State-Augmentation Transformations for Risk-Sensitive Markov Decision Processes

Shuai Ma

A Thesis

in

Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information Systems Engineering) at

Concordia University

Montréal, Québec, Canada

July 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Shuai Ma**

Entitled: **State-Augmentation Transformations for Risk-Sensitive Markov Decision Processes**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Jeremy Clark

_____ External Examiner
Dr. Latha Shanker

_____ Examiner
Dr. Walter Lucia

_____ Examiner
Dr. Jeremy Clark

_____ Supervisor
Dr. Jia Yuan Yu

_____ Co-supervisor
Dr. Ahmet Satir

Approved by _____
Dr. Mohammad Mannan, Graduate Program Director
Concordia Institute for Information Systems Engineering

September 27th, 2019
Date of Defence

_____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

State-Augmentation Transformations for Risk-Sensitive Markov Decision Processes

Shuai Ma, Ph.D.

Concordia University, 2019

Markov decision processes (MDPs) serve as a mathematical framework for modeling sequential decision making (SDM) where system evolution and reward are partly under the control of a decision maker and partly random. MDPs have been widely adopted in numerous fields, such as finance, robotics, manufacturing, and control systems. For stochastic control problems, MDPs are the underlying models in dynamic programming and reinforcement learning (RL) algorithms.

In this thesis, we study risk estimation in MDPs, where the variability of random rewards is taken into consideration. First, we categorize the reward into four classes: deterministic/stochastic and state-/transition-based. Though a number of theoretical methods are designed for MDPs or Markov processes with a deterministic (and state-based) reward, many practical problems are naturally modeled by processes with stochastic (and transition-based) reward. When the optimality criterion refers to the risk-neutral expectation of a (discount) total reward, we can use a model (reward) simplification to bridge the gap. However, when the criterion is risk-sensitive, a model simplification will change the risk value. For preserving the risks, we address that most, if not all, the inherent risks depend on the reward sequence $(R_t, t \in \{1, \cdots, N\})$. In order to extend the theoretical methods for practical problems with respect to risk-sensitive criteria, we propose a state-augmentation transformation (SAT). Four cases are thoroughly studied in which different forms of the SAT should be implemented for risk preservation. In numerical experiments, we compare the results from the model simplifications and the SAT, and illustrate that, i). the model simplifications change (R_t) as well as return (or total reward) distributions; and ii). the proposed SAT transforms

processes with complicated rewards, such as stochastic and transition-based rewards, into ones with deterministic state-based rewards, with intact (R_t).

Second, we consider constrained risk-sensitive SDM problems in dynamic environments. Unlike other studies, we simultaneously consider the three factors—constraint, risk, and dynamic environment. We propose a scheme to generate a synthetic dataset for training an approximator. The reasons for not using historical data are two-fold. The first reason refers to information incompleteness. Historical data usually contains no information on criterion parameters (which risk objective and constraint(s) are concerned) and (or) the optimal policy (usually just an action for each item of data), and in many cases, even the information on environmental parameters (such as all involved costs) is incomplete. The second reason is about optimality. The decision makers might prefer an easy-to-use policy than an optimal one, which is hard to determine whether the preferred policy is optimal (such as an EOQ policy), since the practical problems could be different from the theoretical model diversely and subtly. Therefore, we propose to evaluate or estimate risks with RL methods and train an approximator, such as neural network, with a synthetic dataset. A numerical experiment validates the proposed scheme.

The contributions of this study are three-fold. First, for risk evaluation in different cases, we propose the SAT theorem and corollaries to enable theoretical methods to solve practical problems with a preserved (R_t). Second, we estimate three risks with return variance as examples to illustrate the difference between the results from the SAT and the model simplification. Third, we present a scheme for constrained, risk-sensitive SDM problems in a dynamic environment with an inventory control example.

Acknowledgments

I am grateful to my supervisor, Dr. Jia Yuan Yu, for his continuous support. His scientific acumen, commitment to perfection, and intellectual integrity have a continuing impact on my professional career and personal development. I would like to sincerely thank my co-supervisor, Dr. Ahmet Satir. He was always there to listen and to give advice. I am thankful to Dr. Latha Shanker, for helping me understanding the related knowledge in finance. I would like to express my appreciation to my examining committee, who provided a lot of thoughtful comments for improving my work. I am thankful to my colleagues, Hamid Nabati, Syed Eqbal Alam, Denis Ergashbaev, Maximilien Le Clei, Victor Deleau, and Himani Saini, for organizing informative meetings and providing a stimulating environment. Lastly, and most importantly, I wish to thank my parents and my cousins. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

CDF Cumulative Distribution Function

CLT Central Limit Theorem

CVaR Conditional Value at Risk

DQN Deep Q-Network

EOQ Economic Order Quantity

MDP Markov Decision Process

MSE Mean Squared Error

NN (feed forward) Neural Network

RL Reinforcement Learning

SAT State-Augmentation Transformation

SDM Sequential Decision Making

VaR Value at Risk

Chapter 1

Introduction

> I cannot choose the best. The best chooses me.
>
> Rabindranath Tagore, Stray Birds

Life is a sequence of decision making. Everyday we are confronted with a myriad of situations where decisions need to be made. Some of them are simple, such as what to eat for lunch, or which way to take to the lab, while others could be complicated, such as which university to go, or which country to live in. The problem of decision making is ubiquitous in life, as I am now writing this introduction, deliberating on logic, making up examples, and choosing words.

To make a good decision, one usually needs to consider the impacts of the decision not only at the moment but also on the future. This process is known as a sequential decision making (SDM), and the optimal action to take now depends on the uncertain future. This uncertainty stems from many aspects of the practical situations, and has been investigated in different ways and fields. In general, this problem is known as SDM under uncertainty, and has shown to be crucial in solving a range of important problems, such as planning, scheduling, game playing, and auto-piloting. In the context of Artificial Intelligence (Russell & Norvig, 2016), the decision maker is an autonomous entity named an agent, interacting with an environment defined by parts of the world where the actions are performed. Mathematically, if we consider this decision-making process as a discrete time stochastic control process, we can model it as a Markov decision process. In the next section,

we present Markov decision process preliminaries that are directly relevant to our research. For a through coverage of the Markov decision process formulation, please refer to (Puterman, 1994).

