ω is the set of outcomes (sample space), $\mathcal{F}$ is a σ-algebra over Ω representing the set of events we are interested in, and $P_\theta \in \mathcal{B}$, where $\mathcal{B} := \left\{\xi : \int_{\omega\in\Omega} \xi(\omega) = 1, \xi \geq 0\right\}$ is the set of probability distributions, is a probability measure over $\mathcal{F}$ parameterized by some tunable parameter $\theta \in \mathbb{R}^K$. In the following, we suppress the notation of θ in θ-dependent quantities.

To ease the technical exposition, in this paper we restrict our attention to finite probability spaces, i.e., Ω has a finite number of elements. Our results can be extended to the L_p-normed spaces without loss of generality, but the details are omitted for brevity.

Denote by $\mathcal{Z}$ the space of random variables $Z : \Omega \mapsto (-\infty, \infty)$ defined over the probability space $(\Omega, \mathcal{F}, P_\theta)$. In this paper, a random variable $Z \in \mathcal{Z}$ is interpreted as a cost, i.e., the smaller the realization of Z, the better. For $Z, W \in \mathcal{Z}$, we denote by $Z \leq W$ the point-wise partial order, i.e., $Z(\omega) \leq$

$W(\omega)$ for all $\omega \in \Omega$. We denote by $\mathbb{E}_\xi[Z] \doteq \sum_{\omega\in\Omega} P_\theta(\omega)\xi(\omega)Z(\omega)$ a ξ-weighted expectation of Z.

An MDP is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{U}, C, P, \gamma, x_0)$, where $\mathcal{X}$ and $\mathcal{U}$ are the state and action spaces; $C(x) \in [-C_{\max}, C_{\max}]$ is a bounded, deterministic, and state-dependent cost; $P(\cdot|x,a)$ is the transition probability distribution; γ is a discount factor; and x_0 is the initial state.[1] Actions are chosen according to a θ-parameterized stationary Markov[2] policy $\mu_\theta(\cdot|x)$. We denote by $x_0, a_0, \ldots, x_T, a_T$ a trajectory of length T drawn by following the policy μ_θ in the MDP.

6.3.1 Coherent Risk Measures

A *risk measure* is a function $\rho : \mathcal{Z} \to \mathbb{R}$ that maps an uncertain outcome Z to the extended real line $\mathbb{R} \cup \{+\infty, -\infty\}$, e.g., the expectation $\mathbb{E}[Z]$ or the conditional value-at-risk (CVaR) $\min_{\nu\in\mathbb{R}} \big\{\nu + \frac{1}{\alpha}\mathbb{E}\big[(Z-\nu)^+\big]\big\}$. A risk measure is called *coherent*, if it satisfies the following conditions for all $Z, W \in \mathcal{Z}$ [2]:

A1 Convexity: $\forall\lambda \in [0,1],\ \rho\big(\lambda Z + (1-\lambda)W\big) \leq \lambda\rho(Z) + (1-\lambda)\rho(W)$;

A2 Monotonicity: if $Z \leq W$, then $\rho(Z) \leq \rho(W)$;

A3 Translation invariance: $\forall a \in \mathbb{R},\ \rho(Z+a) = \rho(Z) + a$;

A4 Positive homogeneity: if $\lambda \geq 0$, then $\rho(\lambda Z) = \lambda\rho(Z)$.

Intuitively, these condition ensure the “rationality” of single-period risk assessments: A1 ensures that diversifying an investment will reduce its risk; A2 guarantees that an asset with a higher cost for every possible scenario is indeed riskier; A3, also known as ‘cash invariance’, means that the deterministic part of an investment portfolio does not contribute to its risk; the intuition behind A4 is that doubling a position in an asset doubles its risk. We further refer the reader to [2] for a more detailed motivation of coherent risk.

[1] Our results may easily be extended to random costs, state-action dependent costs, and random initial states.

[2] For the dynamic Markov risk we study, an optimal policy is stationary Markov, while this is not necessarily the case for the static risk. Our results can be extended to history-dependent policies or stationary Markov policies on a state space augmented with the accumulated cost. The latter has shown to be sufficient for optimizing the CVaR risk [8].

The following representation theorem [95] shows an important property of coherent risk measures that is fundamental to our gradient-based approach.

Theorem 6.1 *A risk measure $\rho : \mathcal{Z} \to \mathbb{R}$ is coherent if and only if there exists a convex bounded and closed set $\mathcal{U} \subset \mathcal{B}$ such that*[3]

$$\rho(Z) = \max_{\xi \,:\, \xi P_\theta \in \mathcal{U}(P_\theta)} \mathbb{E}_\xi[Z]. \tag{6.1}$$

The result essentially states that any coherent risk measure is an expectation w.r.t. a worst-case density function ξP_θ, chosen adversarially from a suitable set of test density functions $\mathcal{U}(P_\theta)$, referred to as *risk envelope.* Moreover, it means that any coherent risk measure is *uniquely represented* by its risk envelope. Thus, in the sequel, we shall interchangeably refer to coherent risk-measures either by their explicit functional representation, or by their corresponding risk-envelope.

In this paper, we assume that the risk envelop $\mathcal{U}(P_\theta)$ is given in a canonical convex programming formulation, and satisfies the following conditions.

Assumption 1 (The General Form of Risk Envelope) *For each given policy parameter $\theta \in \mathbb{R}^K$, the risk envelope $\mathcal{U}$ of a coherent risk measure can be written as*

$$\mathcal{U}(P_\theta) = \left\{ \xi P_\theta \colon\, g_e(\xi, P_\theta) = 0,\ \forall e \in \mathcal{E},\ f_i(\xi, P_\theta) \leq 0,\ \forall i \in \mathcal{I},\ \sum_{\omega \in \Omega} \xi(\omega) P_\theta(\omega) = 1,\ \xi(\omega) \geq 0 \right\}, \tag{6.2}$$

where each constraint $g_e(\xi, P_\theta)$ is an affine function in ξ, each constraint $f_i(\xi, P_\theta)$ is a convex function in ξ, and there exists a strictly feasible point $\overline{\xi}$. $\mathcal{E}$ and $\mathcal{I}$ here denote the sets of equality and inequality constraints, respectively. Furthermore, for any given $\xi \in \mathcal{B}$, $f_i(\xi, p)$ and $g_e(\xi, p)$ are twice differentiable in p, and there exists a $M > 0$ such that

$$\max \left\{ \max_{i \in \mathcal{I}} \left| \frac{df_i(\xi, p)}{dp(\omega)} \right|, \max_{e \in \mathcal{E}} \left| \frac{dg_e(\xi, p)}{dp(\omega)} \right| \right\} \leq M,\ \forall \omega \in \Omega.$$

[3] When we study risk in MDPs, the risk envelope $\mathcal{U}(P_\theta)$ in Eq. 6.1 also depends on the state x.

Assumption 1 implies that the risk envelope $\mathcal{U}(P_\theta)$ is known in an *explicit* form. From Theorem 6.6 of [95], in the case of a finite probability space, ρ is a coherent risk if and only if $\mathcal{U}(P_\theta)$ is a convex and compact set. This justifies the affine assumption of g_e and the convex assumption of f_i. Moreover, the additional assumption on the smoothness of the constraints holds for many popular coherent risk measures, such as the CVaR, the mean-semi-deviation, and spectral risk measures [1].

6.3.2 Dynamic Risk Measures

The risk measures defined above do not take into account any temporal structure that the random variable might have, such as when it is associated with the return of a trajectory in the case of MDPs. In this sense, such risk measures are called *static*. *Dynamic* risk measures, on the other hand, explicitly take into account the temporal nature of the stochastic outcome. A primary motivation for considering such measures is the issue of *time consistency*, usually defined as follows [89]: if a certain outcome is considered less risky in all states of the world at stage $t+1$, then it should also be considered less risky at stage t. Example 2.1 in [44] shows the importance of time consistency in the evaluation of risk in a dynamic setting. It illustrates that for multi-period decision-making, optimizing a static measure can lead to "time-inconsistent" behavior. Similar paradoxical results could be obtained with other risk metrics; we refer the readers to [89] and [44] for further insights.

Markov Coherent Risk Measures. Markov risk measures were introduced in [89] and are a useful class of dynamic time-consistent risk measures that are particularly important for our study of risk in MDPs. For a T-length horizon and MDP $\mathcal{M}$, the Markov coherent risk measure $\rho_T(\mathcal{M})$ is

$$\rho_T(\mathcal{M}) = C(x_0) + \gamma\rho\Bigg(C(x_1) + \ldots + \gamma\rho\Big(C(x_{T-1}) + \gamma\rho\big(C(x_T)\big)\Big)\Bigg), \qquad (6.3)$$

where ρ is a static coherent risk measure that satisfies Assumption 1 and $x_0, \ldots, x_T$ is a trajectory drawn from the MDP $\mathcal{M}$ under policy μ_θ. It is important to note that in (6.3), each static coherent risk ρ at state $x \in \mathcal{X}$

is induced by the transition probability $P_\theta(\cdot|x) = \sum_{a\in\mathcal{A}} P(x'|x,a)\mu_\theta(a|x)$. We also define $\rho_\infty(\mathcal{M}) \doteq \lim_{T\to\infty} \rho_T(\mathcal{M})$, which is well-defined since $\gamma < 1$ and the cost is bounded. We further assume that ρ in (6.3) is a *Markov risk* measure, i.e., the evaluation of each static coherent risk measure ρ is not allowed to depend on the whole past.

6.4 Problem Formulation

In this paper, we are interested in solving two risk-sensitive optimization problems. Given a random variable Z and a static coherent risk measure ρ as defined in Section 6.3, the static risk problem (SRP) is given by

$$\min_\theta \quad \rho(Z). \tag{6.4}$$

For example, in an RL setting, Z may correspond to the cumulative discounted cost $Z = C(x_0) + \gamma C(x_1) + \cdots + \gamma^T C(x_T)$ of a trajectory induced by an MDP with a policy parameterized by θ.

For an MDP $\mathcal{M}$ and a dynamic Markov coherent risk measure ρ_T as defined by Eq. 6.3, the dynamic risk problem (DRP) is given by

$$\min_\theta \quad \rho_\infty(\mathcal{M}). \tag{6.5}$$

Except for very limited cases, there is no reason to hope that neither the SRP in (6.4) nor the DRP in (6.5) should be tractable problems, since the dependence of the risk measure on θ may be complex and non-convex. In this work, we aim towards a more modest goal and search for a *locally* optimal θ. Thus, the main problem that we are trying to solve in this paper is how to calculate the gradients of the SRP's and DRP's objective functions

$$\nabla_\theta \rho(Z) \qquad \text{and} \qquad \nabla_\theta \rho_\infty(\mathcal{M}).$$

We are interested in non-trivial cases in which the gradients cannot be calculated analytically. In the static case, this would correspond to a non-trivial dependence of Z on θ. For dynamic risk, we also consider cases where the state space is too large for a tractable computation. Our approach for dealing with such difficult cases is through sampling. We assume that in the static case, we may obtain i.i.d. samples of the random variable Z. For the dynamic case, we assume that for each state and action (x, a) of the MDP, we may obtain i.i.d. samples of the next state $x' \sim P(\cdot|x, a)$. We show that sampling may indeed be used in both cases to devise suitable estimators for the gradients.

To finally solve the SRP and DRP problems, a gradient estimate may be plugged into a standard stochastic gradient descent (SGD) algorithm for learning a locally optimal solution to (6.4) and (6.5). From the structure of the dynamic risk in Eq. 6.3, one may think that a gradient estimator for $\rho(Z)$ may help us to estimate the gradient $\nabla_\theta \rho_\infty(\mathcal{M})$. Indeed, we follow this idea and begin with estimating the gradient in the static risk case.

6.5 Gradient Formula for Static Risk

In this section, we consider a static coherent risk measure $\rho(Z)$ and propose sampling-based estimators for $\nabla_\theta \rho(Z)$. We make the following assumption on the policy parametrization, which is standard in the policy gradient literature [62].

Assumption 2 *The likelihood ratio $\nabla_\theta \log P(\omega)$ is well-defined and bounded for all $\omega \in \Omega$.*

Moreover, our approach implicitly assumes that given some $\omega \in \Omega$, $\nabla_\theta \log P(\omega)$ may be easily calculated. This is also a standard requirement for policy gradient algorithms [62] and is satisfied in various applications such as queueing systems, inventory management, and financial engineering (see, e.g., the survey by Fu [33]).

Using Theorem 6.1 and Assumption 1, for each θ, we have that $\rho(Z)$ is the solution to the convex optimization problem (6.1) (for that value of θ). The Lagrangian function of (6.1), denoted by $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$, may be written as

$$\begin{aligned} L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = & \sum_{\omega \in \Omega} \xi(\omega) P_\theta(\omega) Z(\omega) - \lambda^{\mathcal{P}} \left(\sum_{\omega \in \Omega} \xi(\omega) P_\theta(\omega) - 1 \right) \\ & - \sum_{e \in \mathcal{E}} \lambda^{\mathcal{E}}(e) g_e(\xi, P_\theta) - \sum_{i \in \mathcal{I}} \lambda^{\mathcal{I}}(i) f_i(\xi, P_\theta). \end{aligned} \tag{6.6}$$

The convexity of (6.1) and its strict feasibility due to Assumption 1 implies that $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ has a non-empty set of saddle points $\mathcal{S}$. The next theorem presents a formula for the gradient $\nabla_\theta \rho(Z)$. As we shall subsequently show, this formula is particularly convenient for devising sampling based estimators for $\nabla_\theta \rho(Z)$.

Theorem 6.2 *Let Assumptions 1 and 2 hold. For any saddle point* $(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) \in \mathcal{S}$ *of* (6.6), *we have*

$$\begin{aligned}\nabla_\theta \rho(Z) =& \mathbb{E}_{\xi^*_\theta}\left[\nabla_\theta \log P(\omega)(Z - \lambda^{*,\mathcal{P}}_\theta)\right] \\ &- \sum_{e\in\mathcal{E}} \lambda^{*,\mathcal{E}}_\theta(e) \nabla_\theta g_e(\xi^*_\theta; P_\theta) \\ &- \sum_{i\in\mathcal{I}} \lambda^{*,\mathcal{I}}_\theta(i) \nabla_\theta f_i(\xi^*_\theta; P_\theta).\end{aligned}$$

The proof of this theorem, given in the supplementary material, involves an application of the Envelope theorem [65] and a standard 'likelihood-ratio' trick. We now demonstrate the utility of Theorem 6.2 with several examples in which we show that it generalizes previously known results, and also enables deriving new useful gradient formulas.

6.5.1 Example 1: CVaR

The CVaR at level $\alpha \in [0, 1]$ of a random variable Z, denoted by $\rho_{\text{CVaR}}(Z; \alpha)$, is a very popular coherent risk measure [87], defined as

$$\rho_{\text{CVaR}}(Z; \alpha) \doteq \inf_{t\in\mathbb{R}} \left\{ t + \alpha^{-1}\mathbb{E}\left[(Z - t)_+\right] \right\}.$$

When Z is continuous, $\rho_{\text{CVaR}}(Z; \alpha)$ is well-known to be the mean of the α-tail distribution of Z, $\mathbb{E}\left[Z \middle| Z > q_\alpha\right]$, where q_α is a $(1 - \alpha)$-quantile of Z. Thus, selecting a small α makes CVaR particularly sensitive to rare, but very high costs.

The risk envelope for CVaR is known to be [95] $\mathcal{U} = \left\{\xi P_\theta : \xi(\omega) \in [0, \alpha^{-1}],\quad \sum_{\omega\in\Omega} \xi(\omega) P_\theta(\omega) = 1\right\}$. Furthermore, [95] show that the saddle points of (6.6) satisfy $\xi^*_\theta(\omega) = \alpha^{-1}$ when $Z(\omega) > \lambda^{*,\mathcal{P}}_\theta$, and $\xi^*_\theta(\omega) = 0$ when $Z(\omega) < \lambda^{*,\mathcal{P}}_\theta$, where $\lambda^{*,\mathcal{P}}_\theta$ is any $(1 - \alpha)$-quantile of Z. Plugging this result into Theorem 6.2, we can easily show that

$$\nabla_\theta \rho_{\mathrm{CVaR}}(Z;\alpha) = \mathbb{E}\left[\nabla_\theta \log P(\omega)(Z - q_\alpha)\right| Z(\omega) > q_\alpha\left].$$

This formula was recently proved in [106] for the case of continuous distributions by an explicit calculation of the conditional expectation, and under several additional smoothness assumptions. Here we show that it holds regardless of these assumptions and in the discrete case as well. Our proof is also considerably simpler.

6.5.2 Example 2: Mean-Semideviation

The semi-deviation of a random variable Z is defined as

$$\mathbb{SD}[Z] \doteq \left(\mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])_+^2\right]\right)^{1/2}.$$

The semi-deviation captures the variation of the cost only *above its mean*, and is an appealing alternative to the standard deviation, which does not distinguish between the variability of upside and downside deviations. For some $\alpha \in [0,1]$, the *mean-semideviation* risk measure is defined as $\rho_{\mathrm{MSD}}(Z;\alpha) \doteq \mathbb{E}\left[Z\right] + \alpha \mathbb{SD}[Z]$, and is a coherent risk measure [95]. We have the following result:

Proposition 6.3 *Under Assumption 2, with* $\nabla_\theta \mathbb{E}\left[Z\right] = \mathbb{E}\left[\nabla_\theta \log P(\omega) Z\right]$, *we have*

$$\nabla_\theta \rho_{MSD}(Z;\alpha) = \nabla_\theta \mathbb{E}\left[Z\right] + \frac{\alpha \mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])_+ (\nabla_\theta \log P(\omega)(Z - \mathbb{E}\left[Z\right]) - \nabla_\theta \mathbb{E}\left[Z\right])\right]}{\mathbb{SD}(Z)}.$$

This proposition can be used to devise a sampling based estimator for $\nabla_\theta \rho_{\mathrm{MSD}}(Z;\alpha)$ by replacing all the expectations with sample averages. The algorithm along with the proof of the proposition are in the supplementary material. In Section 6.7 we provide a numerical illustration of optimization with a mean-semideviation objective.

6.5.3 General Gradient Estimation Algorithm

In the two previous examples, we obtained a gradient formula by *analytically* calculating the Lagrangian saddle point (6.6) and plugging it into the formula of Theorem 6.2. We now consider a general coherent risk $\rho(Z)$ for which, in contrast to the CVaR and mean-semideviation cases, the Lagrangian saddle-point is not known analytically. *We only assume that we know the structure of the risk-envelope* as given by (6.2). We show that in this case, $\nabla_\theta \rho(Z)$ may be estimated using a *sample average approximation* (SAA; [95]) of the formula in Theorem 6.2.

Assume that we are given N i.i.d. samples $\omega_i \sim P_\theta$, $i = 1, \dots, N$, and let $P_{\theta;N}(\omega) \doteq \frac{1}{N}\sum_{i=1}^N \mathbb{I}\left\{\omega_i = \omega\right\}$ denote the corresponding empirical distribution. Also, let the *sample risk envelope* $\mathcal{U}(P_{\theta;N})$ be defined according to Eq. 6.2 with P_θ replaced by $P_{\theta;N}$. Consider the following SAA version of the optimization in Eq. 6.1:

$$\rho_N(Z) = \max_{\xi:\xi P_{\theta;N}\in\mathcal{U}(P_{\theta;N})} \sum_{i\in 1,\dots,N} P_{\theta;N}(\omega_i)\xi(\omega_i)Z(\omega_i). \tag{6.7}$$

Note that (6.7) defines a convex optimization problem with $\mathcal{O}(N)$ variables and constraints. In the following, we assume that a solution to (6.7) may be computed efficiently using standard convex programming tools such as interior point methods [18]. Let $\xi^*_{\theta;N}$ denote a solution to (6.7) and $\lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}$ denote the corresponding KKT multipliers, which can be obtained from the convex programming algorithm [18]. We propose the following estimator for the gradient-based on Theorem 6.2:

$$\begin{aligned}\nabla_{\theta;N}\rho(Z) = &\sum_{i=1}^N P_{\theta;N}(\omega_i)\xi^*_{\theta;N}(\omega_i)\nabla_\theta \log P(\omega_i)(Z(\omega_i) - \lambda^{*,\mathcal{P}}_{\theta;N}) \\ &- \sum_{e\in\mathcal{E}} \lambda^{*,\mathcal{E}}_{\theta;N}(e)\nabla_\theta g_e(\xi^*_{\theta;N}; P_{\theta;N}) - \sum_{i\in\mathcal{I}} \lambda^{*,\mathcal{I}}_{\theta;N}(i)\nabla_\theta f_i(\xi^*_{\theta;N}; P_{\theta;N}).\end{aligned} \tag{6.8}$$

Thus, our gradient estimation algorithm is a two-step procedure involving *both sampling and convex programming.* In the following, we show that under some conditions on the set $\mathcal{U}(P_\theta)$, $\nabla_{\theta;N}\rho(Z)$ is a consistent estimator of $\nabla_\theta \rho(Z)$. The proof has been reported in the supplementary material.