1.1 Markov Decision Processes

A Markov decision process (MDP) is a mathematical framework for modeling SDM in situations where outcomes are partly random and partly under the control of a decision maker, and a decision maker, or an agent needs to interact with a stochastic system (environment) by taking an action at each decision epoch. The goal is to choose a sequence of actions to make the system perform optimally with respect to some performance criterion. An MDP consists of eight elements: (time) horizon, state space, action space, transition probability distribution, reward, initial state distribution, discount factor, and salvage reward function. In this study, we focus on discrete time, finite state and action spaces. A discrete-time MDP with a deterministic reward can be represented by

$$\mathcal{M} = \langle N, S, A, r, p, \mu, \gamma, v \rangle, \tag{1}$$

in which $N \in \mathbb{N} \cup \{+\infty\}$ is the time horizon. An action is chosen at each (decision) epoch $t \in \{1, \cdots, N\}$. N is ignored in infinite horizon MDPs; S is a finite state space, and $X_t \in S$ represents the state at t; A_x is the finite allowable action set for $x \in S$, $A = \bigcup_{x \in S} A_x$ is an action space, and $K_t \in A$ represents the action at t; r is a bounded and stationary (time-homogeneous) reward, and R_t denotes the immediate reward at epoch t; $p : S^2 \times A \to [0, 1]$ is a stationary transition probability distribution, and $p(y \mid x, a) = \mathbb{P}(X_{t+1} = y \mid X_t = x, K_t = a)$ denotes the probability that the system transits to $y \in S$ from the current state $x \in S$ and a chosen action $a \in A_x$; $\mu : S \to [0, 1]$ is the initial state distribution; $\gamma \in (0, 1]$ is the discount factor, and when it is one, it can be ignored; and $v : S \to \mathbb{R}$ is the salvage reward function, and we ignore it in MDPs with long or infinite horizons. We consider two types of deterministic reward functions: the state-based reward function

$$r_S : S \times A \to \mathbb{R};$$

and the transition-based reward function

$$r_T : S \times A \times S \to \mathbb{R}.$$

Similarly, a discrete-time MDP with a stochastic reward can be represented by

$$\mathcal{M} = \langle N, S, A, J, d, p, \mu, \gamma, v \rangle, \tag{2}$$

in which J is the set of possible values of the immediate rewards, and it is a finite subset of $\mathbb{R}$; and d is the reward distribution. We consider two types of stochastic reward distributions: the state-based reward distribution

$$d_S : S \times A \to \mathbb{P}(\mathbb{R});$$

and transition-based reward distribution

$$d_T : S \times A \times S \to \mathbb{P}(\mathbb{R}),$$

where $\mathbb{P}(\mathbb{R})$ is a distribution on $\mathbb{R}$ [1]. In our work, we consider d with a finite support. Taking d_S for example, we have

$$J \coloneqq \bigcup_{x \in S, a \in A_x} \operatorname{supp}(d_S(\cdot \mid x, a)).$$

1.1.1 Decision Rule and Policy

Given an MDP, a decision maker needs to choose an action at each epoch t, and a decision rule describes how an action is chosen. Generally, decision rules range from deterministic Markovian to randomized history dependent, depending on how they incorporate past information and select actions (Puterman, 1994). The history of a process at t can be denoted by $h_t = (X_1, R_1, K_1, \cdots, X_t, R_t, K_t)$, where $X_1 \sim \mu$, $X_i \in S$, $R_i \in \mathbb{R}$, and $K_i \in A$ for $i \in \{1, \cdots, t\}$, and we call a decision rule history dependent if it depends on history. We say a decision rule π_t

[1] A stochastic reward can also be represented by a reward function $r : S \times A \to \mathbb{P}(\mathbb{R})$. However, since r returns a distribution instead of a value, we suppose it is not suitable to call it a reward function.

is Markovian if the action is chosen based on the current state only, and deterministic if it is chosen with certainty A deterministic Markovian decision rule $\pi_t : S \to A$. Similarly, a randomized Markovian decision rule is $\pi_t : S \to \mathbb{P}(A)$, i.e., for a given state, it outputs a distribution on the action space. In this work, we focus on Markovian decision rules only. Randomized policy is often considered in constrained MDPs (Altman, 1999) or MDPs with a risk objective (Defourny, Ernst, & Wehenkel, 2008).

A policy π refers to a sequence of decision rules $(\pi_1, \cdots, \pi_{N-1})$, which describes how to choose actions sequentially. Based on its components, we call a policy deterministic Markovian or randomized history dependent as well. For long-horizon or infinite-horizon cases, we focus on stationary policies, where $\pi_t = \pi$ for all $t \in \{1, \cdots, N-1\}$. In our study, we focus on three Markovian policy spaces: the nonstationary deterministic policy space Π_n, which is considered in the finite horizon MDP example; the stationary randomized policy space Π_r, which is considered in the state-augmentation transformation (SAT) theorem; and the stationary deterministic policy space Π_d, which is considered in the long and infinite horizon MDP examples in Section 3.2.4 and Chapter 4.

1.1.2 Markov Reward Processes

The process generated by an MDP with a Markovian policy π is called a Markov reward process. Similar to the definitions of MDPs in Equation (1) and Equation (2), we define two Markov processes as follows. One with a deterministic reward is

$$\mathcal{M}^\pi = \langle N, S, r^\pi, p^\pi, \mu, \gamma, v \rangle,$$

and another with a stochastic reward is

$$\mathcal{M}^\pi = \langle N, S, J^\pi, d^\pi, p^\pi, \mu, \gamma, v \rangle.$$

For a Markov reward process, people optimize some function on the random reward sequence $(R_t : t \in \{1, \cdots, N\})$. From the perspective of MDP, the goal is to find an optimal policy in a specified policy space in terms of an optimality criterion.

1.1.3 Optimality Criteria

An optimality criterion is an objective with a set of constraints if possible. It defines the preference of a decison maker, and provides an objective which a decision maker wants to optimize. Most, if not all, optimality criteria consist of functions on the random reward sequence $(R_t : t \in \{1, \cdots, N\})$ in an induced Markov reward process, which is generated from an MDP with a Markovian policy. In other words, given an MDP with an optimality criterion, the goal is to find a policy which generates the optimal (R_t) with respect to the optimality criterion.

In most cases, people concern the total reward of the induced process. In a finite-horizon MDP, the total reward can be denoted by

$$\Phi_N^\pi = \sum_{t=1}^{N} R_t.$$

In an MDP with a long or infinite time horizon, we usually consider the total rewards with a discount factor $\gamma \in (0, 1)$, and represent them as

$$\Phi_d^\pi = \sum_{t=1}^{N} \gamma^{t-1} R_t$$

and

$$\Phi^\pi = \sum_{t=1}^{+\infty} \gamma^{t-1} R_t,$$

respectively, and we call the latter one *return*. For brevity, we denote the three total rewards by Φ_N, Φ_d, and Φ, respectively.

The standard objective is to optimize the expectation of the (discounted) total reward. Without constraints, in short-horizon MDPs, this problem can be solved by backward induction; in long- or infinite-horizon MDPs, this problem is usually solved by policy iteration or value iteration. The three solutions are all based on the Bellman equation. Since it is not in the scope of this study, for details on problems with the standard optimality criteria, see (Puterman, 1994). Given a set of cost functions [2], the constraints could refer to the expectations of some discounted total costs, the problem can be solved with the aid of occupation measure. For more details on this type of

[2] In standard constrained MDP models in (Altman, 1999), the cost functions are deterministic and depend on state and action spaces.

constrained MDPs, see (Altman, 1999).

For an expectation-based optimality criterion, such as

$$\max \mathbb{E}(\Phi), \tag{3}$$

all of the information necessary to make an optimal decision during the whole process can be summarized in a simplified model. Here we present the model simplification in three cases.

(1) The reward is transition-based. For example, a deterministic transition-based reward function r_T can be simplified to a deterministic state-based reward function r_S in an MDP by

$$r_S(x,a) = \sum_{y \in S} r_T(x,a,y) p(y \mid x,a), \text{ for all } x \in S, a \in A_x.$$

(2) The reward is stochastic. For example, a stochastic state-based reward d_S can be simplified to a deterministic state-based reward function r_S in an MDP by

$$r_S(x,a) = \sum_{j \in J} d_S(j \mid x,a) j, \text{ for all } x \in S, a \in A_x.$$

(3) The policy is randomized. For example, an infinite-horizon MDP with a deterministic state-based reward $r : S \times A \to \mathbb{R}$ and a randomized (stationary) policy $\pi : S \to \mathbb{P}(A)$ induces a Markov process with a stochastic state-based reward

$$d_S^\pi(x) = r(x,a) \text{ with probability } \pi(a \mid x), \text{ for all } x \in S, a \in A_x.$$

Notice that, in this case, the transition probability is simplified as well.

$$p^\pi(y \mid x) = p(y \mid x,a)\pi(a \mid x), \text{ for all } x, y \in S, a \in A_x. \tag{4}$$

The cases for model simplification can be quite complicated, we may consider the most general

case, an MDP with a d_T and a $\pi \in \Pi_r$. To simplify it to a Markov process with an r_S, we have

$$r_S^\pi(x) = \sum_{a \in A_x, y \in S, j \in J} \pi(a \mid x) p(y \mid x, a) d_T(j \mid x, a, y) j, \text{ for all } x \in S,$$

as well as the simplification on transition probability (Equation 4).

When the criterion is risk-neutral, such as the (discounted) expected total reward, we can do all bunches of model simplifications and preserve the expectation. However, when the criterion is risk-sensitive, the model simplifications cannot preserve (R_t), which results in wrong risk values. Next, we review risks for quantifying uncertainty in reinforcement learning (RL), and see how (R_t) is crucial in risk-sensitive RL.

1.2 Risks

In RL, uncertainty is studied from two perspectives. One is the *external* uncertainty, which refers to the parameter uncertainty or disturbance. When the model is unknown, its parameters are usually estimated first, and then the optimal solution is calculated with a model-based approach. However, the parameter estimation depends on noisy data in practice, and the modeling errors may result in negative consequences. In control theory, this problem is known as robust control. Robust control methods consider the uncertain parameters within some compact sets, and optimize the expected return with the worst-case parameters, in order to achieve good robust performance and stability (Nilim & Ghaoui, 2005).

The other concerns the *inherent* (or *internal*) uncertainty, which comes from the stochastic nature of the processes. We claim that most, if not all, inherent risks depend on the reward sequence $(R_t : t \in \{1, \cdots, N\})$, and an inherent risk can be denoted by $\rho : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$, where $N \in \mathbb{N}^+ \cup \{+\infty\}$. The inherent risk can be quantified by a dynamic risk or a law-invariant risk. Given a Markov process with an r_S and a deterministic initial state, a sequence of risk functions $(\rho_i : i \in \{1, \cdots, N\})$, and denote the immediate reward at epoch t by R_t, a dynamic risk can be denoted in general as

$$R_1 + \rho_1(R_2 + \rho_2(R_3 + \rho_3(\cdots))),$$

which is sensitive to the order of the immediate rewards. Notice that R_1 is determined here. Dynamic risks are usually assumed to have a set of properties, such as Markov, monotonicity and coherence, which yields a time-consistent risk with a nested structure. For further information on the dynamic risk, see (Ruszczyński, 2010). Given a discount factor γ, a law-invariant (Kusuoka, 2001) risk in an infinite horizon is a functional Ψ on the return Φ. Three types of law-invariant risk have been widely studied in RL area.

1.2.1 Exponential Utility Risk

The original goal of a utility function is to represent the subjective preference (Howard & Matheson, 1972). One classic example can be the "St. Petersburg Paradox," which refers to a lottery with an infinite expected reward, but no one would put up an arbitrary high stake to play it, since the probability of obtaining an high enough reward is too small. Mathematically, a utility function $U : \mathbb{R} \to \mathbb{R}$ is a mapping from objective value space for all possible outcomes to subjective value space. A utility function is usually in the form $U^{-1}\{\mathbb{E}[U(\Phi)]\}$, where U is a strictly increasing function. The most common used utility risk in RL is the exponential utility (Chung & Sobel, 1987)

$$\Psi_U(\Phi, \beta) = \beta^{-1} \log\{\mathbb{E}[\exp(\beta\Phi)]\},$$

where β models a constant risk sensitivity that risk-averse when $\beta < 0$. This can be seen more clearly with the Taylor expansion of the utility

$$\beta^{-1} \log\{\mathbb{E}[\exp(\beta\Phi)]\} = \mathbb{E}(\Phi) + \frac{\beta}{2}\mathbb{V}(\Phi) + \mathcal{O}(\beta^2). \tag{5}$$

The exponential utility risk is also known as the entropic risk.

1.2.2 Mean-Variance Risk

The mean-variance risk is also known in finance as the modern portfolio theory (Mannor & Tsitsiklis, 2011; Sobel, 1994; White, 1988). The mean-variance analysis aims at optimal return at a given level of risk, or the optimal risk at a given level of return. In RL, several mean-variance

models have been studied. The variance and the standard deviation of the return are denoted by $\mathbb{V}(\Phi)$ and $\sigma(\Phi)$, respectively. One model could be

$$\Psi_V(\Phi, k) = \mathbb{E}(\Phi) - k\sigma(\Phi), \tag{6}$$

where k is a risk parameter, and when $k > 0$, it is a risk-averse risk. This is the first mean-variance model for exploring inventory management related problems (Lau, 1980). The other model can be maximizing the expected return with a variance constraint, or minimizing the variance with an expected return constraint (T.-M. Choi, Li, & Yan, 2008). In chapter 4, we consider the function $\mathbb{E}(\Phi)/\mathbb{V}(\Phi) > q$, where q is a predefined risk parameter, as a constraint. For a review on mean-variance risk, see (Chiu & Choi, 2016).