Proposition 6.4 *Let Assumptions 1 and 2 hold. Suppose there exists a compact set $C = C_\xi \times C_\lambda$ such that: (I) The set of Lagrangian saddle points $\mathcal{S} \subset C$ is non-empty and bounded. (II) The functions $f_e(\xi, P_\theta)$ for all $e \in \mathcal{E}$ and $f_i(\xi, P_\theta)$ for all $i \in \mathcal{I}$ are finite-valued and continuous (in ξ) on C_ξ. (III) For N large enough, the set $\mathcal{S}_N$ is non-empty and $\mathcal{S}_N \subset C$ w.p. 1.*

Further assume that: (IV) If $\xi_N P_{\theta;N} \in \mathcal{U}(P_{\theta;N})$ and ξ_N converges w.p. 1 to a point ξ, then $\xi P_\theta \in \mathcal{U}(P_\theta)$. We then have that $\lim_{N\to\infty} \rho_N(Z) = \rho(Z)$ *and* $\lim_{N\to\infty} \nabla_{\theta;N}\rho(Z) = \nabla_\theta \rho(Z)$ *w.p. 1.*

The set of assumptions for Proposition 6.4 is large, but rather mild. Note that (I) is implied by the Slater condition of Assumption 1. For satisfying (III), we need that the risk be well-defined for every empirical distribution, which is a natural requirement. Since $P_{\theta;N}$ always converges to P_θ uniformly on Ω, (IV) essentially requires smoothness of the constraints. We remark that in particular, constraints (I) to (IV) are satisfied for the popular CVaR, mean-semideviation, and spectral risk measures.

To summarize this section, we have seen that by exploiting the special structure of coherent risk measures in Theorem 6.1 and by the envelope-theorem style result of Theorem 6.2, we were able to derive sampling-based, likelihood-ratio style algorithms for estimating the policy gradient $\nabla_\theta \rho(Z)$ of coherent static risk measures. The gradient estimation algorithms developed here for static risk measures will be used as a sub-routine in our subsequent treatment of dynamic risk measures.

6.6 Gradient Formula for Dynamic Risk

In this section, we derive a new formula for the gradient of the Markov coherent dynamic risk measure, $\nabla_\theta \rho_\infty(\mathcal{M})$. Our approach is based on combining the static gradient formula of Theorem 6.2, with a dynamic-programming decomposition of $\rho_\infty(\mathcal{M})$.

The risk-sensitive *value-function* for an MDP $\mathcal{M}$ under the policy θ is defined as $V_\theta(x) = \rho_\infty(\mathcal{M}|x_0 = x)$, where with a slight abuse of notation, $\rho_\infty(\mathcal{M}|x_0 = x)$ denotes the Markov-coherent dynamic risk in (6.3) when the initial state x_0 is x. It is shown in [89] that due to the structure of the Markov dynamic risk $\rho_\infty(\mathcal{M})$, the value function is the unique solution to the *risk-sensitive Bellman equation*

$$V_\theta(x) = C(x) + \gamma \max_{\xi P_\theta(\cdot|x) \in \mathcal{U}(x, P_\theta(\cdot|x))} \mathbb{E}_\xi[V_\theta(x')], \tag{6.9}$$

where the expectation is taken over the next state transition. Note that by definition, we have $\rho_\infty(\mathcal{M}) = V_\theta(x_0)$, and thus, $\nabla_\theta \rho_\infty(\mathcal{M}) = \nabla_\theta V_\theta(x_0)$.

We now develop a formula for $\nabla_\theta V_\theta(x)$; this formula extends the well-known "policy gradient theorem" [102, 51], developed for the expected return, to Markov-coherent dynamic risk measures. We make a standard assumption, analogous to Assumption 2 of the static case.

Assumption 3 *The likelihood ratio $\nabla_\theta \log \mu_\theta(a|x)$ is well-defined and bounded for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$.*

For each state $x \in \mathcal{X}$, let $(\xi^*_{\theta,x}, \lambda^{*,\mathcal{P}}_{\theta,x}, \lambda^{*,\mathcal{E}}_{\theta,x}, \lambda^{*,\mathcal{I}}_{\theta,x})$ denote a saddle point of (6.6), corresponding to the state x, with $P_\theta(\cdot|x)$ replacing P_θ in (6.6) and V_θ replacing Z. The next theorem presents a formula for $\nabla_\theta V_\theta(x)$; the proof is in the supplementary material.

Theorem 6.5 *Under Assumptions 1 and 3, we have*

$$\nabla V_\theta(x) = \mathbb{E}_{\xi^*_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \mu_\theta(a_t|x_t) h_\theta(x_t, a_t) \middle| x_0 = x\right],$$

where $\mathbb{E}_{\xi^*_\theta}[\cdot]$ denotes the expectation w.r.t. trajectories generated by the Markov chain with transition probabilities $P_\theta(\cdot|x)\xi^*_{\theta,x}(\cdot)$, and the stage-wise cost function $h_\theta(x,a)$ is defined as

$$\begin{aligned} h_\theta(x,a) =& C(x) + \sum_{x' \in \mathcal{X}} P(x'|x,a)\xi^*_{\theta,x}(x') \Bigg[\gamma V_\theta(x') - \lambda^{*,\mathcal{P}}_{\theta,x} \\ & - \sum_{i\in\mathcal{I}} \lambda^{*,\mathcal{I}}_{\theta,x}(i) \frac{df_i(\xi^*_{\theta,x}, p)}{dp(x')} - \sum_{e\in\mathcal{E}} \lambda^{*,\mathcal{E}}_{\theta,x}(e) \frac{dg_e(\xi^*_{\theta,x}, p)}{dp(x')}\Bigg]. \end{aligned}$$

Theorem 6.5 may be used to develop an *actor-critic* style [102, 51] sampling-based algorithm for solving the DRP problem (6.5), composed of two interleaved procedures:

Critic: For a given policy θ, calculate the risk-sensitive value function V_θ, and

Actor: Using the critic's V_θ and Theorem 6.5, estimate $\nabla_\theta \rho_\infty(\mathcal{M})$ and update θ.

For the critic, the main challenge is calculating the value function when the state space $\mathcal{X}$ is large and dynamic programming cannot be applied due to the 'curse of dimensionality'. To overcome this, we exploit the fact that by (6.9), the risk-sensitive value-function V_θ is equivalent to the value function in a robust MDP (this fact was already noted by Osogami [75]), as studied in Chapter 4. Thus, our approximate policy-evaluation algorithm for robust MDPs of Chapter 4 (also in [108]) may be used to estimate V_θ using function approximation.

For the actor, the main challenge is that in order to estimate the gradient using Thm. 6.5, we need to sample from an MDP with ξ^*_θ-weighted transitions. Also, $h_\theta(x, a)$ involves an expectation for each s and a. Therefore, we propose a *two-phase sampling procedure* to estimate ∇V_θ in which we first use the critic's estimate of V_θ to derive ξ^*_θ, and sample a trajectory from an MDP with ξ^*_θ-weighted transitions. For each state in the trajectory, we then sample several next states to estimate $h_\theta(x, a)$.

6.7 Numerical Illustration

In this section, we illustrate our approach with a numerical example. The purpose of this illustration is to emphasize the importance of *flexibility* in designing risk criteria for selecting an *appropriate* risk-measure – such that suits both the user's risk preference *and* the problem-specific properties.

We consider a trading agent that can invest in one of three assets (see Figure 6.1 for their distributions). The returns of the first two assets, $A1$ and $A2$, are normally distributed: $A1 \sim \mathcal{N}(1, 1)$ and $A2 \sim \mathcal{N}(4, 6)$. The return of the third asset $A3$ has a Pareto distribution: $f(z) = \frac{\alpha}{z^{\alpha+1}}$ $\forall z > 1$, with $\alpha = 1.5$. The mean of the return from $A3$ is 3 and its variance is infinite; such heavy-tailed distributions are widely used in financial modeling [85]. The agent selects an action randomly, with probability $P(A_i) \propto \exp(\theta_i)$, where $\theta \in \mathbb{R}^3$ is the policy parameter. We trained three different policies π_1, π_2, and π_3. Policy π_1 is risk-neutral, i.e., $\max_\theta \mathbb{E}[Z]$, and it was trained using standard policy gradient [62]. Policy π_2 is risk-averse and had a mean-semideviation objective $\max_\theta \mathbb{E}[Z] - \mathbb{SD}[Z]$, and was trained using the algorithm in Section 6.5. Policy π_3 is also risk-averse, with a mean-standard-deviation objective, as proposed in [105, 83], $\max_\theta \mathbb{E}[Z] - \sqrt{\mathrm{Var}[Z]}$, and was trained using the algorithm of [105]. For each of these policies, Figure 6.1 shows the probability of selecting each asset vs. training iterations. Although $A2$ has the highest mean return, the risk-averse policy π_2 chooses $A3$, since it has a lower downside, as expected. However, because of the heavy upper-tail of $A3$, policy π_3 opted to choose $A1$ instead. This is counter-intuitive

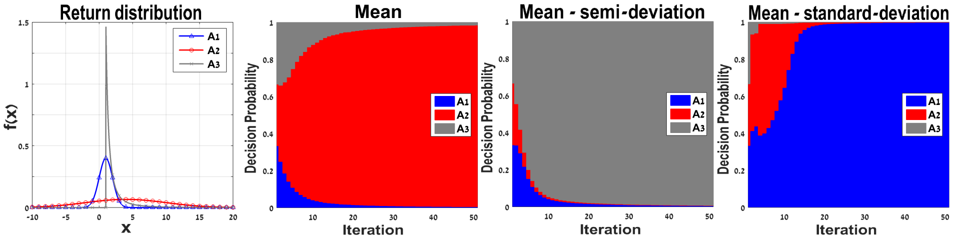


Figure 6.1: Numerical illustration - selection between 3 assets. A: Probability density of asset return. B,C,D: Bar plots of the probability of selecting each asset vs. training iterations, for policies π_1, π_2, and π_3, respectively. At each iteration, 10,000 samples were used for gradient estimation.

as a rational investor should not avert high returns. In fact, in this case $A3$ stochastically dominates $A1$ [41].

6.8 Conclusion

We presented algorithms for estimating the gradient of both static and dynamic coherent risk measures using two new policy gradient style formulas that combine sampling with convex programming. Thereby, our approach extends risk-sensitive RL to the whole class of coherent risk measures, and generalizes several recent studies that focused on specific risk measures.

On the technical side, an important future direction is to improve the convergence rate of gradient estimates using importance sampling methods. This is especially important for risk criteria that are sensitive to rare events, such as the CVaR [5].

From a more conceptual point of view, the coherent-risk framework explored in this work provides the decision maker with *flexibility* in designing risk preference. As our numerical example shows, such flexibility is important for selecting appropriate *problem-specific* risk measures for managing the cost variability. However, we believe that our approach has much more potential than that.

In almost every real-world application, uncertainty emanates from stochastic dynamics, but also, and perhaps more importantly, from modeling errors (model uncertainty). A prudent policy should protect against *both* types of uncertainties. The representation duality of coherent-risk (Theorem 6.1), naturally relates the risk to model uncertainty. In [75], a similar connection was made between model-uncertainty in MDPs and dynamic Markov coherent risk. We believe that by carefully shaping the risk-criterion, the decision maker may be able to take uncertainty into account in a *broad* sense. Designing a principled procedure for such *risk-shaping* is not trivial, and is beyond the scope of this paper. However, we believe that there is much potential to risk shaping as it may be the key for handling model misspecification in dynamic decision making.

Chapter 7

Discussion

This dissertation presented several methods for extending the RL methodology to accommodate a risk-sensitive approach towards uncertainty. In our work, we particulary focused on approximate methods (approximating either the value function, the policy, or both), which may scale-up to problems with large, or continuous state-spaces, as occur in real-world application domains.

We addressed the two most-important factors that contribute to uncertainty in sequential decision-making – inherent uncertainty, which is the result of the system stochasticity, and model uncertainty, which is the outcome of modeling errors and parameter uncertainty.

For the case of inherent uncertainty, we presented several policy-gradient style algorithms that optimize a *risk-measure* of the return, as opposed to the conventional expected return. The variety of the risk-measures explored in this work, which includes the variance, CVaR, and the whole class of coherent-risk, covers a large fraction of the most-popular risk measures in the literature [95], and provides the decision maker great flexibility in designing her risk preferences. The policy-gradient approach has been used efficiently in many RL applications [62, 68, 26], and scales-up to large domains by incorporating an approximation in the policy space.

We have empirically evaluated our policy-gradient algorithms on a financial domain (Figure 2.1), and on the domain of learning to play Tetris in a risk-averse style (Figure 5.1). Both of these problems required approximation to handle the continuous (in the financial domain), or the large discrete (in the Tetris domain) state-space. Indeed, as our results demonstrate, our algorithms were successful in finding a suitable risk-sensitive solution to each problem. Evidently, our algorithms may be successfully used for non-trivial risk-sensitive sequential decision-making problems.

In addition, we have extended the temporal-difference approach for value-function evaluation to the case of learning a value-function for the *variance* of the return. As we have shown (Figure 3.1), our approach provides a significant improvement in terms of sample-efficiency over the current state-of-the-art. Furthermore, it paves the way for designing *actor-critic* algorithms for variance-based objectives, which are preferable to 'vanilla' policy-gradient methods in terms of gradient-estimation variance [77]. Since their publication, our results were indeed used for pursuing such an actor-critic approach for the variance [83, 107].

For the case of model uncertainty, we have extended the popular robust MDP framework to accommodate function-approximation in the value-function, thereby scaling-up the robust MDP approach to potentially large-scale problems. As our experiments show (Figure 4.1), the 'worst-case' approach taken towards model uncertainty in robust MDPs provides a means for handling the important problem of *model mis-specification.* In addition, our experiments corroborate the proposition [60] that the potential performance degradation due to such modeling errors may be significant.

In the following, we further discuss the implications of our work, and its potential future extensions.

7.1 The Relationship Between Inherent Uncertainty and Model Uncertainty

In this dissertation, we followed a strict dichotomy between inherent uncertainty and model uncertainty, which reflected in the different algorithmic approaches for dealing with each source of uncertainty. However, the representation theorem of coherent risk (Theorem 6.1), which formulates the risk as a worst-case expectation over model-perturbations, hints that there is a link between the two uncertainty types.

The first work to relate these uncertainties in the context of sequential decision making was the work of Osogami [75], which related between dynamic Markov coherent-risk and robust MDPs. The underlying relationship between the two uncertainty factors may be observed in the Bellman-equation for the value-function for dynamic Markov coherent-risk (cf. Eq. 6.9):

$$V_\theta(x) = C(x) + \gamma \max_{\xi P_\theta(\cdot|x) \in \mathcal{U}(x, P_\theta(\cdot|x))} \mathbb{E}_\xi[V_\theta(x')],$$

which is equivalent to the Bellman-equation for the value-function in a robust MDP (Eq. 4.1), when the uncertainty-set is given by $\mathcal{U}$.

We use this idea in Chapter 6, where our actor-critic algorithm for dynamic Markov coherent risk-measures (which can be dichotomized under inherent uncertainty) explicitly uses our value-function learning method for robust-MDPs, established in Chapter 4 for dealing with model uncertainty.

The importance of this observation is that the coherent-risk framework may be used to handle *both inherent and model uncertainty* (or alternatively, a suitably defined robust-MDP may be used to account for risk-sensitivity).

Interestingly, the link between inherent and model uncertainty is not limited only to dynamic Markov coherent risk-measures and robust MDPs. In a recent work of ours [22], we have shown an equivalence between the static CVaR risk, and a particular robust MDP that has a *coupled* uncertainty structure, and thus cannot be solved using traditional approaches [73, 45]. This further motivates studying the relationship between inherent and model uncertainty, and suggests that possibly, the strict dichotomy that we have followed may in fact be relaxed.

7.2 A Unified View – Risk-Shaping

The unifying idea behind all the works in this dissertation, is that by *modifying the objective* in MDPs, the decision maker may be able to pursue a more principled approach towards uncertainty.

Our goal in this work was to provide the framework and computational tools for solving MDPs with a variety of such risk-sensitive objectives. We believe that our results demonstrate that indeed, risk-sensitive MDPs may be efficiently solved using the various approximation methods we studied. Thus, our results grant the decision-maker great *flexibility* in choosing the risk-sensitive objective.

As our numerical example of Section 6 shows (Figure 6.1), such flexibility is important for selecting appropriate problem-specific risk measures for managing the cost variability.

However, we believe that our approach has much more potential than that. The link established in the previous section between inherent uncertainty and model uncertainty, shows that by carefully designing her risk-objective, the decision-maker can handle uncertainty in a *broad sense*.

Thus, our work suggests that there is potential in *risk-shaping* – the design of problem specific risk measures, which account for both inherent uncertainty and model mis-specification.

While future research is required to investigate how to perform such risk-shaping in a principled manner, the work in this dissertation suggests that once an appropriate shaping has been selected, the resulting optimization problem can be efficiently solved.

7.3 Future Directions

We conclude with several additional directions for future research.

7.3.1 The Exploration-Exploitation Tradeoff

In this dissertation we have not explicitly discussed the issue of how to explore when learning with a risk-sensitive objective. In our policy-gradient approaches, exploration was inherent in the policy definition, and was not explicitly adapted during learning. While in our robust MDP work, we assumed a planning scenario, in which exploration was not needed.

In general, the issue of how to explore in reinforcement learning, and the well-known *exploration-exploitation tradeoff* [100], has been the focus of many studies [48, 19, 46]. In the literature on learning in multi-armed bandits, several studies have considered learning with risk-sensitive objectives [91, 57]. In the risk-sensitive RL literature, however, we are not aware of any work on this topic.

Interestingly, risk-sensitivity may be used as a *method for efficient exploration* in standard MDPs (i.e., MDPs with an expected-return objective). The intuition behind this idea is that when facing an unknown environment, one should behave as *risk-seeking*, and thereby perform actions that might be dangerous, but may also lead to more information about 'interesting' states. This intuition is formally captured in the UCRL2 algorithm [46], in which the exploration policy is determined by solving an optimistic-MDP –

the ‘best-case’ counterpart of robust MDPs. In another study [70], a CVaR-based risk-seeking policy was used to guide exploration during learning. We believe that the results presented in this dissertation may be used to scale-up these previous approaches, which are limited to small-scale MDPs.

7.3.2 Importance Sampling

Our policy-gradient approach for risk-sensitive RL relies on sampling MDP trajectories for obtaining risk-sensitive gradient estimates. As we have discussed in Chapter 5, when the risk-measure is sensitive to rare events, such as in the case of the CVaR risk, our sampling methods require a large number of samples for an effective gradient estimation.

A standard approach for mitigating the effects of high-variance in sampling-based estimates is through importance-sampling [88]. In Chapter 5 we discussed an importance-sampling method for the CVaR risk, and showed that it indeed provides significant improvement in performance. Our approach, however, required the MDP transition probabilities, which in some applications are not known. Designing an effective importance-sampling approach that is model-free, and applies to general risk-measures is an important direction for future research.

7.3.3 Risk-Sensitive Skills

Another interesting direction for future research is in the area of *skill-learning*. A skill, also known as an *option* [103], is a temporally-extended action, or a *sub-policy* for solving a part of the problem. Skills are often used for hierarchical reinforcement learning, where the task is divided into smaller sub-tasks, and skills are learned for each sub-task and then combined for solving the original task.

In the context of risk-aware decision-making, the methods presented in this work may be used ‘off-the-shelf’ to learn risk-sensitive skills, for solving the subgoals in a risk-sensitive style. An interesting question, however, is how to define the risk-sensitive objective for each skill, such that the objective for the *original* task is met with the desired risk preference.

Another potential use for risk-sensitivity in skill-learning is as a means to induce diversity of skills. Currently, there is no principled approach for dividing a task into sub-tasks for learning, and often this procedure is performed using some heuristic [52]. By adding additional risk-sensitive criteria to each sub-task, more skills may be learned, offering *diversity* for the high-level policy in the hierarchy. Such diversity may potentially be useful for learning better policies for the original task.

7.3.4 Value-Based Methods for Static Coherent Risk

In our treatment of coherent risk in Chapter 6, we used a policy-gradient approach for the static coherent risk, and a value-based approach for the dynamic coherent risk.

Currently, it is not known whether a dynamic-programming formulation exists for general static coherent risk, therefore devising a value-based method for this case seems challenging. However, for the case of the static CVaR risk, a recent study [80] showed that a dynamic programming formulation indeed holds on an augmented state-space, and in our recent results [22] we devised an efficient value-based algorithm for static-CVaR based on that formulation.