1.2.3 Quantile-based Risk

The last type of risk used in practice refers to quantiles, which requires us to pay attention to discontinuities and intervals of quantile numbers. A commonly used quantile-based risk is value at risk (VaR). VaR originates from finance. For a given portfolio, a loss threshold (target level) and a horizon, VaR concerns the probability that the loss on the portfolio exceeds the threshold over the time horizon. Mathematically, VaR can be defined as to find a policy which maximizes the smallest possible outcome with respect to a specified probability level. In RL, the VaR objective can also be defined as to find a policy which maximizes the probability that the return is larger than or equal to a specified target (threshold) (Filar, Krass, Ross, & Member, 1995; Wu & Lin, 1999). Denote F_Φ^π as the return cumulative distribution function (CDF) from a policy π. In this paper, we forcus on VaR and consider two problems (Filar et al., 1995) in an infinite-horizon MDP.

Problem 1.1. *Given a probability* $\alpha \in [0, 1]$, *find the optimal threshold* $\rho_\alpha = \sup\{\tau \in \mathbb{R} \mid \mathbb{P}(\Phi > \tau) \geq \alpha, \pi \in \Pi\} = \sup\{\tau \in \mathbb{R} \mid F_\Phi^\pi(\tau) \leq 1 - \alpha, \pi \in \Pi\}$.

Problem 1.2. *Given a threshold* $\tau \in \mathbb{R}$, *find the optimal probability* $\eta_\tau = \sup\{\alpha \in [0, 1] \mid F_\Phi^\pi(\tau) \leq 1 - \alpha, \pi \in \Pi\}$.

Both VaR problems relate to

$$\inf\{F_\Phi^\pi \mid \pi \in \Pi\}, \tag{7}$$

and we name it VaR function.

In a long- or infinite-horizon case, the return distribution can be estimated with a strictly increasing function (Meyn & Tweedie, 2009). In this case, VaR can be considered as a study of the return distribution, since any point along P_Φ is (estimated) $(\rho_\alpha, 1-\eta_\tau)$ with $\tau = \rho_\alpha$ or $\alpha = 1-\eta_\tau$. Therefore, both VaR problems refer to P_Φ. VaR is straightforward but hard to deal with since it is not a coherent risk (Riedel, 2004). In many cases, conditional VaR (also known as expected shortfall) is preferred over VaR since it is coherent (Artzner, Delbaen, Eber, & Heath, 1998), i.e., it has some intuitively reasonable properties (convexity, for example). However, when the return can be assumed to be approximately normally distributed, VaR can be simply estimated with $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$ (Ma & Yu, 2017).

In our study, we focus on law-invariant risks, which concern the return Φ. The three law-invariant risks can be either calculated (mean-variance risk), estimated (utility risk), or estimated with assumption (quantile-based risk) with $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$. Next, we review the return variance formula proposed by Sobel (1982) for an infinite-horzion Markov process with a deterministic reward.

1.3 Variance Formula for Markov Processes

As shown above, all of the three law-invariant risks can be estimated with the return variance. To calculate the return variance, Sobel (1982) presented a formula for Markov processes with deterministic rewards.

Theorem 1.1. *Given an infinite-horizon Markov process $\langle S, r_S^\pi, p^\pi, \gamma\rangle$ with a finite state space $S = \{1, \cdots, |S|\}$, a bounded r_S^π, any determined initial state, and a discount factor $\gamma \in (0,1)$. Denote the transition matrix by P, in which $P(x,y) = p^\pi(y \mid x), x, y \in S$. Denote the conditional return expectation by $v_x = \mathbb{E}(\Phi \mid X_1 = x)$ for any deterministic initial state $x \in S$, and the conditional expectation vector by v. Similarly, denote the conditional return variance by $\psi_x = \mathbb{V}(\Phi \mid X_1 = x)$, and the conditional variance vector by ψ. Let θ denote the vector whose xth*

component is $\theta_x = \sum_{y \in S} p_\pi(x, y)(r_\pi(x) + \gamma v_y)^2 - v_x^2$. *Then*

$$v = r_\pi + \gamma P v = (I - \gamma P)^{-1} r_\pi,$$
$$\psi = \theta + \gamma^2 P \psi = (I - \gamma^2 P)^{-1} \theta.$$

Notice that the variance formula is for Markov processes with deterministic rewards only. One problem is how to apply the method to practical problems with stochastic rewards.

1.4 Thesis Contributions and Outline

Till now, we present the background knowledge, which includes the notations for MDPs and Markov processes, four types of reward, optimality criteria, three law-invariant risks and the return variance formula for a Markov process with a deterministic reward. With the model simplification in mind, an initial question could be

Question 1.1. *In order to implement the return variance formula on a Markov process with stochastic reward (*d_T *or* d_S*), shall we do the model simplification?*

The answer is negative. In Chapter 3, we illustrate that the model simplification changes the return distribution in different cases. With knowing that most, if not all, law-invariant risks depend on the reward sequence (R_t), the next question could be

Question 1.2. *Given a Markov process with a reward* d_T*, is there a Markov process with a reward* r_S*, such that both processes share the same* (R_t)*?*

The answer is positive. Our work starts from answering the two questions, and extends to a scheme for constrained and risk-sensitive MDPs in dynamic environments. Our contributions are three-fold: i). we propose an SAT for preserving reward sequence (R_t), and enable methods for MDPs with simple rewards for MDPs with complicated rewards; ii). by virtue of the proposed SAT, we estimate the three law-invariant risks in an MDP with a stochastic reward and a deterministic

policy; and iii). we propose a scheme for practical problems with risk-sensitive criteria in a dynamic environment.

Based on the previously reviewed background knowledge, we outline the content of the three chapters dedicated to each of these questions as follows, leaving the precise statement of contributions to the introduction section of each chapter.

In Chapter 2, we propose the SAT theorem as an MDP homomorphism, and derive three corollaries for different cases. In Chapter 3, in an inventory control example, we illustrate that i) the model simplification leads to a wrong risk estimation; ii) the proposed SAT enables the methods for MDPs (or Markov processes) with stochastic rewards and preserves the reward sequence (R_t); and iii) the three law-invariant risks can be estimated with the SAT and return variance formula. In Chapter 4, we present a scheme for constrained and risk-sensitive SDM problems in dynamic environments. In Chapter 5, we give conclusions and discuss future research.

We show that, 1) the model simplification presented in Section 1.1.3 will change the reward sequence (R_t); 2) the three commonly used law-invariant risks can be evaluated or estimated by the return variance in a Markov process with a stochastic reward; and 3) in the framework of MDP, a practical problem with a stochastic reward can be solved with a risk-sensitive criterion (with risk-sensitive constraints) in a dynamic environment.

Chapter 2

State-Augmentation Transformation

2.1 Introduction

In this chapter, we review the state-augmentation transformation (SAT) for risk-sensitive MDPs, which is one of the main contributions of this research. The motivation arises from the model difference between theoretical methods and practical problems with respect to forms of reward. On the one hand, some methods require MDPs, or Markov processes, to be with deterministic (and state-based) reward functions, even in risk-sensitive cases (Berkenkamp, Turchetta, Schoellig, & Krause, 2017; Borkar, 2002; Chow, Ghavamzadeh, Janson, & Pavone, 2017; García & Fernández, 2015; Gilbert & Weng, 2016; Huang & Haskell, 2017; Junges, Jansen, Dehnert, Topcu, & Katoen, 2016; Shen, Tobia, Sommer, & Obermayer, 2014). On the other hand, for many practical problems, whose underlying MDPs or Markov processes have stochastic (and transition-based) rewards. In these cases, if we implement the model simplification (see Section 1.1.3) for implementing the methods, it will change the reward sequences (R_t), as well as the reward distributions, which are crucial in risk-sensitive scenarios. In this chapter, we propose four successively more general forms of SAT to solve this problem in different settings. Since most risks are functions of (R_t) (see Section 1.2), we show that, the proposed SAT transforms an MDP (or a Markov process) for a theoretical method which requires a simple reward while preserving (R_t) for a risk-sensitive objective.

2.1.1 Chapter Contribution

In this section, we propose the SAT for four cases, which successively generalize the case of the Problem 1.2 to a final MDP version with a positive answer. As far as we know, no similar work has been done before. All the theorems and corollaries are for infinite-horizon problems, but can be easily updated for finite-horizon cases. Furthermore, to relieve from the enlarged state space, a definition of isotopic states is proposed for a state lumping theorem, which aims at decreasing the size of augmented state space, with considering the special structure of the transformed transition probability.

2.1.2 Chapter Organization

This chapter is structured as follows. In Section 2.2, we propose the SAT theorem, corollaries and their algorithms for four cases, and discuss the pros and cons of the SAT. Since the transformed process suffers from an enlarged state space, we provide a state lumping theorem in Section 2.3 to combine states which satisfy some conditions. Finally, we draw a conclusion in Section 2.4.

2.2 State-Augmentation Transformation

This section thoroughly describes how to implement the proposed SAT in four cases, which are generalized successively with respect to the different cases in the model simplification(see Section 1.1.3). We firstly give the related theorem and definitions for understanding the SAT as an MDP homomorphism, then present the SAT theorem and algorithm, as well as a proof in the appendix. After unveiling the essence of the SAT, we give three corollaries for the other simpler cases.

2.2.1 State-Augmentation Transformation Theorem

Firstly, we give the SAT theorem as an MDP homomorphism for an MDP considering a stationary randomized policy space Π_r. Comparing with the original SAT theorem (Ma & Yu, 2019a), the homomorphism version of SAT works on a more abstract level. Though it is for infinite-horizon cases, it is easy to be extended to finite-horizon cases.

An MDP homomorphism is a formalism that captures an intuitive notion of specific equivalence between MDPs (Ravindran & Barto, 2002). In order to convert an MDP $\mathcal{M}$ with d_T to $\mathcal{M}^\dagger$ with r_S with considering Π_r and preserve (R_t), we regard each "situation", which determines immediate rewards, as an augmented state. We can then attach each possible reward value to an augmented state in $\mathcal{M}^\dagger$. Formally, we define the SAT homomorphism as follows.

Definition 2.1 (State-augmentation transformation)**.** *The SAT for MDPs is a homomorphism h from an MDP $\mathcal{M} = \langle S, A, J, p, d_T, \mu \rangle$ to an MDP $\mathcal{M}^\dagger = \langle S^\dagger, A, r_S, p^\dagger, \mu^\dagger \rangle$. The state space $S^\dagger = S^\ddagger \cup S_n$, where $S^\ddagger = S^2 \times A \times J$, $S_n = \{s_{n,x}\}_{x \in S}$, and $S_n \cap S^\ddagger = \emptyset$. For $x^\dagger = (x, a_x, y, i), y^\dagger = (y, a_y, z, j) \in S^\ddagger$, $a_x \in A_x$, $a_y \in A_y$, and $s_{n,y} \in S_n$, we have $A_{x^\dagger} = A_{s_{n,y}} = A_y$; $r_S(y^\dagger, a_y) = j$, $r_S(s_{n,y}, a_y) = 0$; $p^\dagger(y^\dagger \mid x^\dagger, a_y) = p^\dagger(y^\dagger \mid s_{n,y}, a_y) = p(z \mid y, a_y) d_T(j \mid y, a_y, z)$; and $\mu^\dagger(s_{n,y}) = \mu(y)$, $\mu^\dagger(y^\dagger) = 0$.*

We call $\mathcal{M}^\dagger$ the homomorphic image of $\mathcal{M}$ under h. For any policy π for $\mathcal{M}$, there exists a policy $\pi^\dagger$ for $\mathcal{M}^\dagger$, such that the two induced processes share the same (R_t). We define the mapping between the two policy spaces as a policy lift.

Definition 2.2 (Policy lift)**.** *Let $\mathcal{M}^\dagger$ be a homomorphic image of $\mathcal{M}$ under h. Let π be a stochastic policy in $\mathcal{M}$. Then π lifted to $\mathcal{M}^\dagger$ is the policy $\pi^\dagger$ such that $\pi^\dagger(a \mid x^\dagger) = \pi(a \mid y)$ for $x^\dagger = (x, a, y, i) \in S^\ddagger$ and $\pi^\dagger(a \mid x^\dagger) = \pi(a \mid x)$ for $x^\dagger = s_{n,x} \in S_n$.*

Given an MDP with a policy, the randomness of the induced Markov reward process can be studied in its underlying probability space.

Definition 2.3 (Underlying probability space)**.** *Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space, and $(E, \mathcal{B})$ a measurable space with $E = S \times J$. An induced Markov reward process can be represented by an $(E, \mathcal{B})$-valued stochastic process on $(\Omega, \mathcal{F}, \mathcal{P})$ with a family $(Y_t)_{t \in \mathbb{N}}$ of random variables $Y_t : (\Omega, \mathcal{F}) \to (E, \mathcal{B})$ for $t \in \mathbb{N}$. $(\Omega, \mathcal{F}, \mathcal{P})$ is called the underlying probability space of the process $(Y_t)_{t \in \mathbb{N}}$. For all $\omega \in \Omega$, the mapping $Y(\cdot, \omega) : t \in \mathbb{N} \to Y_t(\omega) \in E$ is called the trajectory of the process with respect to ω. The process $(Y_t)_{t \in \mathbb{N}}$ is progressively measurable with respect to the filtration $(\mathcal{F}_t)_{t \in \mathbb{N}}$.*

A homomorphism version of the SAT theorem is as follows, which claims that the probability

Algorithm 1 the SAT for an MDP

Input: an MDP with a stochastic reward $\mathcal{M} = \langle S, A, J, p, d_T, \mu \rangle$.
Output: an MDP with a deterministic reward $\mathcal{M}^\dagger = \langle S^\dagger, A, r_S, p^\dagger, \mu^\dagger \rangle$.