An interesting question is whether the dynamic programming formulation for static CVaR may be extended to general static coherent risk.

Appendix A

Supplementary Material - Policy Gradients with Variance Related Risk Criteria

This appendix was published as supplementary material in Proceedings of the 29th International Conference on Machine Learning (ICML-12), pages 935–942. Omnipress, 2012.

Supplementary Material

A. Proof of Lemma 3.2

Proof. From Lemma 3.1 (a) we have that:

$$J - P'J = r,$$

therefore

$$\nabla J - \nabla P'J - P'\nabla J = 0,$$

and

$$\nabla J = \left(I - P'\right)^{-1} \nabla P'J.$$

Similarly, we have from Lemma 3.1 (b)

$$V - P'V = \rho,$$

which leads to

$$\nabla V = \left(I - P'\right)^{-1} \left(\nabla \rho + \nabla P'V\right).$$

Finally, from the definition of ρ we have

$$\begin{aligned}\nabla \rho(x) = &\sum_{y \neq x^*} \left(\nabla P(y|x) J^2(y) + 2P(y|x)J(y)\nabla J(y)\right) \\ &- 2\left(\sum_{y \neq x^*} P(y|x)J(y)\right) \sum_{y \neq x^*} \left(\nabla P(y|x)J(y) + P(y|x)\nabla J(y)\right),\end{aligned}$$

which, written in vector form, gives the stated result. □

B. Proof of Lemma 4.2

Proof. Define $\mathcal{T}_T \triangleq \{x_0^T : x_0, \ldots, x_{T-1} \neq x^* \text{ and } x_T = x^*\}$ for $T = 0, 1, 2, \ldots$, and define $\mathcal{T} \triangleq \bigcup_T \mathcal{T}_T$. Then

$$\begin{aligned}\nabla J(k) &= \nabla \sum_{x_0^\tau \in \mathcal{T}} P(x_0^\tau | x_0 = k) R_0^\tau \\ &= \sum_{x_0^\tau \in \mathcal{T}} R_0^\tau \nabla P(x_0^\tau | x_0 = k) \\ &= \sum_{x_0^\tau \in \mathcal{T}} P(x_0^\tau | x_0 = k) R_0^\tau \frac{\nabla P(x_0^\tau | x_0 = k)}{P(x_0^\tau | x_0 = k)} \\ &= \mathbb{E}\left[R_0^\tau \nabla \log\{P(x_0^\tau | x_0 = k)\}\right]\end{aligned} \tag{19}$$

where in the third equality we multiplied and divided by $P(x_0^\tau | x_0 = k)$. By definition (2.1), we have

$$V(k) = \mathbb{E}\left[(R_0^\tau)^2 | x_0 = k\right] - (J(k))^2, \tag{20}$$

therefore

$$\nabla V(k) = \nabla \mathbb{E}\left[(R_0^\tau)^2 | x_0 = k\right] - 2J(k)\nabla J(k). \tag{21}$$

Following the same development above with R_0^τ replaced with $(R_0^\tau)^2$, we have that

$$\nabla \mathbb{E}\left[(R_0^\tau)^2 | x_0 = k\right] = \mathbb{E}\left[(R_0^\tau)^2 \nabla \log\{P(x_0^\tau | x_0 = k)\}\right]. \tag{22}$$

□

C. Parameters for the experiments

We chose parameters that balance between the different factors of the problem, as specified below.

$$\begin{array}{ccc} T = 50, & r_l = 1, & r_{nl}^{high} = 2, \\ r_{nl}^{high} = 1.1, & N = 4, & p_{risk} = 0.05, \\ \epsilon = 0.05, & \alpha = 0.2, & p_{\text{switch}} = 0.1. \end{array}$$

Appendix B

Supplementary Material - Temporal Difference Methods for the Variance of the Reward-To-Go

This appendix was published as supplementary material in Proceedings of the 30th International Conference on Machine Learning (ICML-13), JMLR Workshop and Conference Proceedings 28 (3) :495-503, 2013.

Supplementary Material

A. Proof of Proposition 2

Proof. The equation for $J(x)$ is well-known, and its proof is given here only for completeness. Choose $x \in X$. Then,

$$\begin{aligned}
J(x) &= \mathbb{E}\left[B|x_0 = x\right] \\
&= \mathbb{E}\left[\sum_{k=0}^{\tau-1} r(x_k) \middle| x_0 = x\right] \\
&= r(x) + \mathbb{E}\left[\sum_{k=1}^{\tau-1} r(x_k) \middle| x_0 = x\right] \\
&= r(x) + \mathbb{E}\left[\mathbb{E}\left[\sum_{k=1}^{\tau-1} r(x_k) \middle| x_0 = x, x_1 = x'\right]\right] \\
&= r(x) + \sum_{x' \in X} P(x'|x)J(x')
\end{aligned}$$

where we excluded the terminal state from the sum since reaching it ends the trajectory.

Similarly,

$$\begin{aligned}
M(x) &= \mathbb{E}\left[B^2|x_0 = x\right] \\
&= \mathbb{E}\left[\left(\sum_{k=0}^{\tau-1} r(x_k)\right)^2 \middle| x_0 = x\right] \\
&= \mathbb{E}\left[\left(r(x_0) + \sum_{k=1}^{\tau-1} r(x_k)\right)^2 \middle| x_0 = x\right] \\
&= r(x)^2 + 2r(x)\mathbb{E}\left[\sum_{k=1}^{\tau-1} r(x_k) \middle| x_0 = x\right] + \mathbb{E}\left[\left(\sum_{k=1}^{\tau-1} r(x_k)\right)^2 \middle| x_0 = x\right] \\
&= r(x)^2 + 2r(x)\sum_{x' \in X} P(x'|x)J(x') + \sum_{x' \in X} P(x'|x)M(x').
\end{aligned}$$

The uniqueness of the value function J for a proper policy is well known, c.f. proposition 3.2.1 in (Bertsekas, 2012). The uniqueness of M follows by observing that in the equation for M, M may be seen as the value function of an MDP with the same transitions but with reward $r(x)^2 + 2r(x)\sum_{x' \in X} P(x'|x)J(x')$. Since only the rewards change, the policy remains proper and proposition 3.2.1 in (Bertsekas, 2012) applies. □

B. Proof of Proposition 8

This result is similar to Lemma 6.9 in (Bertsekas & Tsitsiklis, 1996).

Proof. We have

$$\begin{aligned}
\|z_{true} - z^*\|_\alpha &\leq \|z_{true} - \Pi z_{true}\|_\alpha + \|\Pi z_{true} - z^*\|_\alpha \\
&= \|z_{true} - \Pi z_{true}\|_\alpha + \|\Pi T z_{true} - \Pi T z^*\|_\alpha \\
&\leq \|z_{true} - \Pi z_{true}\|_\alpha + \beta\|z_{true} - z^*\|_\alpha.
\end{aligned}$$

rearranging gives the stated result. □

C. Proof of Theorem 9

Proof. Let $\phi_1(x)$, $\phi_2(x)$ be some vector functions of the state. We claim that

$$\mathbb{E}\left[\sum_{t=0}^{\tau-1}\phi_1(x_t)\phi_2(x_t)^\top\right] = \sum_x q(x)\phi_1(x)\phi_2(x)^\top. \tag{18}$$

To see this, let $\mathbb{1}(\cdot)$ denote the indicator function and write

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{\tau-1}\phi_1(x_t)\phi_2(x_t)^\top\right] &= \mathbb{E}\left[\sum_{t=0}^{\tau-1}\sum_x \phi_1(x)\phi_2(x)^\top\mathbb{1}(x_t=x)\right]\\ &= \mathbb{E}\left[\sum_x \phi_1(x)\phi_2(x)^\top\sum_{t=0}^{\tau-1}\mathbb{1}(x_t=x)\right]\\ &= \sum_x \phi_1(x)\phi_2(x)^\top\mathbb{E}\left[\sum_{t=0}^{\tau-1}\mathbb{1}(x_t=x)\right].\end{aligned}$$

Now, note that the last term on the right hand side is an expectation (over all possible trajectories) of the number of visits to a state x until reaching the terminal state, which is exactly $q(x)$ since

$$\begin{aligned}q(x) &= \sum_{t=0}^{\infty}P(x_t=x)\\ &= \sum_{t=0}^{\infty}\mathbb{E}[\mathbb{1}(x_t=x)]\\ &= \mathbb{E}\left[\sum_{t=0}^{\infty}\mathbb{1}(x_t=x)\right]\\ &= \mathbb{E}\left[\sum_{t=0}^{\tau-1}\mathbb{1}(x_t=x)\right],\end{aligned}$$

where the third equality is by the dominated convergence theorem, and last equality follows from the absorbing property of the terminal state. Similarly, we have

$$\mathbb{E}\left[\sum_{t=0}^{\tau-1}\phi_1(x_t)\phi_2(x_{t+1})^\top\right] = \sum_x\sum_{x'} q(x)P(x'|x)\phi_1(x)\phi_2(x')^\top, \tag{19}$$

since

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{\tau-1}\phi_1(x_t)\phi_2(x_{t+1})^\top\right] &= \mathbb{E}\left[\sum_{t=0}^{\tau-1}\sum_x\sum_{x'}\phi_1(x)\phi_2(x')^\top\mathbb{1}(x_t=x,x_{t+1}=x')\right]\\ &= \mathbb{E}\left[\sum_x\sum_{x'}\phi_1(x)\phi_2(x')^\top\sum_{t=0}^{\tau-1}\mathbb{1}(x_t=x,x_{t+1}=x')\right]\\ &= \sum_x\sum_{x'}\phi_1(x)\phi_2(x')^\top\mathbb{E}\left[\sum_{t=0}^{\tau-1}\mathbb{1}(x_t=x,x_{t+1}=x')\right]\end{aligned}$$

and

$$
\begin{aligned}
q(x)P(x'|x) &= \sum_{t=0}^{\infty} P(x_t = x)P(x'|x) \\
&= \sum_{t=0}^{\infty} P(x_t = x, x_{t+1} = x') \\
&= \sum_{t=0}^{\infty} \mathbb{E}[\mathbb{1}(x_t = x, x_{t+1} = x')] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \mathbb{1}(x_t = x, x_{t+1} = x')\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\tau-1} \mathbb{1}(x_t = x, x_{t+1} = x')\right].
\end{aligned}
$$

Since trajectories between visits to the recurrent state are statistically independent, the law of large numbers together with the expressions in (18) and (19) suggest that the approximate expressions in (13) converge to their expected values with probability 1, therefore we have

$$
\begin{aligned}
A_N &\to A, \quad b_N \to b, \\
C_N &\to C, \quad d_N \to D,
\end{aligned}
$$

and

$$
\begin{aligned}
\hat{w}^*_{J;N} &= A_N^{-1} b_N \to A^{-1} b = w^*_J, \\
\hat{w}^*_{M;N} &= C_N^{-1} d_N \to C^{-1} d = w^*_M.
\end{aligned}
$$

□

D. Proof of Theorem 10

Proof. Using (18) and (19) we have for all k

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=0}^{\tau^k-1} \phi_J(x_t)\delta^k_J(t, w_J, w_M)\right] &= \Phi_J^\top Q r - \Phi_J^\top Q\,(I-P)\,\Phi_J w_J, \\
\mathbb{E}\left[\sum_{t=0}^{\tau^k-1} \phi_M(x_t)\delta^k_M(t, w_J, w_M)\right] &= \Phi_M^\top Q R\,(r + 2P\Phi_J w_J) - \Phi_M^\top Q\,(I-P)\,\Phi_M w_M,
\end{aligned}
$$

Letting $\hat{w}_k = (\hat{w}_{J;k}, \hat{w}_{M;k})$ denote a concatenated weight vector in the joint space $\mathbb{R}^l \times \mathbb{R}^m$ we can write the TD algorithm in a stochastic approximation form as

$$
\hat{w}_{k+1} = \hat{w}_k + \xi_k\,(z + M\hat{w}_k + \delta M_{k+1}), \tag{20}
$$

where

$$
M = \begin{pmatrix} \Phi_J^\top Q\,(P-I)\,\Phi_J & 0 \\ 2\Phi_M^\top Q R P \Phi_J & \Phi_M^\top Q\,(P-I)\,\Phi_M \end{pmatrix},
$$

$$
z = \begin{pmatrix} \Phi_J^\top Q r \\ \Phi_M^\top Q R r \end{pmatrix},
$$

and the noise terms δM_{k+1} satisfy

$$
\mathbb{E}\left[\delta M_{k+1} | F_n\right] = 0,
$$

where F_n is the filtration $F_n = \sigma(\hat{w}_m, \delta M_m, m \leq n)$, since different trajectories are independent.

We first claim that the eigenvalues of M have a negative real part. To see this, observe that M is block triangular, and its eigenvalues are just the eigenvalues of $\Phi_J^\top Q\,(P-I)\,\Phi_J$ and $\Phi_M^\top Q\,(P-I)\,\Phi_M$. By Lemma 6.10 in (Bertsekas & Tsitsiklis, 1996) these matrices are negative definite. It therefore follows (see Bertsekas, 2012 example 6.6) that their eigenvalues have a negative real part. Thus, the eigenvalues of M have a negative real part.

Next, let $h(w) = Mw + z$, and observe that the following conditions hold.

A 1. *The map h is Lipschitz.*

A 2. *The step sizes satisfy*

$$\sum_{k=0}^{\infty} \xi_k = \infty, \quad \sum_{k=0}^{\infty} \xi_k^2 < \infty.$$

A 3. *$\{\delta M_n\}$ is a martingale difference sequence, i.e., $\mathbb{E}\left[\delta M_{n+1} | F_n\right] = 0$.*

The next condition also holds

A 4. *The functions $h_c(w) \triangleq h(cw)/c, c \geq 1$ satisfy $h_c(w) \to h_\infty(w)$ as $c \to \infty$, uniformly on compacts, and $h_\infty(w)$ is continuous. Furthermore, the Ordinary Differential Equation (ODE)*

$$\dot{w}(t) = h_\infty(w(t))$$

has the origin as its unique globally asymptotically stable equilibrium.

This is easily verified by noting that $h(cw)/c = Mw + c^{-1}z$, and since z is finite, $h_c(w)$ converges uniformly as $c \to \infty$ to $h_\infty(w) = Mw$. The stability of the origin is guaranteed since the eigenvalues of M have a negative real part.

Theorem 7 in Chapter 3 of (Borkar, 2008) states that if A1 - A4 hold, the following condition holds

A 5. *The iterates of (20) remain bounded almost surely, i.e., $\sup_k \|\hat{w}_k\| < \infty$, a.s.*

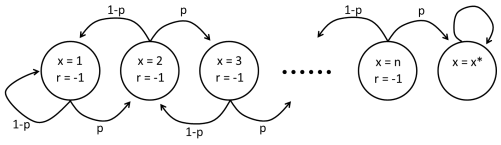
Finally, we use a standard stochastic approximation result that, given that the above conditions hold, relates the convergence of the iterates of (20) with the asymptotic behavior of the ODE

$$\dot{w}(t) = h(w(t)). \tag{21}$$

Since the eigenvalues of M have a negative real part, (21) has a unique globally asymptotically stable equilibrium point, which by (11) is exactly $\hat{w}* = (\hat{w}_J^*, \hat{w}_M^*)$. Formally, by Theorem 2 in Chapter 2 of (Borkar, 2008) we have that if A1 - A3 and A5 hold, then $\hat{w}_k \to \hat{w}*$ as $k \to \infty$ with probability 1. □

E. Illustration of the Positive Variance Constraint

We illustrate the effect of the positive variance constraint in a simple example. Consider the following Markov chain



which consists of N states with reward -1 and a terminal state x^* with zero reward. The transitions from each state is either to a subsequent state (with probability p) or to a preceding state (with probability $1-p$), with the exception of the first state which transitions to itself instead. We chose to approximate J and M with polynomials of degree 1 and 2, respectively. For such a small problem the fixed point equation (15) may be solved exactly, yielding the approximation depicted in Figure 2 (dotted line), for $p = 0.7$, $N = 30$, and $\lambda = 0.95$.

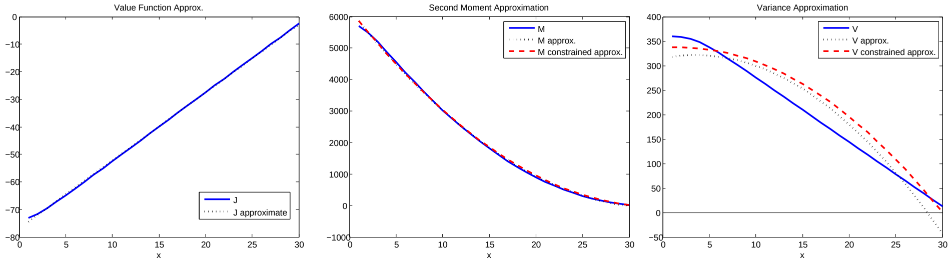


Figure 2. Value, second moment and variance approximation

Note that the variance is negative for the last two states. Using algorithm (17) we obtained a positive variance constrained approximation, which is depicted in figure 2 (dashed line). Note that the variance is now positive for all states (as was required by the constraints).

Appendix C

Supplementary Material - Scaling Up Robust MDPs using Function Approximation

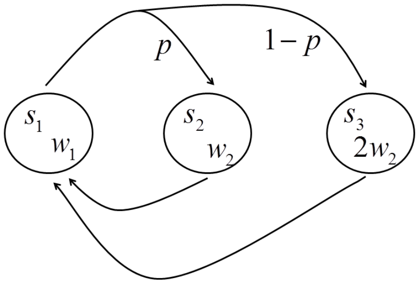
This appendix was published as supplementary material in Proceedings of the 31th International Conference on Machine Learning (ICML-14), JMLR Workshop and Conference Proceedings 32 :181–189, 2014.

Scaling Up Robust MDPs by Function Approximation - Supplementary Material

1 A Divergent Example

We show that even if Assumption 2 fails for a single state $\tilde{x}$, and for that state there is no approximation - i.e., there is a feature $\tilde{\phi}(x) = \mathbb{1}\{x = \tilde{x}\}$ that is orthogonal to all other features, iteratively applying ΠT^{π} may diverge.

Consider the following MDP with 3 states $\{s1, s2, s3\}$, zero rewards, and let the value function approximation be $(w_1, w_2, 2w_2)^T$.



The Bellman operator for some $v = (w_1, w_2, 2w_2)^T$ is

$$T^{\pi}v = \gamma \left(\begin{array}{c} pw_2 + (1-p)\,2w_2 \\ w_1 \\ w_1 \end{array} \right)$$

Consider an exploration policy ($\hat{P}$) where $p = 0.5$, and therefore the steady state distribution of $s2$ and $s3$ are equal. Note that the only transition change (between the exploration policy and the true MDP) is in s_1, for which there is no approximation in the value function. The least squares regression of a vector (x_1, x_2) onto $(w_2, 2w_2)$ gives $w_2 = \frac{1}{5}(x_1 + 2x_2)$, and therefore the projected Bellman operator is

$$\Pi T^{\pi}v = \gamma \left(\begin{array}{c} 2w_2 - pw_2 \\ \frac{3}{5}w_1 \\ \frac{6}{5}w_1 \end{array} \right),$$

and in terms of w, we can write the result of applying ΠT^{π} as w', and we have

$$w' = \gamma \begin{pmatrix} 0 & (2-p) \\ \frac{3}{5} & 0 \end{pmatrix} w.$$

The eigenvalues of the above matrix are $\pm\gamma\frac{\sqrt{15(2-p)}}{5}$, and we have that for $p < 2 - \frac{5}{3\gamma^2}$ (For example $p = 0.1, \gamma = 0.95$) the eigenvalues are outside the unit circle and the process of repeatedly applying ΠT^π diverges.

2 Complexity of Solving the Inner Problem using SAA

We state a result of Shapiro & Nemirovski (2005) that bounds the sample complexity of SAA. Recall that we approximate the solution of the problem

$$\inf_{\theta\in\Theta} \mathbb{E}_{\tilde{p}}\left[\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k\right], \tag{1}$$

using the solution of the SAA

$$\inf_{\theta\in\Theta} \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{p_\theta(x_i)}{\tilde{p}(x_i)}\phi(x_i)^\top w_k. \tag{2}$$

Assume that $\mathbb{E}_{\tilde{p}}\left[\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k\right]$ is convex in θ, and that $\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k$ is Lipschitz continuous on Θ with constant L independent of x. Let $D \doteq \sup_{\theta,\theta'\in\Theta}\|\theta'-\theta\|$ denote the diameter of Θ. Then the following sample complexity result holds.