Set $S^\ddagger = S^2 \times A \times J$;
Set a null state space $S_n = \{s_{n,1}, \cdots, s_{n,|S|}\}$, with $S_n \cap S^\ddagger = \emptyset$; ▷ needed when policy is uncertain
Define the state space $S^\dagger = S^\ddagger \cup S_n$;
Initialize the transition probability $p^\dagger(\cdot \mid \cdot, \cdot) = 0$;
for all $x^\dagger = (x, a_x, y, i), y^\dagger = (y, a_y, z, j) \in S^\ddagger$, $a_x \in A_x$, $a_y \in A_y$, and $s_{n,y} \in S_n$ **do**
    Define the action space $A_{x^\dagger} = A_{s_{n,y}} = A_y$;
    Define the reward $r_S(y^\dagger, a_y) = j$, and $r_S(s_{n,y}, a_y) = 0$;
    Define the transition probability $p^\dagger(y^\dagger \mid x^\dagger, a_y) = p(y^\dagger \mid s_{n,y}, a_y) = p(z \mid y, a_y) d_T(j \mid y, a_y, z)$;
    Define the initial state distribution $\mu^\dagger(s_{n,y}) = \mu(y)$, and $\mu^\dagger(y^\dagger) = 0$.
end for

measure on trajectories is preserved under h. Therefore, as a subsequence of a sample path, the probability measure on (R_t) is preserved as well.

Theorem 2.1 (Probability measure preservation)**.** *Let $\mathcal{M}^\dagger$ be an image of $\mathcal{M}$ under homomorphism h. Let $\pi^\dagger$ be the stochastic policy lifted from π. For the two processes $\mathcal{M}$ with π and $\mathcal{M}^\dagger$ with $\pi^\dagger$, there exists a bijection $f_\Omega : \Omega \to \Omega^\dagger$, such that for the underlying probability space $(\Omega, \mathcal{F}, \mathcal{P})$ for the first process, we have a sample path probability space $(\Omega^\dagger, \{f_\Omega(b) : b \in \mathcal{F}\}, P^\dagger)$ for the second process, such that for any $t \in \mathbb{N}$, $\mathcal{P}^\dagger(\{f_\Omega(b) : b \in \mathcal{F}_t\}) = \mathcal{P}(\{\mathcal{F}_t\})$.*

The proof of the Theorem 2.1 is detailed in the appendix. The SAT theorem claims that, for an $\mathcal{M}$ with a d_T, there exists an $\mathcal{M}^\dagger$ with an r_S, such that for any given π for $\mathcal{M}$, there exists a corresponding $\pi^\dagger$ for $\mathcal{M}^\dagger$, such that both Markov reward processes share the same (R_t) (with a null state in the transformed one). The algorithm of the theorem for an MDP with a d_T, which has a $|J| \leq +\infty$, is is presented in Algorithm 1. Notice that the discount factor γ is ignored, since it has not effect on (R_t).

The essence of the SAT lies in regarding all the factors determining one immediate reward together as a "situation." In the most general case considered in the Theorem 2.1, this can be explained as follows.

Remark 2.1 (State augmentation in the transformations)**.** *In order to transform an MDP with a d_T to the one with an r_S, and preserve (R_t), a bijective mapping from an "augmented" state space to a possible "situation" space is needed. In this case, a situation can be defined by a tuple $\langle x, a, y, j \rangle$,*

in which $x, y \in S, a \in A_x, j \in \operatorname{supp}(d_T(\cdot \mid x, a, y))$.

In short, the transformation varies according to the situations of the problems. Next, we present the corollaries for three simplified situations.

2.2.2 Four Cases for SAT

Originally, to acquire an MDP (or a Markov process) with an r_S (or r_S^π) with a preserved (R_t), the SAT theorem 2.1 is generalized from simple cases.

- Case 1: a Markov process with an r_T^π;
- Case 2: a Markov process with a d_T^π;
- Case 3: an MDP with a d_T and a $\pi \in \Pi_r$; and
- Case 4: an MDP with a Π_r.

The transformation for Case 1 is firstly proposed in (Ma & Yu, 2017), which is a special case of Case 2. We denote this relationship by *Case 0* $\prec$ *Case 1*. Case 2 can be considered as an MDP with an r_T and a deterministic policy. Case 3 describes a policy evaluation scenario, and can be also used for constrained MDPs. Case 4 is for a direct policy search (such as gradient descent) scenario from a risk-sensitive perspective. In all the four cases, the model simplification (see Section 1.1.3) will lose all moment information except for the first one (mean). The four cases have the relationship

$$\textit{Case 1} \prec \textit{Case 2} \prec \textit{Case 3} \prec \textit{Case 4}.$$

Next, we review the SAT for Case 1–3 with corollaries and algorithms. Considering the relationship between the cases, all needed algorithms and theorems for different cases can be derived from the constructive proof for Theorem 2.1 (see the appendix) with some slight changes.

Case 1

The SAT for Case 1 is originally proposed in (Ma & Yu, 2017). When the criterion is risk-sensitive, for a Markov reward process derived from an MDP with an r_T and a policy $\pi \in \Pi_d$,

Algorithm 2 the SAT for a Markov process with a deterministic transition-based reward

Input: a Markov process $\langle S, r_T^\pi, p^\pi, \mu \rangle$. ▷ r_T^π is deterministic and transition-based
Output: a Markov process $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$. ▷ r_S^π is deterministic and state-based

Define the state space $S^\dagger = S^2$;
Initialize the transition probability $p^{\pi\dagger}(\cdot \mid \cdot) = 0$;
for all $x^\dagger = (x, y) \in S^\dagger$ **do**
    Define the reward $r_S^\pi(x^\dagger) = r_T^\pi(x, y)$;
    for all $y^\dagger = (\cdot, x) \in S^\dagger$ **do**
        Define the transition probability $p^{\pi\dagger}(x^\dagger \mid y^\dagger) = p^\pi(y \mid x)$;
    end for
    Define the initial state distribution $\mu^{\pi\dagger}(x^\dagger) = \mu(x) p^\pi(y \mid x)$.
end for

we cannot directly apply the methods if they require the reward to be state-based. If the reward is simplified as described in Section 1.1.3, the reward sequence (R_t) will be changed, which results in a wrong risk value. This case can be considered in a scenario that a deterministic policy π is evaluated in an MDP with an r_T with a risk-sensitive criterion. In order to implement those methods, we apply the SAT for Case 1 to transform the Markov reward process with an r_T^π to one with an r_S^π. The original theorem in (Ma & Yu, 2017) now can be considered as a corollary of the SAT Theorem 2.1, which can be stated as follows.