Theorem 1. *(Theorem 2 in Shapiro & Nemirovski 2005) For a sample size N_s that satisfies*

$$N_s \geq \mathcal{O}(1)\left(\frac{DL}{\epsilon}\right)^2\left[n\log\left(\frac{DL}{\epsilon}\right) + \log\left(\frac{\mathcal{O}(1)}{\alpha}\right)\right]$$

we are guaranteed that every $(\epsilon/2)$-optimal solution of the SAA problem (2) *is an ϵ-optimal solution of the true problem* (1) *with probability* $1-\alpha$.

Theorem 1 bounds the number of samples required for constructing the SAA approximation (2). However, one still needs to solve the SAA problem. When $\mathbb{E}_{\tilde{p}}\left[\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k\right]$ is convex and twice continuously differentiable, the SAA may be solved efficiently using, e.g., interior point methods (Boyd & Vandenberghe, 2004).

3 Proof of Proposition 6

Proposition 2. *The sequence $\{\pi_i\}$ generated by the general approximate robust policy iteration algorithm satisfies*

$$\limsup_{i\to\infty}\|V^{\pi_i} - V^*\|_\infty \leq \frac{\epsilon + 2\gamma\delta}{(1-\gamma)^2}.$$

Proof. The proof of Proposition 2.5.8 of Bertsekas (2012) holds provided that the operators T^π and T are both γ-contractions in the sup-norm and monotone. The contraction property was shown by Iyengar (2005). We now show the monotonicity.

Choose some policy π and $\epsilon' > 0$. Let $v, v' \in \mathbb{R}^{|\mathcal{X}|}$ satisfy $v(x) \leq v'(x)$ for all x. Also let $\bar{p}_x \in \mathcal{P}(x, \pi(x))$ such that $\bar{p}_x^\top v' \leq \inf_{p \in \mathcal{P}(x,\pi(x))} p^\top v' + \epsilon'$. We have that for all x

$$\inf_{p \in \mathcal{P}(x,\pi(x))} p^\top v' + \epsilon' \geq \bar{p}_x^\top v' \geq \bar{p}_x^\top v \geq \inf_{p \in \mathcal{P}(x,\pi(x))} p^\top v,$$

where the second inequality holds since by definition $\bar{p}_x \geq 0$. Since ϵ' was arbitrary we conclude that $\inf_{p \in \mathcal{P}(x,\pi(x))} p^\top v \leq \inf_{p \in \mathcal{P}(x,\pi(x))} p^\top v'$, therefore $T^\pi v \leq T^\pi v'$, which proves the monotonicity of T^π. Since this holds for every π, it holds also for T. □

4 Optimal Stopping Problems

Consider the optimal stopping problem setting, and let $\hat{\pi}$ denote a policy that never chooses to terminate, i.e., $\hat{\pi}(x) = 0$, $\forall x$. We now show that if Assumption 2 is satisfied for $\pi = \hat{\pi}$, then it is immediately satisfied for all other policies.

Proposition 3. *Consider an optimal stopping problem, and let Assumption 2 hold for $\pi = \hat{\pi}$. Then, for every policy π we have*

$$\gamma P(x'|x, \pi(x)) \leq \beta \hat{P}(x'|x, \hat{\pi}(x)), \quad \forall P \in \mathcal{P}, x \in \mathcal{X}, x' \in \mathcal{X}. \tag{3}$$

Proof. We prove by induction. Assume (3) holds for some π. Let $\tilde{\pi}$ be the same as π for all states except $\tilde{x}$, for which $\pi(\tilde{x}) = 0$ and $\tilde{\pi}(\tilde{x}) = 1$. Then we have for all $x \neq \tilde{x}$ that $P(x'|x, \tilde{\pi}(x)) = P(x'|x, \pi(x)), \quad \forall P \in \mathcal{P}$. For $\tilde{x}$, a transition to a terminal state occurs without uncertainty, namely $P(x'|\tilde{x}, \tilde{\pi}(\tilde{x})) = 0 \quad \forall P \in \mathcal{P}, x' \in \mathcal{X}$, therefore (3) is satisfied with π replaced by $\tilde{\pi}$.

Since (3) is assumed to hold for $\hat{\pi}$, by induction it holds for all π. □

5 Optimistic MDPs

Interestingly, as was recognized by Iyengar (2005), results on robust MDPs may be extended to optimistic MDPs. An optimistic MDP is similar to an RMDP, but the optimization goal is different. Here, instead of the worst case performance, we seek the most optimistic value $V^+(x) = \sup_\pi \left\{ \sup_{P \in \mathcal{P}} V^{\pi,P}(x) \right\}$. In addition to obtaining risk-seeking policies, optimistic MDPs have been used for efficient exploration, driving algorithms such as UCRL2 Jaksch et al. (2010) by employing the principle of 'optimism in the face of uncertainty'. Our work may be important for large-scale implementations of such algorithms, by use of function approximation. We also conjecture that the performance gap due to uncertainty $V^+(x) - V(x)$ may be important for feature selection and model selection, tasks that are critical for truly large-scale applications.

For some x and u let us define the operator $\sigma^+_{\mathcal{P}(x,u)} : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}$ as (cf. the definition of $\sigma_{\mathcal{P}(x,u)}$)

$$\sigma^+_{\mathcal{P}(x,u)} v \doteq \sup \left\{ p^\top v : p \in \mathcal{P}(x,u)] \right\}.$$

All our results extend to optimistic MDPs, namely, by replacing the operator $\sigma_{\mathcal{P}(x,u)}$ with $\sigma^+_{\mathcal{P}(x,u)}$. We now show this for the proof of Proposition 3.

Proof. Fix $x \in \mathcal{X}$, and assume that $T^\pi y(x) \leq T^\pi z(x)$. Choose some $\epsilon > 0$, and $P_x \in \mathcal{P}$ such that

$$\mathbb{E}^{P_x} \left[z(x') | \, x, \pi(x) \right] \geq \sup_{P \in \mathcal{P}} \mathbb{E}^P \left[z(x') | \, x, \pi(x) \right] - \epsilon. \tag{4}$$

Also, note that by definition

$$\sup_{P\in\mathcal{P}} \mathbb{E}^{P}\left[y(x') \middle| x, \pi(x)\right] \geq \mathbb{E}^{P_x}\left[y(x') \middle| x, \pi(x)\right]. \tag{5}$$

Now, we have

$$\begin{aligned} 0 &\leq T^{\pi}z(x) - T^{\pi}y(x) \\ &\leq \left(\gamma\mathbb{E}^{P_x}\left[z(x') \middle| x, \pi(x)\right] + \gamma\epsilon\right) - \left(\gamma\mathbb{E}^{P_x}\left[y(x') \middle| x, \pi(x)\right]\right) \\ &= \gamma\mathbb{E}^{P_x}\left[z(x') - y(x') \middle| x, \pi(x)\right] + \gamma\epsilon \\ &\leq \beta\mathbb{E}^{\hat{P}}\left[\left|z(x') - y(x')\right| \middle| x, \pi(x)\right] + \gamma\epsilon, \end{aligned}$$

where the second inequality is by (4) and (5), and the last inequality is by Assumption 2. Conversely, if $T^{\pi}y(x) \geq T^{\pi}z(x)$, following the same procedure we obtain $0 \leq T^{\pi}y(x) - T^{\pi}z(x) \leq \beta\mathbb{E}^{\hat{P}}\left[\left|z(x') - y(x')\right| \middle| x, \pi(x)\right] + \gamma\epsilon$, and we therefore conclude that $\left|T^{\pi}y(x) - T^{\pi}z(x)\right| \leq \beta\mathbb{E}^{\hat{P}}\left[\left|y(x') - z(x')\right| \middle| x, \pi(x)\right] + \gamma\epsilon$. Since ϵ was arbitrary, we have that $\left|T^{\pi}y(x) - T^{\pi}z(x)\right| \leq \beta\mathbb{E}^{\hat{P}}\left[\left|y(x') - z(x')\right| \middle| x, \pi(x)\right]$ for all x, and therefore

$$\left\| T^{\pi}y - T^{\pi}z \right\|_d \leq \beta \left\| \hat{P}\left| y - z\right| \right\|_d \leq \beta \left\| y - z \right\|_d,$$

where in last equality we used the well-known result that the state transition matrix $\hat{P}$ is contracting in the d-weighted Euclidean norm. □

We now show that Proposition 6 also holds.

Proof. We need to show the monotonicity property.

Choose some policy π and $\epsilon' > 0$. Let $v, v' \in \mathbb{R}^{|\mathcal{X}|}$ satisfy $v(x) \geq v'(x)$ for all x. Also let $\bar{p}_x \in \mathcal{P}(x, \pi(x))$ such that $\bar{p}_x^{\top} v' \geq \sup_{p\in\mathcal{P}(x,\pi(x))} p^{\top} v' - \epsilon'$. We have that for all x

$$\sup_{p\in\mathcal{P}(x,\pi(x))} p^{\top} v' - \epsilon' \leq \bar{p}_x^{\top} v' \leq \bar{p}_x^{\top} v \leq \sup_{p\in\mathcal{P}(x,\pi(x))} p^{\top} v,$$

where the second inequality holds since by definition $\bar{p}_x \geq 0$. Since ϵ' was arbitrary we conclude that $\sup_{p\in\mathcal{P}(x,\pi(x))} p^{\top} v \geq \sup_{p\in\mathcal{P}(x,\pi(x))} p^{\top} v'$, therefore $T^{\pi}v \geq T^{\pi}v'$, which proves the monotonicity of T^{π}. Since this holds for every π, it holds also for T. □

6 Parameters for Option Trading Experiments

The parameters for the experiments in Section 6 were chosen to balance the different factors in the problem. Specifically, we chose

Experiment	$Kput$	$Kcall$	T	γ	f_u	f_d	p	p_1	p_2	N_{data}	N_{sim}	δ	x_0	N_{test}
Put option	1	1.5	20	0.98	9/8	8/9	0.45			5	500	1	1	50,000
Put and call	1	1.5	20	0.98	9/8	8/9	0.45			3	500	1	1.25	50,000
Model misspecification	1	1.5	20	0.98	9/8	8/9		0.3	0.6	3	500	1	1.25	50,000

We used 2-Dimensional Gaussian RBF features with a uniform spacing $\Delta x = 0.4$, $\Delta t = 6$, and widths $\sigma_x = 0.235$ and $\sigma_t = 3.535$. The outputs of the RBFs were normalized.

7 Experiments with a Geometric Brownian Motion Model

In this section we consider the option trading domain of Section 6, where the price model follows a Geometric Brownian Motion (GBM), a popular model for stock price fluctuations. In continuous time, a GBM obeys the stochastic differential equation $dx_t = \mu x_t dt + \sigma x_t dW_t$, where μ is the risk free interest rate, σ is the stock volatility, and W is a standard Brownian motion. In discrete time, a GBM trajectory may be simulated by $x_{t+\Delta t} = x_t \exp\left\{(\mu - \sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}\omega\right\}$, where $\omega \sim \mathcal{N}(0,1)$. Thus, x_{t+1}/x_t has a lognormal distribution

$$\frac{x_{t+\Delta t}}{x_t} \sim \ln\mathcal{N}\left(\Delta t(\mu - \sigma^2/2), \sigma^2\Delta t\right).$$

In practice, the volatility is not known, but estimated from data. Thus, we construct the uncertainty set as the 95% confidence intervals for the estimated volatility. Our empirical evaluation proceeds as follows. In each experiment, we generate N_{data} trajectories of length T from the true model M with parameters μ and σ where μ is the risk-free interest rate. From these trajectories we estimate the volatility $\hat{\sigma}$, and the 95% confidence intervals $\hat{\sigma}_-$ and $\hat{\sigma}_+$ using the Matlab function `lognfit`, which constructs our uncertain model M_{robust}. We also build a model without uncertainty $M_{nominal}$ by setting $\hat{\sigma}_- = \hat{\sigma}_+ = \hat{\sigma}$. Using $\hat{\sigma}$, we then simulate N_{sim} trajectories of length T (this corresponds to a policy that never executes the option), where $x_0 = K + \epsilon$, and ϵ is uniformly distributed in $[-\delta, \delta]$. These trajectories are used as input data for the ARPI algorithm of Section 4. For solving the inner problem, we use the SAA method of Section 3.4 with N_s samples, where we set $\tilde{p}$ to the lognormal distribution corresponding to $\hat{\sigma}_+$. The deterministic sampled problem was solved using Matlab's `fminbnd` method.

In Figure 1 we plot the tail distribution of the total reward R (from 20 independent experiments) obtained by π_{robust} and $\pi_{nominal}$ for the put option scenario (cf. Section 6.2.2 of the main text). The results are similar to the case of the Bernoulli price fluctuation model. These results confirm that our method scales robust MDPs to truly large scale domains.

The parameters for this experiment were chosen to balance the different factors in the problem. Specifically, we chose

$Kput$	T	μ	σ	Δt	γ	N_s	N_{data}	N_{sim}	δ	x_0	N_{test}
1	20	0.0025	3	0.01	0.9975	50	5	500	1	1	50,000

The RBFs were the same as in the experiments in the main text.

References

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol II.* Athena Scientific, fourth edition, 2012.

Boyd, S. P. and Vandenberghe, L. *Convex optimization.* Cambridge Univ Pr, 2004.

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Shapiro, A. and Nemirovski, A. On complexity of stochastic programming problems. In *Continuous optimization*, pp. 111–146. Springer, 2005.

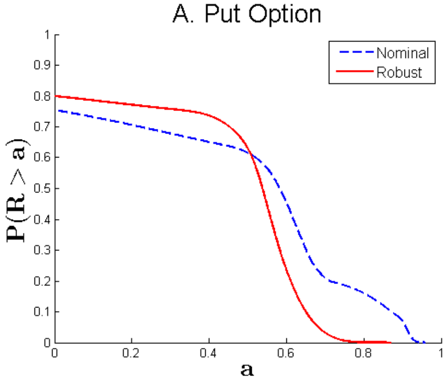


Figure 1: Performance of robust vs. nominal policies. The tail distribution (complementary cumulative distribution function) of the total reward R for the put option scenario, obtained from 20 independent experiments.

Appendix D

Supplementary Material - Optimizing the CVaR via Sampling

This appendix contains supplementary material for the paper published in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), pages 2993–2999. AAAI Press, 2015.

A Proof of Proposition 2

Proof. The main difficulty in extending the proof of Proposition 1 to this case is in applying the Leibnitz rule in a multi-dimensional case. Such an extension is given by (Flanders 1973), which we now state.

We are given an $n-$dimensional $\theta-$dependent chain (field of integration) D_θ in $\mathbb{R}^n$. We also have an exterior differential $n-$form whose coefficients are θ-dependent:

$$\omega = f(\mathbf{x}, \theta) dx_1 \wedge \cdots \wedge dx_n$$

The general Leibnitz rule[5] is given by

$$\frac{\partial}{\partial \theta} \int_{D_\theta} \omega = \int_{\partial D_\theta} \mathbf{v} \lrcorner \omega + \int_{D_\theta} \frac{\partial \omega}{\partial \theta} \tag{12}$$

where $\mathbf{v}$ denotes the vector field of velocities $\frac{\partial}{\partial \theta}\mathbf{x}$ of D_θ, and $\mathbf{v} \lrcorner \omega$ denotes the interior product between $\mathbf{v}$ and ω (see (Flanders 1973) for more details).

We now write the CVaR explicitly as

$$\begin{aligned}\Phi_\alpha(R;\theta) &= \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \int_{\mathbf{x} \in \mathcal{D}_{y;\theta}} f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) r(\mathbf{x},y) d\mathbf{x} \\ &= \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{\mathbf{x} \in D^i_{y;\theta}} f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) r(\mathbf{x},y) d\mathbf{x},\end{aligned}$$

therefore

$$\begin{aligned}\frac{\partial}{\partial \theta_j} \Phi_\alpha(R;\theta) = &\frac{1}{\alpha} \sum_{y \in \mathcal{Y}} \frac{\partial f_Y(y;\theta)}{\partial \theta_j} \sum_{i=1}^{L_{y;\theta}} \int_{\mathbf{x} \in D^i_{y;\theta}} f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) r(\mathbf{x},y) d\mathbf{x} \\ &+ \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \frac{\partial}{\partial \theta_j} \int_{\mathbf{x} \in D^i_{y;\theta}} f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) r(\mathbf{x},y) d\mathbf{x}\end{aligned} \tag{13}$$

We now treat each $D^i_{y;\theta}$ in the last sum separately. Let $\mathcal{X}$ denote the set $[-b,b]^n$ over which $\mathbf{X}$ is defined. Obviously, $D^i_{y;\theta} \subset \mathcal{X}$.

We now make an important observation. By definition of the level-set $D^i_{y;\theta}$, and since it is closed by Assumption 5, for every $\mathbf{x} \in \partial D^i_{y;\theta}$ we have that either

$$\text{(a) } r(\mathbf{x},y) = \nu_\alpha(R;\theta), \tag{14}$$

or

$$\text{(b) } \mathbf{x} \in \partial \mathcal{X}, \text{ and } r(\mathbf{x},y) < \nu_\alpha(R;\theta). \tag{15}$$

We write $\partial D^i_{y;\theta} = \partial D^{i,a}_{y;\theta} + \partial D^{i,b}_{y;\theta}$ where the two last terms correspond to the two possibilities in (14) and (15).

We now claim that for the boundary term $\partial D^{i,b}_{y;\theta}$, we have

$$\int_{\partial D^{i,b}_{y;\theta}} \mathbf{v} \lrcorner \omega = 0. \tag{16}$$

To see this, first note that by definition of $\mathcal{X}$, the boundary $\partial \mathcal{X}$ is smooth and has a unique normal vector at each point, except for a set of measure zero (the corners of $\mathcal{X}$). Let $\partial \tilde{D}^{i,b}_{y;\theta}$ denote the set of all points in $\partial D^{i,b}_{y;\theta}$ for which a unique normal vector exists. For each $\mathbf{x} \in \partial \tilde{D}^{i,b}_{y;\theta}$ we let $\mathbf{v}_\perp$ and $\mathbf{v}_\parallel$ denote the normal and tangent (with respect to $\partial \mathcal{X}$) elements of the velocity $\frac{\partial}{\partial \theta}\mathbf{x}$ at $\mathbf{x}$, respectively. Thus,

$$\mathbf{v} = \mathbf{v}_\perp + \mathbf{v}_\parallel.$$

For some $\epsilon > 0$ let d_ϵ denote the set $\left\{ \mathbf{x} \in \partial D^{i,b}_{y;\theta} : r(\mathbf{x},y) < \nu_\alpha(R;\theta) - \epsilon \right\}$. From Assumption 4 we have that $\frac{\partial}{\partial \theta_j}\nu_\alpha(R;\theta)$ is bounded, therefore there exists $\delta(\epsilon) > 0$ such that for all θ' that satisfy $\|\theta - \theta'\| < \delta(\epsilon)$ we have $|\nu_\alpha(R;\theta') - \nu_\alpha(R;\theta)| < \epsilon$, and therefore $d_\epsilon \in \partial D^{i,b}_{y;\theta'}$. Since this holds for every $\epsilon > 0$, we conclude that a small change in θ does not change $\partial D^{i,b}_{y;\theta}$, and therefore we have

$$\mathbf{v}_\perp = 0, \quad \forall \mathbf{x} \in \partial \tilde{D}^{i,b}_{y;\theta}.$$

Furthermore, by definition of the interior product we have

$$\mathbf{v}_\parallel \lrcorner \omega = 0.$$

[5] The formula in (Flanders 1973) is for a more general case where D_θ is not necessarily $n-$dimensional. That formula includes an additional term $\int_{D_\theta} \mathbf{v} \lrcorner d_{\mathbf{x}} \omega$, where $d_{\mathbf{x}}$ is the exterior derivative, which cancels in our case.

Therefore we have

$$\int_{\partial D^{i,b}_{y;\theta}} \mathbf{v} \lrcorner \omega = \int_{\partial \tilde{D}^{i,b}_{y;\theta}} \mathbf{v} \lrcorner \omega = \int_{\partial \tilde{D}^{i,b}_{y;\theta}} \mathbf{v}_{\parallel} \lrcorner \omega = 0,$$

and the claim follows.

Now, let $\omega_y = f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) r(\mathbf{x},y) dx_1 \wedge \cdots \wedge dx_n$. Using (12), we have

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \int_{\mathbf{x} \in D^i_{y;\theta}} \omega_y &= \int_{\partial D^i_{y;\theta}} \mathbf{v} \lrcorner \omega_y + \int_{D^i_{y;\theta}} \frac{\partial \omega_y}{\partial \theta} \\ &= \int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \omega_y + \int_{D^i_{y;\theta}} \frac{\partial \omega_y}{\partial \theta} \end{aligned} \tag{17}$$

where the last equality follows from (16) and the definition of $\mathbf{v}$.

Let $\tilde{\omega}_y = f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta) dx_1 \wedge \cdots \wedge dx_n$. By the definition of $\mathcal{D}_{y;\theta}$ we have that for all θ

$$\alpha = \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \int_{\mathcal{D}_{y;\theta}} \tilde{\omega}_y,$$

therefore, by taking a derivative, and using (16) we have

$$\begin{aligned} 0 = \frac{\partial}{\partial \theta_j} \left(\sum_{y \in \mathcal{Y}} f_Y(y;\theta) \int_{\mathcal{D}_{y;\theta}} \tilde{\omega}_y \right) &= \sum_{y \in \mathcal{Y}} \frac{\partial f_Y(y;\theta)}{\partial \theta_j} \int_{\mathcal{D}_{y;\theta}} \tilde{\omega}_y \\ &+ \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \left(\int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \tilde{\omega}_y + \int_{D^i_{y;\theta}} \frac{\partial \tilde{\omega}_y}{\partial \theta} \right) \end{aligned} \tag{18}$$

From (14), and linearity of the interior product we have

$$\int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \omega_y = \nu_\alpha(R;\theta) \int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \tilde{\omega}_y,$$

therefore, plugging in (18) we have

$$\begin{aligned} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \omega_y &= - \nu_\alpha(R;\theta) \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{D^i_{y;\theta}} \frac{\partial \tilde{\omega}_y}{\partial \theta} \\ &- \nu_\alpha(R;\theta) \sum_{y \in \mathcal{Y}} \frac{\partial f_Y(y;\theta)}{\partial \theta_j} \int_{\mathcal{D}_{y;\theta}} \tilde{\omega}_y \end{aligned} \tag{19}$$

Now, note that from (13) and (17) we have

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \Phi_\alpha(R;\theta) =& \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} \frac{\partial f_Y(y;\theta)}{\partial \theta_j} \sum_{i=1}^{L_{y;\theta}} \int_{\mathbf{x} \in D^i_{y;\theta}} \omega_y \\ &+ \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{D^i_{y;\theta}} \frac{\partial \omega_y}{\partial \theta} \\ &+ \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{\partial D^{i,a}_{y;\theta}} \mathbf{v} \lrcorner \omega_y, \end{aligned}$$

and by plugging in (19) we obtain

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \Phi_\alpha(R;\theta) =& \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} \frac{\partial f_Y(y;\theta)}{\partial \theta_j} \sum_{i=1}^{L_{y;\theta}} \int_{D^i_{y;\theta}} \omega_y - \nu_\alpha(R;\theta) \tilde{\omega}_y \\ &+ \frac{1}{\alpha} \sum_{y \in \mathcal{Y}} f_Y(y;\theta) \sum_{i=1}^{L_{y;\theta}} \int_{D^i_{y;\theta}} \frac{\partial \omega_y}{\partial \theta} - \nu_\alpha(R;\theta) \frac{\partial \tilde{\omega}_y}{\partial \theta}. \end{aligned}$$

Finally, using the standard likelihood ratio trick – multiplying and dividing by $f_Y(y;\theta)$ inside the first sum, and multiplying and dividing by $f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)$ inside the second integral we obtain the required expectation. □

B Proof of Theorem 3

Proof. Let $\nu = \nu_\alpha(R;\theta)$. To simplify notation, we also introduce the functions $h_1(\mathbf{x},y) \doteq \left(\frac{\partial \log f_Y(y;\theta)}{\partial\theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)}{\partial\theta_j}\right) r(\mathbf{x},y)$, and $h_2(\mathbf{x},y) \doteq \left(\frac{\partial \log f_Y(y;\theta)}{\partial\theta_j} + \frac{\partial \log f_{\mathbf{X}|Y}(\mathbf{x}|y;\theta)}{\partial\theta_j}\right)$. Thus we have

$$\begin{aligned}
\Delta_{j;N} =& \frac{1}{\alpha N}\sum_{i=1}^{N}\left(h_1(\mathbf{x}_i,y_i) - h_2(\mathbf{x}_i,y_i)\tilde{v}\right)\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} \\
=& \frac{1}{\alpha N}\sum_{i=1}^{N}\left(h_1(\mathbf{x}_i,y_i) - h_2(\mathbf{x}_i,y_i)\nu\right)\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu} \\
&+ \frac{1}{\alpha N}\sum_{i=1}^{N}\left(h_1(\mathbf{x}_i,y_i) - h_2(\mathbf{x}_i,y_i)\nu\right)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right) \\
&+ (\nu - \tilde{v})\frac{1}{\alpha N}\sum_{i=1}^{N} h_2(\mathbf{x}_i,y_i)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}}\right)
\end{aligned} \tag{20}$$

We furthermore let $D(\mathbf{x},y) \doteq h_1(\mathbf{x},y) - h_2(\mathbf{x},y)\nu$. Note that by Assumption 4, D is bounded.

By Proposition 2, and the strong law of large numbers, we have that w.p. 1

$$\frac{1}{\alpha N}\sum_{i=1}^{N}\left(h_1(\mathbf{x}_i,y_i) - h_2(\mathbf{x}_i,y_i)\nu\right)\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu} \to \frac{\partial}{\partial\theta_j}\Phi_\alpha(R;\theta). \tag{21}$$

We now show that the two additional terms in (20) vanish as $N \to \infty$. By Hölder's inequality

$$\left|\frac{1}{N}\sum_{i=1}^{N} D(\mathbf{x}_i,y_i)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right)\right| \le \left(\frac{1}{N}\sum_{i=1}^{N}\left|D(\mathbf{x}_i,y_i)\right|^2\right)^{0.5}\cdot\left(\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right|^2\right)^{0.5}, \tag{22}$$

and $\left(\frac{1}{N}\sum_{i=1}^{N}\left|D(\mathbf{x}_i,y_i)\right|^2\right)^{0.5}$ is bounded. Also, note that

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right|^2 &= \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right| \\
&= \left(\mathbf{1}_{\tilde{v}\le\nu} - \mathbf{1}_{\nu\le\tilde{v}}\right)\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right)
\end{aligned}$$

By Proposition 4.1 of (Hong and Liu 2009), we have that w.p. 1

$$\left(\mathbf{1}_{\tilde{v}\le\nu} - \mathbf{1}_{\nu\le\tilde{v}}\right)\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right) \to 0.$$

By the continuous mapping theorem, we thus have that w.p. 1

$$\left(\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right|^2\right)^{0.5} \to 0,$$

therefore, using Eq. (22) we have that w.p. 1

$$\frac{1}{N}\sum_{i=1}^{N} D(\mathbf{x}_i,y_i)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}} - \mathbf{1}_{r(\mathbf{x}_i,y_i)\le\nu}\right) \to 0. \tag{23}$$

We now turn to the last sum in (20). by Assumption 4, h_2 is bounded, and therefore $\frac{1}{\alpha N}\sum_{i=1}^{N} h_2(\mathbf{x}_i,y_i)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}}\right)$ is bounded. It is well-known (David 1981) that the sample-quantile is a consistent estimator, thus $\nu - \tilde{v} \to 0$, and therefore

$$(\nu - \tilde{v})\frac{1}{\alpha N}\sum_{i=1}^{N} h_2(\mathbf{x}_i,y_i)\left(\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}}\right) \to 0. \tag{24}$$

Plugging (21), (23), and (24) in (20) gives the stated result. □

C Proof of Theorem 4

We follow the notation of Section B.

In our analysis we use a result of (Hong and Liu 2009), which we now state. Let $\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N$ be N samples drawn i.i.d. from $f_{\mathbf{X},Y}(\mathbf{x}, y; \theta)$.

Theorem 6. *(Theorem 4.2 of (Hong and Liu 2009)) Let Assumption 6, and the assumptions required for Proposition 2 hold. Let*

$$\bar{\Delta}_N = \frac{1}{\alpha N} \sum_{i=1}^{N} D(\mathbf{x}_i, y_i) \cdot \mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}.$$

Then $\mathbb{E}\left[\bar{\Delta}_N\right] - \frac{\partial}{\partial \theta_j} \Phi_\alpha(R; \theta)$ *is* $o(N^{-1/2})$.

In the original theorem of (Hong and Liu 2009), D is defined differently, corresponding to the perturbation analysis type gradient estimator. However, the proof of the theorem follows through also with our definition of D, and using Proposition 2.

We are now ready to prove Theorem 4.

Proof. From Eq. (20) we have

$$\begin{aligned}\mathbb{E}\left[\Delta_{j;N}\right] - \frac{\partial}{\partial \theta_j} \Phi_\alpha(R; \theta) =& \mathbb{E}\left[\frac{1}{\alpha N} \sum_{i=1}^{N} D(\mathbf{x}_i, y_i) \cdot \mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}\right] \\ &+ \mathbb{E}\left[(\nu - \tilde{v}) \frac{1}{\alpha N} \sum_{i=1}^{N} h_2(\mathbf{x}_i, y_i) \left(\mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}\right)\right].\end{aligned} \tag{25}$$

The first term in the r.h.s. of Eq. (25) is $o(N^{-1/2})$ by Theorem 6. We now bound the second term.

Let $\bar{h}_2$ denote a bound on h_2, which, by Assumption 4, is finite. Note that we have $\left|\frac{1}{N} \sum_{i=1}^{N} h_2(\mathbf{x}_i, y_i) \left(\mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}\right)\right| \le \bar{h}_2$ with probability 1. Therefore,

$$\mathbb{E}\left[(\nu - \tilde{v}) \frac{1}{\alpha N} \sum_{i=1}^{N} h_2(\mathbf{x}_i, y_i) \left(\mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}\right)\right] \le \mathbb{E}\left[|\nu - \tilde{v}| \left|\frac{1}{\alpha N} \sum_{i=1}^{N} h_2(\mathbf{x}_i, y_i) \left(\mathbf{1}_{r(\mathbf{x}_i, y_i) \le \tilde{v}}\right)\right|\right] \le \frac{\bar{h}_2}{\alpha} \mathbb{E}\left[|\nu - \tilde{v}|\right].$$

We will show that $\mathbb{E}\left[|\nu - \tilde{v}|\right]$ is $O(N^{-1/2})$. It is well-known (David 1981) that the empirical $\alpha-$quantile may be written as follows:

$$\tilde{v} = \nu - \frac{\hat{F}_R(\nu) - \alpha}{f_R(\nu)} + \tilde{R}, \tag{26}$$

where $\hat{F}_R(\cdot)$ is the empirical C.D.F. of R, and $\tilde{R}$ is $O(N^{-1/2})$ in probability. Thus, we have

$$\mathbb{E}\left[|\nu - \tilde{v}|\right] \le f_R(\nu)^{-1} \left(\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|\right] + \mathbb{E}\left[\left|\tilde{R}\right|\right]\right). \tag{27}$$

Note that since R is bounded, $\tilde{v}$ is also bounded, and it is clear from Eq. (26) that $\tilde{R}$ is bounded, and therefore uniformly integrable. Since $\tilde{R}$ is also $O(N^{-1/2})$ in probability, we conclude that $\mathbb{E}\left[\left|\tilde{R}\right|\right]$ is $O(N^{-1/2})$. Let $y_i \doteq \mathbf{1}_{r(\mathbf{x}_i, y_i) \le \nu}$. Then by definition, the empirical C.D.F. satisfies

$$\hat{F}_R(\nu) = \frac{1}{N} \sum_{i=1}^{N} y_i,$$

and the y_i's are i.i.d., and satisfy $\mathbb{E}[y_i] = \alpha$, and $\mathrm{Var}[y_i] = \alpha(1 - \alpha)$. Observe that

$$0 \le \mathrm{Var}\left[\left|\hat{F}_R(\nu) - \alpha\right|\right] = \mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|^2\right] - \left(\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|\right]\right)^2,$$

therefore

$$\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|\right] \le \sqrt{\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|^2\right]},$$

but

$$\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|^2\right] = \mathrm{Var}\left[\hat{F}_R(\nu)\right] = \frac{\alpha(1 - \alpha)}{N},$$

therefore $\mathbb{E}\left[\left|\hat{F}_R(\nu) - \alpha\right|\right]$ is $O(N^{-1/2})$. From Eq. (27) we thus have that $\mathbb{E}\left[|\nu - \tilde{v}|\right]$ is $O(N^{-1/2})$, which completes the proof. □

D Example: the Importance of the VaR Baseline in GCVaR

Here we show that the subtraction of the VaR baseline from the reward in GCVaR (Eq. (6)) is crucial, and without it the error in the gradient estimate may be arbitrarily large.

Consider the following example, in the setting of proposition 1: $Z \sim Normal(\theta, 1)$, and $\alpha = 0.5$. The true CVaR gradient is constant:

$$\frac{\partial}{\partial \theta_j}\Phi_\alpha(Z;\theta) = \mathbb{E}^\theta\left[\frac{\partial \log f_Z(Z;\theta)}{\partial \theta_j}(Z - \nu_\alpha(Z;\theta))\middle| Z \le \nu_\alpha(Z;\theta)\right] = 1,$$

while the term due to the baseline is

$$\mathbb{E}^\theta\left[\frac{\partial \log f_Z(Z;\theta)}{\partial \theta_j}(-\nu_\alpha(Z;\theta))\middle| Z \le \nu_\alpha(Z;\theta)\right] = -\sqrt{\frac{2}{\pi}}\theta,$$

which is unbounded in θ.

Thus, we have that $\mathbb{E}^\theta\left[\frac{\partial \log f_Z(Z;\theta)}{\partial \theta_j}(Z)\middle| Z \le \nu_\alpha(Z;\theta)\right] = 1 + \sqrt{\frac{2}{\pi}}\theta$, meaning that a naive estimator without the baseline may have an arbitrarily large error, and, for $\theta < -\sqrt{\frac{\pi}{2}}$, would even point in the opposite direction!

E Importance Sampling

For very low quantiles, i.e., α close to 0, the estimator GCVaR of Eq. (6) would have a high variance, since the averaging is effectively only over αN samples. In order to mitigate this problem, we now propose an importance sampling procedure for estimating $\frac{\partial}{\partial \theta_j}\Phi_\alpha(R;\theta)$.

Importance sampling (IS; (Rubinstein and Kroese 2011)) is a general procedure for reducing the variance of Monte–Carlo (MC) estimates. We first describe it in a general context, and then give the specific implementation for the CVaR sensitivity estimator.

E.1 Background

Consider the following general problem. We wish to estimate the expectation $l = \mathbb{E}[H(X)]$ where X is a random variable with P.D.F. $f(x)$, and $H(x)$ is some function. The MC solution is given by $\hat{l} = \frac{1}{N}\sum_{i=1}^N H(x_i)$, where $x_i \sim f$ are drawn i.i.d.

The IS method aims to reduce the variance of the MC estimator by using a different sampling distribution for the samples x_i. Assume we are given a sampling distribution $g(x)$, and that g dominates f in the sense that $g(x) = 0 \Rightarrow f(x) = 0$. We let $\mathbb{E}_f$ and $\mathbb{E}_g$ denote expectations w.r.t. f and g, respectively. Observe that $l = \mathbb{E}_f[H(X)] = \mathbb{E}_g\left[H(X)\frac{f(X)}{g(X)}\right]$,and we thus define the IS estimator $\hat{l}_{\text{IS}}$ as

$$\hat{l}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^N H(x_i)\frac{f(x_i)}{g(x_i)}, \tag{28}$$

where the x_i's are drawn i.i.d., and now $x_i \sim g$. Obviously, selecting an appropriate g such that $\hat{l}_{\text{IS}}$ indeed has a lower variance than $\hat{l}$ is the heart of the problem. One approach is by the *variance minimization* method (Rubinstein and Kroese 2011). Here, we are given a family of distributions $g(x;\omega)$ parameterized by ω, and we aim to find an ω that minimizes the variance $V(\omega) = \text{Var}_{x_i \sim g(\cdot;\omega)}\left(\hat{l}_{\text{IS}}\right)$. A straightforward calculation shows that $V(\omega) = \mathbb{E}_f\left[H(X)^2\frac{f(X)}{g(X;\omega)}\right] - l^2$, and since l does not depend on ω, we are left with the optimization problem $\min_\omega \mathbb{E}_f\left[H(X)^2\frac{f(X)}{g(X;\omega)}\right]$, which is typically solved approximately, by solving the sampled average approximation (SAA)

$$\min_\omega \frac{1}{N_{\text{SAA}}}\sum_{i=1}^{N_{\text{SAA}}}\left[H(x_i)^2\frac{f(x_i)}{g(x_i;\omega)}\right], \tag{29}$$

where $x_i \sim f$ are i.i.d. Numerically, the SAA may be solved using (deterministic) gradient descent, by noting that $\frac{\partial}{\partial \omega}\left(\frac{f(x_i)}{g(x_i;\omega)}\right) = -\frac{f(x_i)}{g(x_i;\omega)}\frac{\partial}{\partial \omega}\log g(x_i;\omega)$.

Thus, in order to find an IS distribution g from a family of distributions $g(x;\omega)$, we draw N_{SAA} samples from the original distribution f, and solve the SAA (29) to obtain the optimal ω. We now describe how this procedure is applied for estimating the CVaR sensitivity $\frac{\partial}{\partial \theta_j}\Phi_\alpha(R;\theta)$.

E.2 IS Estimate for CVaR Sensitivity

We recall the setting of Proposition 2, and assume that in addition to $f_{\mathbf{X},Y}(\mathbf{x},y;\theta)$ we have access to a family of distributions $g_{\mathbf{X},Y}\left(\mathbf{x},y;\theta,\omega\right)$ parameterized by ω. We follow the procedure outlined above and, using Proposition 2, set

$$H_j(\mathbf{X},Y) = \frac{1}{\alpha}\left(\frac{\partial\log f_Y(Y;\theta)}{\partial\theta_j} + \frac{\partial\log f_{\mathbf{X}|Y}(\mathbf{X}|Y;\theta)}{\partial\theta_j}\right)(R - \nu_\alpha(R;\theta))\mathbf{1}_{R\le\nu_\alpha(R;\theta)}.$$

However, since $\nu_\alpha(R;\theta)$ is not known in advance, we need a procedure for estimating it in order to plug it into Eq. (28). The empirical quantile $\tilde{v}$ of Eq. (5) is not suitable since it uses samples from $f_{\mathbf{X},Y}(\mathbf{x},y;\theta)$. Thus, we require an IS estimator for $\nu_\alpha(R;\theta)$ as well. Such was proposed by Glynn (1996). Let $\hat{F}_{\text{IS}}(z)$ denote the IS empirical C.D.F. of R: $\hat{F}_{\text{IS}}(z) \doteq \frac{1}{N}\sum_{i=1}^{N}\frac{f_{\mathbf{X},Y}(\mathbf{x}_i,y_i;\theta)}{g_{\mathbf{X},Y}(\mathbf{x}_i,y_i;\theta,\omega)}\mathbf{1}_{r(\mathbf{x}_i,y_i)\le z}$. Then, the IS empirical VaR is given by

$$\tilde{v}_{\text{IS}} = \inf_z \hat{F}_{\text{IS}}(z) \ge \alpha. \tag{30}$$

We also need to modify the variance minimization method, as we are not estimating a scalar function but a gradient in $\mathbb{R}^k$. We assume independence between the elements, and replace $H(x_i)^2$ in Eq. (29) with $\sum_{j=1}^{k} H_j(x_i)^2$.

Let us now state the estimation procedure explicitly. We first draw N_{SAA} i.i.d. samples from $f_{\mathbf{X},Y}(\mathbf{x},y;\theta)$, and find a suitable ω by solving the following equivalent of (29)

$$\min_\omega \frac{1}{N_{\text{SAA}}}\sum_{i=1}^{N_{\text{SAA}}}\left[\sum_{j=1}^{k} H_j(\mathbf{x}_i,y_i)^2\frac{f_{\mathbf{X},Y}(\mathbf{x}_i,y_i;\theta)}{g_{\mathbf{X},Y}\left(\mathbf{x}_i,y_i;\theta,\omega\right)}\right], \tag{31}$$

with $H_j(\mathbf{X},Y) = \frac{1}{\alpha}\left(\frac{\partial\log f_Y(Y;\theta)}{\partial\theta_j} + \frac{\partial\log f_{\mathbf{X}|Y}(\mathbf{X}|Y;\theta)}{\partial\theta_j}\right)(r(\mathbf{X},Y) - \tilde{v})\mathbf{1}_{r(\mathbf{X},Y)\le\tilde{v}}$, where $\tilde{v}$ is given in (5).