Corollary 2.1 (Transformation for a Markov process with a transition-based reward)**.** *For a Markov process $\langle S, r_T^\pi, p^\pi, \mu \rangle$, there exists a Markov process $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$, such that both processes share one reward sequence (R_t).*

Notice that, i) the original version of the Corollary 2.1 in (Ma & Yu, 2017) refers to a finite-horizon case, and here we revise it for consistency; and ii) the initial state distribution in the transformed Markov process depends on the policy π. For the concern on i) the finite horizon with a salvage reward function, and ii) a Markov process derived from an MDP with a nonstationary policy, see (Ma & Yu, 2017).

The algorithm for the Corollary 2.1 is presented in Algorithm 2.

Case 2

The SAT for Case 2 is originally proposed in (Ma & Yu, 2019a). Similarly, when the criterion is risk-sensitive, for a Markov reward process derived from an MDP with a d_T and a policy $\pi \in \Pi_d$,

Algorithm 3 the SAT for a Markov process with a stochastic transition-based reward

Input: a Markov process $\langle S, J^\pi, d_T^\pi, p^\pi, \mu \rangle$. ▷ d_T^π is stochastic and transition-based
Output: a Markov process $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$. ▷ r_S^π is deterministic and state-based

Define the state space $S^\dagger = S^2 \times J$;
Initialize the transition probability $p^{\pi\dagger}(\cdot \mid \cdot) = 0$;
for all $x^\dagger = (x, y, j) \in S^\dagger$ **do**
    Define the reward $r_S^\pi(x^\dagger) = j$;
    for all $y^\dagger = (\cdot, x, \cdot) \in S^\dagger$ **do**
        Define the transition probability $p^{\pi\dagger}(x^\dagger \mid y^\dagger) = p^\pi(y \mid x) d_T^\pi(j \mid x, y)$;
    end for
    Define the initial state distribution $\mu^{\pi\dagger}(x^\dagger) = \mu(x) p^\pi(y \mid x) d_T^\pi(j \mid x, y)$.
end for

we cannot directly apply the methods if they require the reward to be state-based, or simplify the reward function as described in Section 1.1.3. This case can be considered in a scenario that a deterministic policy π is evaluated in an MDP with a d_T and a risk-sensitive criterion. In order to implement those methods, we propose the SAT for Case 2 to transform the Markov process with a d_T^π to one with an r_S^π, which can be stated as follows.

Corollary 2.2 (Transformation for a Markov process with a stochastic reward)**.** *For a Markov process* $\langle S, J^\pi, d_T^\pi, p^\pi, \mu \rangle$*, there exists a Markov reward process* $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$*, such that both Markov reward processes share one reward sequence* (R_t).

The algorithm for the Corollary 2.2 is presented in Algorithm 3.

Case 3

The SAT for Case 3 is originally proposed in (Ma & Yu, 2019a). Similar to Case 1 and 2, Case 3 also refers to a policy evaluation case. When the criterion is risk-sensitive, for a Markov reward process derived from an MDP with a d_T and a policy $\pi \in \Pi_r$, we cannot directly apply the methods if they require the reward to be state-based, or simplify the reward function as described in Section 1.1.3. This case can be considered in a scenario that a stochastic policy π is evaluated in an MDP with a d_T and a risk-sensitive criterion. In order to implement those methods, we propose the SAT for Case 3 to transform the Markov process with a d_T^π to one with an r_S^π, which can be stated as follows.

Algorithm 4 the SAT for MDP with a randomized policy

Input: an MDP $\langle S, A, J, d_T, p, \mu \rangle$ with a $\pi \in \Pi_r$. ▷ d_T is stochastic and transition-based
Output: a Markov process $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$. ▷ r_S^π is deterministic and state-based

Define the state space $S^\dagger = S^2 \times A \times J$;
Initialize the transition probability $p^{\pi\dagger}(\cdot \mid \cdot) = 0$;
for all $x^\dagger = (x, a, y, j) \in S^\dagger$ **do**
    Define the reward $r_S^\pi(x^\dagger) = j$;
    for all $y^\dagger = (\cdot, \cdot, x, \cdot) \in S^\dagger$ **do**
        Define the transition probability $p^{\pi\dagger}(x^\dagger \mid y^\dagger) = \pi(a \mid x)p(y \mid x, a)d_T(j \mid x, a, y)$;
    end for
    Define the initial state distribution $\mu^{\pi\dagger}(x^\dagger) = \mu(x)\pi(a \mid x)p(y \mid x, a)d_T(j \mid x, a, y)$.
end for

Corollary 2.3 (Transformation for an MDP with a randomized policy)**.** *For an MDP $\langle S, A, J, d_T, p, \mu \rangle$ with a randomized policy $\pi \in \Pi_r$, there exists a Markov reward process $\langle S^\dagger, r_S^\pi, p^{\pi\dagger}, \mu^{\pi\dagger} \rangle$, such that both Markov reward processes share one reward sequence (R_t).*

The algorithm for the Corollary 2.3 is presented in Algorithm 4. Notice that unlike Case 1 and 2, we cannot operate on a derived Markov reward porcess directly this time, since now the action for a given state is uncertain.

2.2.3 Discussion

There are mainly two benefits brought by the SAT. Firstly, it converts an $\mathcal{M}$ (or $\mathcal{M}^\pi$) with a discrete stochastic reward distributed on a finite support, into an $\mathcal{M}^\dagger$ (or $\mathcal{M}^{\pi\dagger}$) with a deterministic state-based reward. The essence of the SAT is a bijective mapping from an "augmented" state space to a possible "situation" space, in which each situation determines an immediate reward. This mapping not only extends a number of methods provided for $\mathcal{M}^\dagger$ to work for $\mathcal{M}$, but also allows us to analyze the immediate rewards as states with methods such as z-transform (Howard, 1964). Secondly, action is "removed" from the reward function and becomes a component of the augmented state. This renders action to impact on transitions only. There are two disadvantages as well. One is the size of the augmented state space is now $(|S^2 \times A \times J| + |S|)$, which is much larger than $|S|$. The other is that the SAT removes the recurrence property from a recurrent MDP, which may result in prohibitions of some methods. In the next section, we propose a state lumping theorem to relieve transformed Markov processes of enlarged state spaces.