We then run the IS_GCVaR algorithm, as follows. We draw N i.i.d. samples $\mathbf{x}_1,y_1,\ldots,\mathbf{x}_N,y_N$ from $g_{\mathbf{X},Y}\left(\mathbf{x},y;\theta,\omega\right)$. The IS estimate of the CVaR gradient $\Delta_{j;N}^{\text{IS}}$ is given by

$$\Delta_{j;N}^{\text{IS}} = \frac{1}{\alpha N}\sum_{i=1}^{N}\left(\frac{\partial\log f_Y(y_i;\theta)}{\partial\theta_j} + \frac{\partial\log f_{\mathbf{X}|Y}(\mathbf{x}_i|y_i;\theta)}{\partial\theta_j}\right)\frac{f_{\mathbf{X},Y}(\mathbf{x}_i,y_i;\theta)(r(\mathbf{x}_i,y_i) - \tilde{v}_{\text{IS}})\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}_{\text{IS}}}}{g_{\mathbf{X},Y}\left(\mathbf{x}_i,y_i;\theta,\omega\right)}, \tag{32}$$

where $\tilde{v}_{\text{IS}}$ is given in (30).

Algorithm 2 IS_GCVaR

```
1: Given:
  • CVaR level $\alpha$
  • A reward function $r(\mathbf{x},y) : \mathbb{R}^n \otimes \mathcal{Y} \to \mathbb{R} \to \mathbb{R}$
  • A density function $f_{\mathbf{X},Y}(\mathbf{x},y;\theta)$
  • A density function $g_{\mathbf{X},Y}\left(\mathbf{x},y;\theta\right)$
  • A sequence $\mathbf{x}_1,y_1,\ldots,\mathbf{x}_N,y_N \sim g_{\mathbf{X},Y}$, i.i.d.
2: Set $\mathbf{x}_1^s,y_1^s\ldots,\mathbf{x}_N^s,y_N^s$ = Sort$(\mathbf{x}_1,y_1,\ldots,\mathbf{x}_N,y_N)$ by $r(\mathbf{x},y)$
3: For $i = 1,\ldots,N$ do
       $L(i) = \sum_{j=1}^{i} f_{\mathbf{X},Y}\left(\mathbf{x}_j^s,y_j^s;\theta\right)/g_{\mathbf{X},Y}\left(\mathbf{x}_j^s,y_j^s;\theta\right)$
4: Set $l = \arg\min_i L(i) \ge \alpha$
5: Set $\tilde{v}_{\text{IS}} = r(\mathbf{x}_l^s,y_l^s)$
6: For $j = 1,\ldots,k$ do
       $\Delta_{j;N}^{\text{IS}} = \frac{1}{\alpha N}\sum_{i=1}^{N}\left(\frac{\partial\log f_Y(y_i;\theta)}{\partial\theta_j} + \frac{\partial\log f_{\mathbf{X}|Y}(\mathbf{x}_i|y_i;\theta)}{\partial\theta_j}\right)\frac{f_{\mathbf{X},Y}(\mathbf{x}_i,y_i;\theta)(r(\mathbf{x}_i,y_i) - \tilde{v}_{\text{IS}})\mathbf{1}_{r(\mathbf{x}_i,y_i)\le\tilde{v}_{\text{IS}}}}{g_{\mathbf{X},Y}\left(\mathbf{x}_i,y_i;\theta\right)}$,
7: Return: $\Delta_{1;N}^{\text{IS}},\ldots,\Delta_{k;N}^{\text{IS}}$
```

Note that in our SAA program for finding ω, we estimate ν_α using crude Monte Carlo. In principle, IS may also be used for that estimate as well, with an additional optimization process for finding a suitable sampling distribution. However, a typical application of the CVaR gradient is in optimization of θ by stochastic gradient descent. There, one only needs to update ω intermittently, therefore a large sample size N_{SAA} is affordable and IS is not needed.

So far, we have not discussed how the parameterized distribution family $g_{\mathbf{X},Y}(\mathbf{x}, y; \theta, \omega)$ is obtained. While there are some standard approaches such as exponential tilting (Rubinstein and Kroese 2011), this task typically requires some domain knowledge. For the RL domain, we present a heuristic method for selecting $g_{\mathbf{X},Y}(\mathbf{x}, y; \theta, \omega)$.

E.3 CVaR Policy Gradient with Importance Sampling

As explained earlier, when dealing with small values of α, an IS scheme may help reduce the variance of the CVaR gradient estimator. In this section, we apply the IS estimator to the RL domain. As is typical in IS, the main difficulty is finding a suitable sampling distribution, and actually sampling from it. In RL, a natural method for modifying the trajectory distribution is by modifying the MDP transition probabilities. We note, however, that by such our method actually requires access to a simulator of this modified MDP. In many applications a simulator of the original system is available anyway, thus modifying it should not be a problem.

Consider the RL setting of Section 5, and denote the original MDP by M. The P.D.F. of a trajectory $\{X, Y\}$ from the MDP M, where, as defined in the main text $Y \doteq s_0, a_0, s_1, a_1, \dots, s_\tau, X \doteq \rho_0, \rho_1, \dots, \rho_{\tau-1}$ is given by

$$f_{\mathbf{X},Y}(\mathbf{x}, y; \theta) = \zeta_0(s_0) \prod_{t=0}^{\tau-1} f_{a|s}(a_t|s_t; \theta) f_{\rho|s,a}(\rho|s_t, a_t) f_{s'|s,a}(s_{t+1}|s_t, a_t).$$

Consider now an MDP $\hat{M}$ that is similar to the original MDP M but with transition probabilities $\hat{f}_{s'|s,a}(s'|s, a; \omega)$, where ω is some controllable parameter. We will later specify $\hat{f}_{s'|s,a}(s'|s, a; \omega)$ explicitly, but for now, observe that the P.D.F. of a trajectory $\{X, Y\}$ from the MDP $\hat{M}$ is given by

$$g_{\mathbf{X},Y}(\mathbf{x}, y; \theta, \omega) = \zeta_0(s_0) \prod_{t=0}^{\tau-1} f_{a|s}(a_t|s_t; \theta) f_{\rho|s,a}(\rho|s_t, a_t) \hat{f}_{s'|s,a}(s_{t+1}|s_t, a_t; \omega).$$

and therefore

$$\frac{f_{\mathbf{X},Y}(\mathbf{x}, y; \theta)}{g_{\mathbf{X},Y}(\mathbf{x}, y; \theta, \omega)} = \prod_{t=0}^{\tau-1} \frac{f_{s'|s,a}(s_{t+1}|s_t, a_t)}{\hat{f}_{s'|s,a}(s_{t+1}|s_t, a_t; \omega)}. \tag{33}$$

Using Eq. (10), Eq. (33), and the fact that $\partial \log f_{\mathbf{X}|Y}(x_i|y_i; \theta) / \partial \theta = 0$ in our formulation, the `IS_GCVaR` algorithm may be used to obtain the IS estimated gradient $\Delta_{j;N}^{\text{IS}}$, which may then be used instead of $\Delta_{j;N}$ in the parameter update equation (8).

We now turn to the problem of choosing the transition probabilities $\hat{f}_{s'|s,a}(s'|s, a; \omega)$ in the MDP $\hat{M}$, and propose a heuristic approach that is suitable for the RL domain. We first observe that by definition, the CVaR takes into account only the 'worst' trajectories for a given policy, therefore a suitable IS distribution should give more weight to such bad outcomes in some sense. The difficulty is how to modify the transition probabilities, which are defined per state, such that the whole trajectory will be 'bad'. We note that this difficulty is in a sense opposite to the action selection problem: how to choose an action at each state such that the long-term reward is high. Action selection is a fundamental task in RL, and has a very elegant solution, which inspires our IS approach.

A standard approach to action selection is through the *value-function* $V(s)$ (Sutton and Barto 1998), which assigns to each state s its expected long term outcome $\mathbb{E}[B|s_0 = s]$ under the current policy. Once the value function is known, the 'greedy selection' rule selects the action that maximizes the expected value of the next state. The intuition behind this rule is that since $V(s)$ captures the long-term return from s, states with higher values lead to better trajectories, and should be preferred.

By a similar reasoning, we expect that encouraging transitions to low-valued states will produce worse trajectories. We thus propose the following heuristic for the transition probabilities $\hat{f}_{s'|s,a}(s'|s, a; \omega)$. Assume that we have access to an approximate value function $\tilde{V}(s)$ for each state. We propose the following IS transitions for $\hat{M}$

$$\hat{f}_{s'|s,a}(s'|s, a; \omega) = \frac{f_{s'|s,a}(s'|s, a) \exp\left(-\omega \tilde{V}(s'; \theta)\right)}{\sum_y f_{s'|s,a}(y|s, a) \exp\left(-\omega \tilde{V}(y; \theta)\right)}. \tag{34}$$

Note that increasing ω encourages transitions to low value states, thus increasing the probability of 'bad' trajectories.

Obtaining an approximate value function for a given policy has been studied extensively in RL literature, and many efficient solutions for this task are known, such as LSTD (Boyan 2002) and TD(λ) (Sutton and Barto 1998). Here, we don't restrict ourselves to a specific method.

E.4 Empirical Results with Importance Sampling

We report the full details about the experimental results with importance sampling mentioned in the main text.

Fig. 2 demonstrates the importance of IS in optimizing the CVaR when α is small. We chose $\alpha = 0.01$, and $N = 200$, and compared the naive `GCVaR` against `IS_GCVaR`. As our value function approximation, we exploited the fact that the soft-max policy uses $\phi(s, a)^\top \theta$ as a sort of state-action value function, and therefore set $\tilde{V}(s) = \max_a \phi(s, a)^\top \theta$. We chose ω using SAA, with trajectories from the initial policy θ_0. We observe that `IS_GCVaR` converges significantly faster than `GCVaR`, due to the lower variance in gradient estimation.

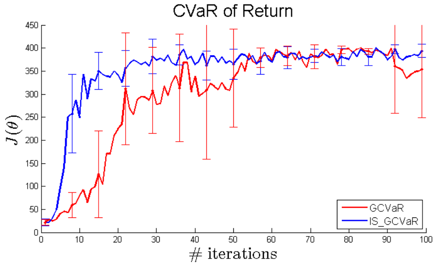


Figure 2: **IS_GCVaR vs. GCVaR** CVaR ($\alpha = 0.01$) of the return for IS_GCVaR and GCVaR vs. iteration.

References

Agarwal, V., and Naik, N. Y. 2004. Risks and portfolio decisions involving hedge funds. *Review of Financial Studies* 17(1):63–98.

Bardou, O.; Frikha, N.; and Pagès, G. 2009. Computing var and cvar using stochastic approximation and adaptive unconstrained importance sampling. *Monte Carlo Methods and Applications* 15(3):173–210.

Bäuerle, N., and Ott, J. 2011. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research* 74(3):361–379.

Baxter, J., and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *JAIR* 15:319–350.

Bertsekas, D. P. 2012. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, 4th edition.

Boda, K., and Filar, J. A. 2006. Time consistent dynamic risk measures. *Mathematical Methods of Operations Research* 63(1):169–186.

Borkar, V., and Jain, R. 2014. Risk-constrained Markov decision processes. *IEEE TAC* PP(99):1–1.

Borkar, V. S. 2001. A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. *Systems & Control Letters* 44(5):339–346.

Boyan, J. A. 2002. Technical update: Least-squares temporal difference learning. *Machine Learning* 49(2):233–246.

David, H. 1981. *Order Statistics*. A Wiley publication in applied statistics. Wiley.

Flanders, H. 1973. Differentiation under the integral sign. *The American Mathematical Monthly* 80(6):615–627.

Fu, M. C. 2006. Gradient estimation. In Henderson, S. G., and Nelson, B. L., eds., *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*. Elsevier. 575 – 616.

Furmston, T., and Barber, D. 2012. A unifying perspective of parametric policy search methods for Markov decision processes. In *NIPS*.

Gabillon, V.; Ghavamzadeh, M.; and Scherrer, B. 2013. Approximate dynamic programming finally performs well in the game of tetris. In *NIPS*.

Glynn, P. W. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33(10):75–84.

Glynn, P. W. 1996. Importance sampling for monte carlo estimation of quantiles. In *Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation*, 180–185.

Hong, L. J., and Liu, G. 2009. Simulating sensitivities of conditional value at risk. *Management Science*.

Iancu, D. A.; Petrik, M.; and Subramanian, D. 2011. Tight approximations of dynamic risk measures. *arXiv preprint arXiv:1106.6102*.

Iyengar, G., and Ma, A. 2013. Fast gradient descent method for mean-cvar optimization. *Annals of Operations Research* 205(1):203–212.

Kakade, S. 2001. A natural policy gradient. In *NIPS*.

Kushner, H., and Yin, G. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer Verlag.

Marbach, P., and Tsitsiklis, J. N. 1998. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control* 46(2):191–209.

Morimura, T.; Sugiyama, M.; Kashima, H.; Hachiya, H.; and Tanaka, T. 2010. Nonparametric return distribution approximation for reinforcement learning. In *ICML*, 799–806.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21(4):682–697.

Petrik, M., and Subramanian, D. 2012. An approximate solution method for large risk-averse Markov decision processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Prashanth, L., and Ghavamzadeh, M. 2013. Actor-critic algorithms for risk-sensitive mdps. In *NIPS*.

Prashanth, L. 2014. Policy gradients for CVaR-constrained MDPs. In *International Conference on Algorithmic Learning Theory*.

Rockafellar, R. T., and Uryasev, S. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2:21–42.

Roorda, B.; Schumacher, J. M.; and Engwerda, J. 2005. Coherent acceptability measures in multiperiod models. *Mathematical Finance* 15(4):589–612.

Rubinstein, R. Y., and Kroese, D. P. 2011. *Simulation and the Monte Carlo method*. John Wiley & Sons.

Ruszczyński, A. 2010. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming* 125(2):235–261.

Scaillet, O. 2004. Nonparametric estimation and sensitivity analysis of expected shortfall. *Mathematical Finance*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. Cambridge Univ Press.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Tamar, A.; Di Castro, D.; and Mannor, S. 2012. Policy gradients with variance related risk criteria. In *ICML*.

Thiery, C., and Scherrer, B. 2009. Improvements on learning tetris with cross entropy. *International Computer Games Association Journal* 32.

Tsitsiklis, J. N., and Van Roy, B. 1996. Feature-based methods for large scale dynamic programming. *Machine Learning* 22(1-3):59–94.

Appendix E

Supplementary Material - Policy Gradient for Coherent Risk Measures

This appendix contains supplementary material for the paper in Chapter 6.

E.1 Proof of Theorem 6.2

First note from Assumption 1 that

(i) Slater's condition holds in the primal optimization problem (6.1),

(ii) $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ is convex in ξ and concave in $(\lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$.

Thus by the duality result in convex optimization [18], the above conditions imply strong duality and we have $\rho(Z) = \max_{\xi\geq 0} \min_{\lambda^{\mathcal{P}},\lambda^{\mathcal{I}}\geq 0,\lambda^{\mathcal{E}}} L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = \min_{\lambda^{\mathcal{P}},\lambda^{\mathcal{I}}\geq 0,\lambda^{\mathcal{E}}} \max_{\xi\geq 0} L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$. From Assumption 1, one can also see that the family of functions $\{L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})\}_{(\xi,\lambda^{\mathcal{P}},\lambda^{\mathcal{E}},\lambda^{\mathcal{I}})\in\mathbb{R}^{|\Omega|}\times\mathbb{R}\times\mathbb{R}^{|\mathcal{E}|}\times\mathbb{R}^{|\mathcal{I}|}}$ is equi-differentiable in θ, $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ is Lipschitz, as a result, an absolutely continuous function in θ, and thus, $\nabla_\theta L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ is continuous and bounded at each $(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$. Then for every selection of saddle point $(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) \in \mathcal{S}$ of (6.6), using the Envelop theorem for saddle-point problems (see Theorem 4 of [65]), we have

$$\nabla_\theta \max_{\xi \geq 0} \min_{\lambda^{\mathcal{P}}, \lambda^{\mathcal{I}} \geq 0, \lambda^{\mathcal{E}}} L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = \nabla_\theta L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})|_{(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta)} . \tag{E.1}$$

The result follows by writing the gradient in (E.1) explicitly, and using the likelihood-ratio trick:

$$\sum_{\omega \in \Omega} \xi(\omega) \nabla_\theta P_\theta(\omega) Z(\omega) - \lambda^{\mathcal{P}} \sum_{\omega \in \Omega} \xi(\omega) \nabla_\theta P_\theta(\omega) = \sum_{\omega \in \Omega} \xi(\omega) P(\omega) \nabla_\theta \log P(\omega) \left(Z(\omega) - \lambda^{\mathcal{P}} \right),$$

where the last equality is justified by Assumption 2.

E.2 Gradient Results for Static Mean-Semideviation

In this section we consider the mean-semideviation risk measure, defined as follows:

$$\rho_{\text{MSD}}(Z) = \mathbb{E}\left[Z\right] + c \left(\mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])^2_+ \right] \right)^{1/2}, \tag{E.2}$$

Following the derivation in [95], note that $\left(\mathbb{E}\left[|Z|^2 \right] \right)^{1/2} = \|Z\|_2$, where $\|\cdot\|_2$ denotes the L_2 norm of the space $\mathcal{L}_2(\Omega, \mathcal{F}, P_\theta)$. The norm may also be written as:

$$\|Z\|_2 = \sup_{\|\xi\|_2 \leq 1} \langle \xi, Z \rangle,$$

and hence

$$\begin{aligned} \left(\mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])^2_+ \right] \right)^{1/2} = \sup_{\|\xi\|_2 \leq 1} \langle \xi, (Z - \mathbb{E}\left[Z\right])_+ \rangle &= \sup_{\|\xi\|_2 \leq 1, \xi \geq 0} \langle \xi, Z - \mathbb{E}\left[Z\right] \rangle \\ &= \sup_{\|\xi\|_2 \leq 1, \xi \geq 0} \langle \xi - \mathbb{E}\left[\xi\right], Z \rangle. \end{aligned}$$

It follows that Eq. (6.1) holds with

$$\mathcal{U} = \left\{ \xi' \in \mathcal{Z}^* : \quad \xi' = 1 + c\xi - c\mathbb{E}\left[\xi\right], \quad \|\xi\|_q \leq 1, \quad \xi \geq 0 \right\}.$$

For this case it will be more convenient to write Eq. (6.1) in the following form

$$\rho_{\text{MSD}}(Z) = \sup_{\|\xi\|_q \leq 1, \xi \geq 0} \langle 1 + c\xi - c\mathbb{E}\left[\xi\right], Z \rangle. \tag{E.3}$$

Let $\bar{\xi}$ denote an optimal solution for (E.3). In [95] it is shown that $\bar{\xi}$ is a contact point of $(Z - \mathbb{E}[Z])_+$, that is

$$\bar{\xi} \in \arg\max \{ \langle \xi, (Z - \mathbb{E}[Z])_+ \rangle : \|\xi\|_2 \leq 1 \},$$

and we have that

$$\bar{\xi} = \frac{(Z - \mathbb{E}[Z])_+}{\|(Z - \mathbb{E}[Z])_+\|_2} = \frac{(Z - \mathbb{E}[Z])_+}{\mathbb{SD}(Z)}. \tag{E.4}$$

Note that $\bar{\xi}$ is not necessarily a probability distribution, but for $c \in [0, 1]$, it can be shown [95] that $1 + c\bar{\xi} - c\mathbb{E}\left[\bar{\xi}\right]$ always is.

In the following we show that $\bar{\xi}$ may be used to write the gradient $\nabla_\theta \rho_{\mathrm{MSD}}(Z)$ as an expectation, which will lead to a sampling algorithm for the gradient.

Proposition E.1 *Under Assumption 2, we have that*

$$\nabla_\theta \rho_{MSD}(Z) = \nabla_\theta \mathbb{E}[Z] + \frac{c}{\mathbb{SD}(Z)} \mathbb{E}\left[(Z - \mathbb{E}[Z])_+ \left(\nabla_\theta \log P(\omega)(Z - \mathbb{E}[Z]) - \nabla_\theta \mathbb{E}[Z] \right) \right],$$

and, according to the standard likelihood-ratio method,

$$\nabla_\theta \mathbb{E}[Z] = \mathbb{E}[\nabla_\theta \log P(\omega) Z].$$

Proof. Note that in Eq. (E.3) the constraints do not depend on θ. Therefore, using the envelope theorem we obtain that

$$\begin{aligned} \nabla_\theta \rho(Z) &= \nabla_\theta \langle 1 + c\bar{\xi} - c\mathbb{E}\left[\bar{\xi}\right], Z \rangle \\ &= \nabla_\theta \langle 1, Z \rangle + c\nabla_\theta \langle \bar{\xi}, Z \rangle - c\nabla_\theta \langle \mathbb{E}\left[\bar{\xi}\right], Z \rangle. \end{aligned} \tag{E.5}$$

We now write each of the terms in Eq. (E.5) as an expectation. We start with the following standard likelihood-ratio result:

$$\nabla_\theta \langle 1, Z \rangle = \nabla_\theta \mathbb{E}[Z] = \mathbb{E}[\nabla_\theta \log P(\omega) Z].$$

Also, we have that

$$\langle \mathbb{E}\left[\bar{\xi}\right], Z \rangle = \mathbb{E}\left[\bar{\xi}\right] \mathbb{E}[Z],$$

therefore, by the derivative of a product rule:

$$\nabla_\theta \langle \mathbb{E}\left[\bar{\xi}\right], Z\rangle = \nabla_\theta \mathbb{E}\left[\bar{\xi}\right]\mathbb{E}\left[Z\right] + \mathbb{E}\left[\bar{\xi}\right]\nabla_\theta \mathbb{E}\left[Z\right].$$

By the likelihood-ratio trick and Eq. (E.4) we have that

$$\nabla_\theta \mathbb{E}\left[\bar{\xi}\right] = \frac{1}{\mathbb{SD}(Z)}\mathbb{E}\left[\nabla_\theta \log P(\omega)(Z - \mathbb{E}\left[Z\right])_+\right].$$

Also, by the likelihood-ratio trick

$$\nabla_\theta \mathbb{E}\left[\bar{\xi}Z\right] = \mathbb{E}\left[\nabla_\theta \log P(\omega)\bar{\xi}Z\right].$$

Plugging these terms back in Eq. (E.5), we have that

$$\begin{aligned}
\nabla_\theta \rho(Z) &= \nabla_\theta \mathbb{E}\left[Z\right] + c\nabla_\theta \mathbb{E}\left[\bar{\xi}Z\right] - c\nabla_\theta \mathbb{E}\left[\bar{\xi}\right]\mathbb{E}\left[Z\right] - c\mathbb{E}\left[\bar{\xi}\right]\nabla_\theta \mathbb{E}\left[Z\right] \\
&= \nabla_\theta \mathbb{E}\left[Z\right] + c\mathbb{E}\left[\bar{\xi}\left(\nabla_\theta \log P(\omega)Z - \nabla_\theta \mathbb{E}\left[Z\right]\right)\right] - c\nabla_\theta \mathbb{E}\left[\bar{\xi}\right]\mathbb{E}\left[Z\right] \\
&= \nabla_\theta \mathbb{E}\left[Z\right] + \frac{c}{\mathbb{SD}(Z)}\mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])_+\left(\nabla_\theta \log P(\omega)Z - \nabla_\theta \mathbb{E}\left[Z\right]\right)\right] - c\nabla_\theta \mathbb{E}\left[\bar{\xi}\right]\mathbb{E}\left[Z\right] \\
&= \nabla_\theta \mathbb{E}\left[Z\right] + \frac{c}{\mathbb{SD}(Z)}\mathbb{E}\left[(Z - \mathbb{E}\left[Z\right])_+\left(\nabla_\theta \log P(\omega)(Z - \mathbb{E}\left[Z\right]) - \nabla_\theta \mathbb{E}\left[Z\right]\right)\right].
\end{aligned}$$

■

Proposition 6.3 naturally leads to a sampling-based gradient estimation algortihm, which we term `GMSD` (Gradient of Mean Semi-Deviation). The algorithm is described in Algorithm 1.

E.3 Consistency Proof

Let $(\Omega_{SAA}, \mathcal{F}_{SAA}, P_{SAA})$ denote the probability space of the SAA functions (i.e., the randomness due to sampling).

Let $L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ denote the Lagrangian of the SAA problem

$$\begin{aligned}
L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = &\sum_{\omega\in\Omega}\xi(\omega)P_{\theta;N}(\omega)Z(\omega) - \lambda^{\mathcal{P}}\left(\sum_{\omega\in\Omega}\xi(\omega)P_{\theta;N}(\omega) - 1\right) \\
&- \sum_{e\in\mathcal{E}}\lambda^{\mathcal{E}}(e)f_e(\xi, P_{\theta;N}) - \sum_{i\in\mathcal{I}}\lambda^{\mathcal{I}}(i)f_i(\xi, P_{\theta;N}).
\end{aligned} \tag{E.6}$$

Algorithm 1 GMSD

```
Given:
    • Risk level $c$
    • An i.i.d. sequence $z_1, \ldots, z_N \sim P_\theta$.
Set
    $\widehat{\mathbb{E}[Z]} = \frac{1}{N} \sum_{i=1}^{N} z_i.$
Set
    $\widehat{\mathbb{SD}(Z)} = \left( \frac{1}{N} \sum_{i=1}^{N} (z_i - \widehat{\mathbb{E}[Z]})_+^2 \right)^{1/2}.$
Set
    $\widehat{\nabla_\theta \mathbb{E}[Z]} = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log P(z_i) z_i.$
Return:
    $\nabla_\theta \hat{\rho}(Z) = \widehat{\nabla_\theta \mathbb{E}[Z]} + \frac{c}{\widehat{\mathbb{SD}(Z)}} \frac{1}{N} \sum_{i=1}^{N} (z_i - \widehat{\mathbb{E}[Z]})_+ \left( \nabla_\theta \log P(z_i)(z_i - \widehat{\mathbb{E}[Z]}) - \widehat{\nabla_\theta \mathbb{E}[Z]} \right).$
```

Recall that $\mathcal{S} \subset \mathbb{R}^{|\Omega|} \times \mathbb{R} \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}_+^{|\mathcal{I}|}$ denotes the set of saddle points of the true Lagrangian (6.6). Let $\mathcal{S}_N \subset \mathbb{R}^{|\Omega|} \times \mathbb{R} \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}_+^{|\mathcal{I}|}$ denote the set of SAA Lagrangian (E.6) saddle points.

Suppose that there exists a compact set $C \equiv C_\xi \times C_\lambda$, where $C_\xi \subset \mathbb{R}^{|\Omega|}$ and $C_\lambda \subset \mathbb{R} \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}_+^{|\mathcal{I}|}$ such that:

(i) The set of Lagrangian saddle points $\mathcal{S} \subset C$ is non-empty and bounded.

(ii) The functions $f_e(\xi, P_\theta)$ for all $e \in \mathcal{E}$ and $f_i(\xi, P_\theta)$ for all $i \in \mathcal{I}$ are finite valued and continuous (in ξ) on C_ξ.

(iii) For N large enough the set $\mathcal{S}_N$ is non-empty and $\mathcal{S}_N \subset C$ w.p. 1.

Recall from Assumption 1 that for each fixed $\xi \in \mathcal{B}$, both $f_i(\xi, p)$ and $g_e(\xi, p)$ are continuous in p. Furthermore, by the S.L.L.N. of Markov chains, for each policy parameter, we have $P_{\theta,N} \to P_\theta$ w.p. 1. From the definition of the Lagrangian function and continuity of constraint functions, one can easily see that for each $(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \in \mathbb{R}^{|\Omega|} \times \mathbb{R} \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}_+^{|\mathcal{I}|}$, $L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \to L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ w.p. 1. Denote with $\mathbb{D}\{A, B\}$ the deviation of set A from set B, i.e., $\mathbb{D}\{A, B\} = \sup_{x \in A} \inf_{y \in B} \|x - y\|$. Further assume that:

(iv) If $\xi_N \in \mathcal{U}(P_{\theta;N})$ and ξ_N converges w.p. 1 to a point ξ, then $\xi \in \mathcal{U}(P_\theta)$.

According to the discussion in Page 161 of [95], the Slater condition of Assumption 1 guarantees the following condition:

(v) For some point $\xi \in \mathcal{P}$ there exists a sequence $\xi_N \in \mathcal{U}(P_{\theta;N})$ such that $\xi_N \to \xi$ w.p. 1,

and from Theorem 6.6 in [95], we know that both sets $\mathcal{U}(P_{\theta;N})$ and $\mathcal{U}(P_\theta)$ are convex and compact. Furthermore, note that we have

(vi) The objective function on (6.1) is linear, finite valued and continuous in ξ on C_ξ (these conditions obviously hold for almost all $\omega \in \Omega$ in the integrand function $\xi(\omega)Z(\omega)$).

(vii) S.L.L.N. holds point-wise for any ξ.

From (i,iv,v,vi,vii), and under the same lines of proof as in Theorem 5.5 of [95], we have that

$$\rho_N(Z) \to \rho(Z) \text{ w.p. 1 as } N \to \infty, \tag{E.7}$$

$$\mathbb{D}\{\mathcal{P}_N, \mathcal{P}\} \to 0 \text{ w.p. 1 as } N \to \infty, \tag{E.8}$$

In part 1 and part 2 of the following proof, we show, by following similar derivations as in Theorem 5.2, Theorem 5.3 and Theorem 5.4 of [95], that $L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) \to L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta)$ w.p. 1 and $\mathbb{D}\{\mathcal{S}_N, \mathcal{S}\} \to 0$ w.p. 1 as $N \to \infty$. Based on the definition of the deviation of sets, the limit point of any element in $\mathcal{S}_N$ is also an element in $\mathcal{S}$.

Assumptions (i) and (iii) imply that we can restrict our attention to the set C.

Part 1 We first show that $L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N})$ converges to

$$L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta)$$

w.p. 1 as $N \to \infty$.

For each fixed $(\lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \in C_\lambda$, the function $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ is convex and continuous in ξ. Together with the point-wise S.L.L.N. property, Theorem 7.49 of [95] implies that $L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) - L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \overset{e}{\to} 0$, where $\overset{e}{\to}$ denotes epi-convergence. Furthermore, since the objective and constraint functions are convex in ξ and are finite valued on C_ξ, the set $\mathrm{dom} L_\theta(\cdot, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ has non-empty interior. It follows from Theorem 7.27 of [95] that epi-convergence of $L_{\theta,N}$ to L_θ implies uniform convergence on C_ξ, i.e., $\sup_{\xi \in C_\xi} \left| L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) - L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \right| \le \epsilon$. On the other hand, for each fixed $\xi \in C_\xi$, the function $L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ is linear and thus continuous in $(\lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}})$ and $\mathrm{dom} L_\theta(\xi, \cdot, \cdot, \cdot) = \mathbb{R} \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{I}|}$ has non-empty interior. It follows from analogous arguments that $\sup_{(\lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \in C_\lambda} \left| L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) - L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \right| \le \epsilon$. Combining these results implies that for any $\epsilon > 0$ and a.e. $\omega_{SAA} \in \Omega_{SAA}$ there is a $N^*(\epsilon, \omega_{SAA})$ such that

$$\sup_{(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \in C} \left| L_{\theta;N}(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) - L_\theta(\xi, \lambda^{\mathcal{P}}, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \right| \le \epsilon. \tag{E.9}$$

Now, assume by contradiction that for some $N > N^*(\epsilon, \omega_{SAA})$ we have $L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) - L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) > \epsilon$. Then by definition of the saddle points

$$\begin{aligned} L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) &\geq L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) \\ &> L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) + \epsilon \\ &\geq L_\theta(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) + \epsilon, \end{aligned}$$

contradicting (E.9).

Similarly, assuming by contradiction that

$$L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) - L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) > \epsilon$$

gives

$$\begin{aligned} L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) &\geq L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) \\ &> L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) + \epsilon \\ &\geq L_{\theta;N}(\xi^*_\theta, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) + \epsilon, \end{aligned}$$

also contradicting (E.9).

It follows that $\left| L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) - L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta) \right| \leq \epsilon$ for all $N > N^*(\epsilon, \omega_{SAA})$, and therefore

$$\lim_{N\to\infty} L_{\theta;N}(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) = L_\theta(\xi^*_\theta, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta), \tag{E.10}$$

w.p. 1.

Part 2 Let us now show that $\mathbb{D}\{\mathcal{S}_N, \mathcal{S}\} \to 0$. We argue by a contradiction. Suppose that $\mathbb{D}\{\mathcal{S}_N, \mathcal{S}\} \nrightarrow 0$. Since C is compact, we can assume that there exists a sequence $(\xi^*_{\theta;N}, \lambda^{*,\mathcal{P}}_{\theta;N}, \lambda^{*,\mathcal{E}}_{\theta;N}, \lambda^{*,\mathcal{I}}_{\theta;N}) \in \mathcal{S}_N$ that converges to a point $(\bar{\xi}^*, \bar{\lambda}^{*,\mathcal{P}}, \bar{\lambda}^{*,\mathcal{E}}, \bar{\lambda}^{*,\mathcal{I}}) \in C$ and $(\bar{\xi}^*, \bar{\lambda}^{*,\mathcal{P}}, \bar{\lambda}^{*,\mathcal{E}}, \bar{\lambda}^{*,\mathcal{I}}) \notin \mathcal{S}$. However, from (E.8) we must have that $\bar{\xi}^* \in \mathcal{P}$. Therefore, we must have that

$$L_\theta(\bar{\xi}^*, \bar{\lambda}^{*,\mathcal{P}}, \bar{\lambda}^{*,\mathcal{E}}, \bar{\lambda}^{*,\mathcal{I}}) > L_\theta(\bar{\xi}^*, \lambda^{*,\mathcal{P}}_\theta, \lambda^{*,\mathcal{E}}_\theta, \lambda^{*,\mathcal{I}}_\theta),$$

by definition of the saddle point set.

Now,

$$\begin{aligned}
&L_{\theta;N}(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N}) - L_\theta(\bar{\xi}^*,\bar{\lambda}^{*,\mathcal{P}},\bar{\lambda}^{*,\mathcal{E}},\bar{\lambda}^{*,\mathcal{I}})\\
&=\Big[L_{\theta;N}(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N}) - L_\theta(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N})\Big]+ \qquad \text{(E.11)}\\
&\quad+\Big[L_\theta(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N}) - L_\theta(\bar{\xi}^*,\bar{\lambda}^{*,\mathcal{P}},\bar{\lambda}^{*,\mathcal{E}},\bar{\lambda}^{*,\mathcal{I}})\Big].
\end{aligned}$$

The first term in the r.h.s. of (E.11) tends to zero, using the argument from (E.9), and the second by continuity of L_θ guaranteed by (ii). We thus obtain that $L_{\theta;N}(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N})$ tends to $L_\theta(\bar{\xi}^*,\bar{\lambda}^{*,\mathcal{P}},\bar{\lambda}^{*,\mathcal{E}},\bar{\lambda}^{*,\mathcal{I}}) > L_\theta(\xi^*_\theta,\lambda^{*,\mathcal{P}}_\theta,\lambda^{*,\mathcal{E}}_\theta,\lambda^{*,\mathcal{I}}_\theta)$, which is a contradiction to (E.10).

Part 3 We now show the consistency of $\nabla_{\theta;N}\rho(Z)$.

Consider Eq. (6.8). Since $\nabla_\theta \log P(\cdot)$ is bounded by Assumption 2, and $\nabla_\theta f_i(\cdot;P_\theta)$ and $\nabla_\theta g_e(\cdot;P_\theta)$ are bounded by Assumption 1, and using our previous result $\mathbb{D}\{\mathcal{S}_N,\mathcal{S}\}\to 0$, we have that for a.e. $\omega_{SAA}\in\Omega_{SAA}$

$$\begin{aligned}
\lim_{N\to\infty}\nabla_{\theta;N}\rho(Z) &= \sum_{\omega\in\Omega} P_\theta(\omega)\xi^*_\theta(\omega)\nabla_\theta \log P(\omega)(Z(\omega)-\lambda^{*,\mathcal{P}}_\theta)\\
&-\sum_{e\in\mathcal{E}}\lambda^{*,\mathcal{E}}_\theta(e)\nabla_\theta g_e(\xi^*_\theta;P_\theta)\\
&-\sum_{i\in\mathcal{I}}\lambda^{*,\mathcal{I}}_\theta(i)\nabla_\theta f_i(\xi^*_\theta;P_\theta)\\
&=\nabla_\theta\rho(Z).
\end{aligned}$$

where the first equality is obtained from the Envelop theorem (see Theorem 6.2) with $(\xi^*_\theta,\lambda^{*,\mathcal{P}}_\theta,\lambda^{*,\mathcal{E}}_\theta,\lambda^{*,\mathcal{I}}_\theta)\in\mathcal{S}_N\cap\mathcal{S}$ is the limit point of the converging sequence $\{(\xi^*_{\theta;N},\lambda^{*,\mathcal{P}}_{\theta;N},\lambda^{*,\mathcal{E}}_{\theta;N},\lambda^{*,\mathcal{I}}_{\theta;N})\}_{N\in\mathbb{N}}$.

E.4 Proof of Theorem 6.5

Similar to the proof of Theorem 6.2, recall the saddle point definition of $(\xi^*_{\theta,x}, \lambda^{*,\mathcal{P}}_{\theta,x}, \lambda^{*,\mathcal{E}}_{\theta,x}, \lambda^{*,\mathcal{I}}_{\theta,x}) \in \mathcal{S}$ and strong duality result, i.e.,

$$\begin{aligned}\max_{\xi\,:\,\xi P_\theta(\cdot|x)\in\mathcal{U}(x,P_\theta(\cdot|x))} \sum_{x'\in\mathcal{X}} \xi(x')P_\theta(x'|x)V_\theta(x') &= \max_{\xi\geq 0}\min_{\lambda^{\mathcal{P}},\lambda^{\mathcal{I}}\geq 0,\lambda^{\mathcal{E}}} L_{\theta,x}(\xi,\lambda^{\mathcal{P}},\lambda^{\mathcal{E}},\lambda^{\mathcal{I}})\\ &= \min_{\lambda^{\mathcal{P}},\lambda^{\mathcal{I}}\geq 0,\lambda^{\mathcal{E}}}\max_{\xi\geq 0} L_{\theta,x}(\xi,\lambda^{\mathcal{P}},\lambda^{\mathcal{E}},\lambda^{\mathcal{I}}).\end{aligned}$$

the gradient formula in (E.1) can be written as

$$\begin{aligned}\nabla_\theta V_\theta(x) &= \nabla_\theta\left[C_\theta(x)+\gamma\max_{\xi\,:\,\xi P_\theta(\cdot|x)\in\mathcal{U}(x,P_\theta(\cdot|x))}\mathbb{E}_\xi[V_\theta]\right]\\ &= \gamma\sum_{x'\in\mathcal{X}}\xi^*_{\theta,x}(x')P_\theta(x'|x)\nabla_\theta V_\theta(x') + \sum_{a\in\mathcal{A}}\mu_\theta(a|x)\nabla_\theta\log\mu_\theta(a|x)h_\theta(x,a),\end{aligned}$$

where the stage-wise cost function $h_\theta(x,a)$ is defined in (6.5). By defining $\widehat{h}_\theta(x) = \sum_{a\in\mathcal{A}}\mu_\theta(a|x)\nabla_\theta\log\mu_\theta(a|x)h_\theta(x,a)$ and unfolding the recursion, the above expression implies

$$\begin{aligned}\nabla_\theta V_\theta(x_0) =&\widehat{h}_\theta(x_0)+\gamma\sum_{x_1\in\mathcal{X}}P_\theta(x_1|x_0)\xi^*_\theta(x_1)\Bigg[\widehat{h}_\theta(x_1)\\ &+\gamma\sum_{x_2\in\mathcal{X}}P_\theta(x_2|x_1)\xi^*_\theta(x_2)\nabla_\theta V_\theta\left(x_2\right)\Bigg].\end{aligned}$$

Now since $\nabla_\theta V_\theta$ is continuously differentiable with bounded derivatives, when $t\to\infty$, one obtains $\gamma^t\nabla_\theta V_\theta(x)\to 0$ for any $x\in\mathcal{X}$. Therefore, by Bounded Convergence Theorem, $\lim_{t\to\infty}\rho(\gamma^t V_\theta(x_t)) = 0$, when $x_0 = x$ the above expression implies the result of this theorem.

Bibliography

[1] C. Acerbi. Spectral measures of risk: a coherent representation of subjective risk aversion. *Journal of Banking & Finance*, 26(7):1505–1518, 2002.

[2] P. Artzner, F. Delbaen, J. Eber, and D. Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[3] J. Bagnell and J. Schneider. Covariant policy search. In *International Joint Conference on Artificial Intelligence*, 2003.

[4] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the twelfth international conference on machine learning*, pages 30–37, 1995.

[5] O. Bardou, N. Frikha, and G. Pagès. Computing VaR and CVaR using stochastic approximation and adaptive unconstrained importance sampling. *Monte Carlo Methods and Applications*, 15(3):173–210, 2009.

[6] A. Barto and R.H. Crites. Improving elevator performance using reinforcement learning. *Advances in neural information processing systems*, 8:1017–1023, 1996.

[7] A. Basu, T. Bhattacharyya, and V. Borkar. A learning algorithm for risk-sensitive cost. *Mathematics of Operations Research*, 33(4):880–898, 2008.

[8] N. Bäuerle and J. Ott. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, 2011.

[9] J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[10] R. Bellman. *Dynamic programming.* Dover Publications, Mineola, N.Y, 2003.

[11] D. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 4th edition, 2012.

[12] D. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

[13] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[14] K. Boda and J. A. Filar. Time consistent dynamic risk measures. *Mathematical Methods of Operations Research*, 63(1):169–186, 2006.

[15] V. Borkar. A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. *Systems & Control Letters*, 44(5):339–346, 2001.

[16] V. Borkar and R. Jain. Risk-constrained Markov decision processes. *IEEE Trans. Auto. Control*, PP(99):1–1, 2014.

[17] J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.

[18] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2009.

[19] R. Brafman and M. Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

[20] Y. Chow and M. Ghavamzadeh. Algorithms for CVaR optimization in MDPs. In *Advances in neural information processing systems*, 2014.

[21] Y. Chow and M. Pavone. A unifying framework for time-consistent, risk-averse model predictive control: theory and algorithms. In *American Control Conference*, 2014.

[22] Y. Chow, A. Tamar, S. Mannor, and M. Pavone. Risk-sensitive and robust decision-making: a CVaR optimization approach. *arXiv preprint arXiv:1506.02188*, 2015.

[23] M. Christopher and H. Lee. Mitigating supply chain risk through improved confidence. *International journal of physical distribution & logistics management*, 34(5):388–396, 2004.

[24] D. De Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of operations research*, 29(3):462–478, 2004.

[25] M. Deisenroth and C.E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[26] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[27] E. Delage and S. Mannor. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58(1):203–213, 2010.

[28] V. Desai, V. F. Farias, and C. Moallemi. Approximate dynamic programming via a smoothed linear program. *Operations Research*, 60(3):655–674, 2012.

[29] D. Duffie and J. Pan. An overview of value at risk. *The Journal of derivatives*, 4(3):7–49, 1997.

[30] Y. Engel, S. Mannor, and R. Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *International Conference on Machine Learning*, 2003.

[31] A. Farahmand, M. Ghavamzadeh, S. Mannor, and C. Szepesvári. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, pages 441–448, 2009.

[32] J. Filar, D. Krass, and K. Ross. Percentile performance criteria for limiting average markov decision processes. *IEEE Trans. Auto. Control*, 40(1):2–10, 1995.

[33] M. Fu. Gradient estimation. In *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, pages 575 – 616. Elsevier, 2006.

[34] T. Furmston and D. Barber. A unifying perspective of parametric policy search methods for Markov decision processes. In *Advances in neural information processing systems*, 2012.

[35] V. Gabillon, M. Ghavamzadeh, and B. Scherrer. Approximate dynamic programming finally performs well in the game of tetris. In *Advances in neural information processing systems*, 2013.

[36] P. Geibel and F. Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24(1):81–108, 2005.

[37] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: a survey. *under review*, 2015.

[38] P. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

[39] G. Gordon. Stable function approximation in dynamic programming. In *Proceedings of the 12th international conference on machine learning*, pages 261–268, 1995.

[40] E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of machine learning research*, 5:1471–1530, 2004.

[41] J. Hadar and W. R. Russell. Rules for ordering uncertain prospects. *The American Economic Review*, pages 25–34, 1969.

[42] R. Howard. Dynamic programming and Markov processes. 1960.

[43] R. Howard and J. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

[44] D. Iancu, M. Petrik, and D. Subramanian. Tight approximations of dynamic risk measures. *arXiv:1106.6102*, 2011.

[45] G. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[46] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

[47] S. Kakade. A natural policy gradient. In *Advances in neural information processing systems*, 2001.

[48] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[49] J. Kober and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1):171–203, 2011.

[50] J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th international conference on machine learning*, pages 521–528. ACM, 2009.

[51] V. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, 2000.

[52] G. Konidaris and A. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023, 2009.

[53] S.R. Kuindersma, R.A. Grupen, and A.G. Barto. Variable risk control via stochastic optimization. *The International Journal of Robotics Research*, 32(7):806–825, 2013.

[54] M. Lagoudakis and R. Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.

[55] K. Levy and N. Shimkin. Unified inter and intra options learning using policy gradient methods. In *Recent Advances in Reinforcement Learning*, pages 153–164. Springer, 2012.

[56] Y. Liu and S. Koenig. Risk-sensitive planning with one-switch utility functions: value iteration. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 993–999. AAAI Press, 2005.

[57] O. Maillard. Robust risk-averse stochastic multi-armed bandits. In *Algorithmic Learning Theory*, pages 218–233. Springer, 2013.

[58] A. S. Manne. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.

[59] S. Mannor, O. Mebel, and H. Xu. Lightning does not strike twice: Robust MDPs with coupled uncertainty. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 385–392, New York, NY, USA, 2012. ACM.

[60] S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

[61] S. Mannor and J. N. Tsitsiklis. Algorithmic aspects of mean-variance optimization in Markov decision processes. *European Journal of Operational Research*, 231(3):645 – 653, 2013.

[62] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 1998.

[63] H. Markowitz. *Portfolio Selection: Efficient Diversification of Investment*. John Wiley and Sons, 1959.

[64] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

[65] P. Milgrom and I. Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

[66] V. Mnih, K. Kavukcuoglu, D. Silver, A. A Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[67] T. M. Moldovan and P. Abbeel. Risk aversion in markov decision processes via near optimal chernoff bounds. In *Advances in neural information processing systems*, pages 3140–3148, 2012.

[68] J. Moody and M. Saffell. Learning to trade via direct reinforcement. *Neural Networks, IEEE Transactions on*, 12(4):875–889, 2001.

[69] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:241–276, 1999.

[70] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *International Conference on Machine Learning*, pages 799–806, 2010.

[71] S. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

[72] A. Nedic and D. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13(1-2), 2003.

[73] A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[74] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

[75] T. Osogami. Robustness and risk-sensitivity in Markov decision processes. In *Advances in neural information processing systems*, 2012.

[76] J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[77] J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.

[78] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the European Conference on Machine Learning*, pages 280–291, 2005.

[79] M. Petrik and D. Subramanian. An approximate solution method for large risk-averse Markov decision processes. In *Uncertinty in Artificial Intelligence*, 2012.

[80] G. Pflug and A. Pichler. Time consistent decisions and temporal decomposition of coherent risk functionals. *Optimization online*, 2012.

[81] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *International Conference on Machine Learning*, 2006.

[82] W. B. Powell. *Approximate Dynamic Programming.* John Wiley and Sons, 2011.

[83] L. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Advances in neural information processing systems*, 2013.

[84] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, Inc., 1994.

[85] S. Rachev and S. Mittnik. *Stable Paretian models in finance.* John Willey & Sons, New York, 2000.

[86] M. Riedmiller. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer, 2005.

[87] R. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[88] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2011.

[89] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125(2):235–261, 2010.

[90] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Math. OR*, 31(3):433–452, 2006.

[91] A. Sani, A. Lazaric, and R. Munos. Risk-aversion in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 3275–3283, 2012.

[92] B. Scherrer, M. Ghavamzadeh, V. Gabillon, B. Lesner, and M. Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 2015.

[93] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

[94] A. Shapiro. On a time consistency concept in risk averse multistage stochastic programming. *Operations Research Letters*, 37(3):143–147, 2009.

[95] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming*, chapter 6, pages 253–332. SIAM, 2009.

[96] W. F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

[97] M. Sobel. The variance of discounted markov decision processes. *Journal of Applied Probability*, pages 794–802, 1982.

[98] P. Stone and R. Sutton. Scaling reinforcement learning toward robocup soccer. In *International Conference on Machine Learning*, volume 1, pages 537–544, 2001.

[99] R. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[100] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.

[101] R. Sutton, H. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009.

[102] R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 2000.

[103] R. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.

[104] I. Szita and A. Lörincz. Learning tetris using the noisy cross-entropy method. *Neural computation*, 18(12):2936–2941, 2006.

[105] A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *International Conference on Machine Learning*, 2012.

[106] A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[107] A. Tamar and S. Mannor. Variance adjusted actor critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.

[108] A. Tamar, S. Mannor, and H. Xu. Scaling up robust MDPs using function approximation. In *International Conference on Machine Learning*, 2014.

[109] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[110] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 11:3137–3181, 2010.

[111] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *Automatic Control, IEEE Transactions on*, 42(5):674–690, 1997.

[112] S. Uryasev, S. Sarykalin, G. Serraino, and K. Kalinchenko. VaR vs CVaR in risk management and optimization. In *CARISMA conference*, 2010.

[113] C. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

[114] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[115] H. Xu and S. Mannor. Probabilistic goal Markov decision processes. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2046–2052. AAAI Press, 2011.

[116] H. Xu and S. Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

אלגוריתמים יעילים ורגישים לסיכון בלמידה באמצעות חיזוקים

אביב תמר

אלגוריתמים יעילים ורגישים לסיכון
בלמידה באמצעות חיזוקים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

אביב תמר

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
תמוז ה'תשע"ה חיפה יוני 2015

המחקר נעשה בהנחיית פרופ' שי מנור בפקולטה להנדסת חשמל.

אני מודה לפרופ' שי מנור על ההדרכה המסורה במהלך לימודיי. בידע הנרחב, באינטואיציה המדויקת, ובמצוינות המדעית הבלתי מתפשרת שי היווה עבורי מקור גדול להשראה, ועיצב את תפישתי לגבי עבודת המחקר. בנוסף לכך, שי תמך בי בצורה בלתי רגילה, והשקיע משאבים רבים בעידוד הקריירה שלי. אינני מסוגל לדמיין הנחייה יותר טובה, ואני מודה על כך מעומק ליבי.

בנוסף, אני מודה לחוקרים איתם שיתפתי פעולה בעבודה על התזה: דותן די קסטרו, פרופ' הואן שו, יונתן גלסנר, יינלם צ'או ופרופ' מוחמד גבמזדה. נהניתי מהעבודה המשותפת ולמדתי מכך רבות.

אני מודה לוועדת הבוחנים: פרופ' רון מאיר, פרופ' נחום שימקין ופרופ' ישי מנסור על ההערות וההארות.

הלחץ והעבודה הקשה בשלוש השנים האחרונות אוזנו בהרבה הפסקות קפה ושיחות מהנות עם חברי לקבוצת המערכות הלומדות, ואני מודה להם על כך.

לבסוף, אני מודה להוריי על התמיכה והעידוד המתמידים, ובמיוחד לאשתי אלינור. העידוד שלה ברגעים הקשים, התמיכה המתמדת והעצות החכמות עזרו לי מאד להשיג את מטרותיי.

אני מודה לטכניון, לקרן קלור, לקרן גוטווירת', לקרן אולנדורף, ולקרן ג'ורי על התמיכה הכספית הנדיבה בהשתלמותי. מחקר זה קיבל מימון ממועצת המחקר האירופי, תחת תכנית המסגרת השביעית של האיחוד האירופי (FP/2007 − 2013) בהסכם מלגת ERC מס' 306638, ומתכנית המסגרת השביעית של האיחוד האירופי (FP/2007−2013) תחת מלגת (PUMPPRIMING) PASCAL2 מס' 216886.

תקציר

למידה באמצעות חיזוקים הינה מסגרת חישובית לקבלת החלטות בבעיות רב־שלביות. בלמידה באמצעות חיזוקים, המסגרת הינה של סוכן היכול לבצע פעולות שונות בסביבה דינאמית, כאשר כל פעולה שהסוכן מבצע גורמת לשינוי (שיכול להיות אקראי) במצב הסביבה. בנוסף, הסוכן מקבל חיווי כמותי על איכות פעולותיו בצורה של תגמול סקלרי, אשר תלוי במצב הסביבה ובפעולת הסוכן. האופי הרב־שלבי של בעיית ההחלטה מתבטא במטרת הסוכן, שהינה לבחור פעולות אשר יביאו למקסימום את סכום כל התגמולים אשר יקבל תחת אופק זמן מוגדר מראש. מאחר והפעולות משנות את מצב הסביבה, על הסוכן לבחור נכונה בין פעולות המשיגות תגמול מידי גבוה, לבין פעולות אשר יביאו את הסוכן למצבים בהם יוכל להשיג תגמול גבוה בעתיד.

כאשר הסביבה מכילה אקראיות (למשל, כאשר שינוי המצב הוא אקראי, או רועש), סכום התגמולים שהסוכן יכול לקבל הינו אקראי גם כן. הגישה הרווחת להתמודד עם אי וודאות כזאת, הינה למקסם את תוחלת סכום כל התגמולים אשר יקבל הסוכן. על אף שגישה זו מובילה לאלגוריתמים ופיתוחים מתמטיים אלגנטיים, היא לאו דווקא מתארת את הגישה הרצויה להתמודדות עם אי־ודאות בבעיות מסוימות. בפרט, גישה זו מתעלמת מהשונות של התגמול, ומהאפשרות של מקרים נדירים בהם התגמול נמוך מאד, ואשר רצוי להימנע מהם בכל מחיר.

סוג נוסף של אי־וודאות אשר חשוב במקרים רבים הינו אי־וודאות פרמטרית לגבי אופי הסביבה. המקור לאי־וודאות זו נובע למשל משגיאות שערוך של הדינמיקה של הסביבה, ממצבים בהם הדינמיקה יכולה להשתנות לאורך זמן, או משגיאות במודל. הנקודה החשובה היא שלשגיאות אלו, גם אם הן קטנות יחסית, יכולה להיות השפעה משמעותית על הביצועים של הסוכן, ולכן מדיניות קבלת החלטות זהירה צריכה להתחשב בהן. הגישה בה אנו נוקטים בעבודה זו, הינה שעבור תחומים מסוימים, הקריטריון של מקסום תוחלת התגמול הינו בלתי מספק להתמודדות עם אי־וודאות. לחלופין, אנו מציעים קריטריונים אשר לוקחים בחשבון את מידת הסיכון בתגמול, על ידי התחשבות במאפיינים סטטיסטיים נוספים של התגמול מלבד התוחלת. בספרות הפיננסית, קריטריונים כאלו

נפוצים תחת המסגרת של ניהול סיכונים. לפיכך, בעבודה זו אנו מציעים מסגרת לניהול סיכונים בלמידה באמצעות חיזוקים.

בעבודה זו חקרנו מספר גישות לניהול סיכונים המתאימות למסגרת של למידה באמצעות סיכונים. התייחסנו לשני מקורות האי־וודאות בנפרד: אי־וודאות אינהרנטית, שמקורה באקראיות המערכת, ואי־הוודאות במודל. לגבי אי־וודאות אינהרנטית, חקרנו מספר קריטריונים חלופיים לתוחלת התגמול בלמידה באמצעות חיזוקים, אשר מתוארים בפרקים 2,3,5,6.

בפרק 2, אנו מתארים שיטת גרדיאנט־מדיניות (policy gradient) ללמידה באמצעות חיזוקים, אשר מתייחסת לקריטריוני ביצועים אשר לוקחים בחשבון את שונות התגמול. לדוגמא, ניתן למקסם את תוחלת התגמול תחת אילוץ ששונות התגמול תהיה קטנה מערך מסוים, או לחלופין, ניתן למקסם את קריטריון Sharpe, אשר מתאר את המנה בין תוחלת התגמול לסטיית התקן שלו. בפרק זה, אנו מראים כי על ידי הרחבה של שיטת גרדיאנט־המדיניות ניתן לטפל בקריטריוני ביצועים מסוגים אלו. אנו מוכיחים כי האלגוריתמים שאנו מציעים מתכנסים לנקודת אופטימום מקומית של קריטריון הביצועים. בנוסף, אנו מדגימים את ביצועי האלגוריתמים בבעיה פיננסית של ניהול תיק השקעות המכיל נכס נזיל ונכס בלתי־נזיל, באופן הרגיש לסיכון.

בפרק 3, אנו מציעים שיטה ללמידת פונקציית הערך לשונות התגמול. באנלוגיה לפונקציית הערך הרגילה בלמידה באמצעות חיזוקים, פונקציית הערך לשונות התגמול מתארת את שונות התגמול אשר יתקבל תחת מדיניות מסוימת כאשר המערכת מתחילה ממצב מסוים. לפונקציה זו חשיבות לתיאור הסיכון של מצבי מערכת שונים, והיא שלב־ביניים בדרך לפיתוח אלגוריתמים מסוג שחקן־מבקר $(\mathrm{actor} - \mathrm{critic})$ לקריטריוני ביצועים המכילים את שונות התגמול. בפרק זה, אנו מציעים שיטה ללמידת פונקציית הערך לשונות התגמול על ידי גישה של הפרשים־זמניים וקירוב פונקציונאלי. שיטה זו מתאימה למערכות בהם אוסף מצבי המערכת האפשריים הינו גדול מאד, ושיטות קודמות לפתרון בעיה זו אינן מתאימות. אנו מראים כי השיטה שאנו מציעים משיגה ביצועים טובים בהרבה לעומת שיטות קודמות כאשר תקציב הדגימה של המערכת מוגבל. בנוסף, אנו מדגימים את ביצועי האלגוריתמים בבעיית ניווט עם מרחב רציף וארבע־ממדי.

בפרק 5, אנו מרחיבים את שיטת גרדיאנט־המדיניות לקריטריון ביצועים מסוג Conditional Value at Risk (CVaR). קריטריון ה־CVaR מתייחס לתוחלת הביצועים על פני אחוזון המקרים הגרועים ביותר. זהו קריטריון נפוץ בתחומי הכלכלה והמימון, וניתן לכווננו כך שיהיה רגיש למאורעות נדירים, אך קטסטרופליים. אנו מציעים אלגוריתם חדש לשערוך גרדיאנט־המדיניות תחת מידת סיכון מסוג CVaR, ומציעים אנליזה של האלגוריתם אשר מראה כי הוא מתכנס לנקודת אופטימום מקומית, וכי

ההיסט של שערוך הגרדיאנט חסום. בנוסף, אנו מדגימים את השיטה שלנו במשימת לימוד מדיניות רגישה לסיכון למשחק Tetris. הדגמה זו ממחישה את האופי הנמנע מסיכונים של המדיניות שלמד האלגוריתם.

בפרק 6, אנו מרחיבים את השיטה שפיתחנו בפרק 5 לקריטריון ביצועים כללי ממשפחת מידות הסיכון הקוהרנטיות. משפחת מידות הסיכון הקוהרנטיות מתארת אוסף נרחב של מידות סיכון, המאופיין על ידי תנאים מתמטיים אשר הודגמו להיות מתאימים לניהול סיכונים בהשקעות פיננסיות. על ידי הרחבת שיטת גרדיאנט־המדיניות לאוסף מידות סיכון זה, אנו בעצם מציעים למקבל ההחלטות גמישות רבה בבחירת אופי ניהול הסיכונים שלו. בנוסף, אנו מתייחסים בפרק זה למידות סיכון עקביות בזמן, ומראים כי ניתן להרחיב את שיטת השחקן־מבקר (actor critic) למידות סיכון קוהרנטיות מסוג זה. בכך אנו מציעים אפשרויות נרחבות לניהול סיכונים בלמידה באמצעות חיזוקים.

לגבי אי־וודאות במודל, אנו חקרנו הרחבה של שיטה סטנדרטית לתיאור אי־וודאות מודל תחת מסגרת של תהליכי החלטה מרקוביים עמידים (robust Markov decision processes). תהליכי החלטה מרקוביים עמידים מניחים אוסף של מודלים אפשריים, ומתכננים מדיניות הטובה ביותר תחת המודל הגרוע ביותר באוסף. בצורה זו, תהליכי החלטה מרקוביים עמידים מאפשרים להבטיח ביצועים מספקים גם אם המודל עליו מבצעים את תכנון המדיניות הוא שגוי.

בפרק 4, על ידי שימוש בקירוב פונקציונאלי, אנו מרחיבים את תהליכי ההחלטה המרקוביים העמידים למסגרת של למידה באמצעות חיזוקים, ובכך מאפשרים להשתמש בהם גם בבעיות בעלות מרחב מצב גדול, או רציף. בעיות אלו לא ניתנות לפתרון על ידי שיטות סטנדרטיות לתהליכי החלטה מרקוביים עמידים, אשר מתבססות על תכנות דינאמי, ולכן מועדות לסבול מ"קללת הממדיות" (curse of dimensionality). השיטה שלנו מאפשרת להימנע מבעיה זו על ידי דגימה וקירוב פונקציונאלי. אנו מדגימים את היכולת של השיטה שלנו להתמודד עם שגיאות מודל בבעיה של מסחר עם אופציות אמריקאיות מסוג put. בבעיה זו מרחב המצב הוא רציף, ולכן מחייב קירוב פונקציונאלי. כפי שהתוצאות שלנו מראות, על ידי נקיטת גישה "פסימיסטית" לגבי המודל בפועל, השיטה שלנו מתגברת על הקשיים שנובעים משגיאות מידול. בפרט, כאשר השגיאות במודל משמעותיות, היא משיגה ביצועים טובים בהרבה משל שיטה סטנדרטית, אשר מתעלמת משגיאות אפשריות במודל.