2.3 State Lumping

In this section, we propose a state lumping theorem based on a definition of isotopic states. The propose of the theorem is to decrease the size of the enlarged state space in transformed Markov processes. The state lumping theorem 2.2 is originally proposed in (Ma & Yu, 2019b). Considering the special structure of $p_{\pi\dagger}$, we propose a definition of isotopic states for combining two states in the enlarged state space.

Definition 2.4 (Isotopic states)**.** *In a Markov process $\langle S, r_S^\pi, p^\pi, \mu\rangle$, if there exist two states $x_i, x_j \in S$, such that,*

- *Condition 1: $r_S^\pi(x_i) = r_S^\pi(x_j)$; and*
- *Condition 2: $p^\pi(y \mid x_i) = p^\pi(y \mid x_j)$ for $y \in S \setminus \{x_i, x_j\}$,*

then we say x_i and x_j are isotopic.

With the definition of isotopic states, we propose a theorem on reward preservation in state lumping as follows. The proof of the Theorem 2.2 is detailed in the appendix.

Theorem 2.2 (Reward preservation in state lumping)**.** *If the two states $x_i, x_j \in S$ are isotopic in a Markov process $\mathcal{M} = \langle S, r^\pi, p^\pi, \mu\rangle$, then there exists a Markov process $\mathcal{M}' = \langle S', r^{\pi\prime}, p^{\pi\prime}, \mu'\rangle$, in which $S' = S \setminus \{x_j\}$, and for $x, y \in S' \setminus \{x_i\}$ and $z \in S'$, $r^{\pi\prime}(z) = r^\pi(z)$, $p^{\pi\prime}(x_i \mid y) = p^\pi(x_i \mid y) + p^\pi(x_j \mid y)$, $p^{\pi\prime}(x \mid y) = p^\pi(x \mid y)$, $p^{\pi\prime}(x \mid x_i) = p^\pi(x \mid x_i)$, $p^{\pi\prime}(x_i \mid x_i) = p^\pi(x_i \mid x_i) + p^\pi(x_j \mid x_i)$, and $\mu'(x) = \mu(x)$, $\mu'(x_i) = \mu(x_i) + \mu(x_j)$, such that the two Markov process $\mathcal{M}$ and $\mathcal{M}'$ share the same $(R_t : t \in \{1, \cdots, N\})$.*

Theorem 2.2 claims that, the isotopic state lumping will not change the reward sequence. Therefore, most, if not all, risks will be preserved by an induced Markov process with a smaller state space from the isotopic state lumping. Furthermore, we call a set of states isotopic class if any two members in the set are isotopic. It is easy to generalize the Theorem 2.2 for isotopic classes.

Corollary 2.4 (Reward preservation in class lumping)**.** *In a transformed Markov process $\langle S, r^\pi, p^\pi, \mu\rangle$, if there exists a state set S_l, such that for any two states $x_i, x_j \in S_l$, two conditions are satisfied:*

i). $r^{\pi}(x_i) = r^{\pi}(x_j)$; and ii). $p^{\pi}(y \mid x_i) = p^{\pi}(y \mid x_j)$ for $y \in S \setminus S_l$, then we say S_l is an isotopic class, which can be regarded as one state.

The results of this section are based in part on the results of (Burke & Rosenblatt, 1958; Kemeny & Snell, 1976), in which the lumpability in Markov processes was thoroughly studied. By comparison, the Definition 2.4 and the Theorem 2.2 are based on single states instead of partitions of state space, i.e., in Condition 2 in Definition 2.4, x_i, x_j and y are single states instead of partitions. Though it is usually hard to partition a general state space with equivalent partitioned transition probabilities (Definition 8.4 in (Harrison & Patel, 1992)), a special structural property of the transformed transition probability from the SAT increases the chance of state lumping. We consider Case 2 for example. Since the transition distribution $p^{\pi\dagger}(\cdot \mid x^{\dagger})$ for $x^{\dagger} = (x, y, j) \in S^{\dagger}$ depends on y only, a sufficient condition for two states being isotopic could be that, the two states share the last two components. This sufficient condition can be stated as follows.

Corollary 2.5 (A sufficient condition for isotopic states)**.** *In a transformed Markov process with a state space $S^{\dagger}$, if there exist two states $x^{\dagger} = (x_1, x_2, i), y^{\dagger} = (y_1, y_2, j) \in S^{\dagger}$, such that $x_2 = y_2$ and $i = j$, then x and y are isotopic.*

In summary, for states which satisfied the two conditions in Definition 2.4, we may regard them as one state and keep (R_t) intact. Considering that the transition probability in the transformed Markov process has a special structural property above, the augmented state space can be shrunk under a fair condition.

2.4 Conclusion

In this chapter, we solved the problems stem and generalized from the Question 1.2. We present the main SAT theorem and three corollaries for the four cases (see Section 2.2.2). Though a transformed process preserves (R_t), it suffers from an enlarged state space. Taking the Case 4 as an example, the size of the state space is increased from $|S|$ to $(|S^2 \times A \times J| + |S|)$, which may cause a computational problem. To make the enlarged state space smaller, we propose a state lumping theorem based on a definition of isotopic states.

In the next chapter, we estimate three law-invariant risks in different cases. firstly, in a small inventory control MDP, we illustrate how the SAT works in short, long, and infinite horizons, and show how the model simplification leads to wrong results. Secondly, the VaR is considered as a risk-sensitive objective in the inventory control MDP, and estimated in different cases. Thirdly, we illustrate how the state lumping theorem works in a toy example, and the other two law-invariant risks—exponential utility risk and mean-variance risk—are estimated in an MDP with a d_T by virtue of the proposed SAT.

Chapter 3

Risk Evaluations with State-Augmentation Transformations

3.1 Introduction

In RL, many practical problems can be modeled as MDPs with complicated forms of reward, such as d_T, whereas many theoretical methods are for MDPs or Markov processes with relatively simple forms of reward, such as r_S or r_T. When the objective is the standard criterion—the expectation of (discounted) total reward, the model simplification works perfectly. However, when the objective is risk-sensitive, the model simplification leads to an incorrect value. Many practical problems require stronger reliability guarantees, especially where the small probability events have serious consequences, such as self-driving cars and medical diagnosis. In these cases, a risk-sensitive criterion should be considered. In sequential decision making problems, a risk-sensitive criterion refers to a risk, or a risk function, which assigns a scalar to a reward sequence $\{R_t\}$. In this chapter, we focus on law-invariant risks described in Section 1.2, which are functions mapping random variables (return Φ) to real numbers. With the aid of the SATs, we can apply the return variance formula, which requires processes with deterministic rewards, to ones with stochastic transition-based rewards, and at the same time preserve the reward sequence (and the reward distribution).

In Section 1.2, we claim that, the three law-invariant risks—exponential utility, mean-variance,

and VaR—can be estimated with the return variance. The variance can be calculated in a Markov process, but with a deterministic reward only. Besides, a number of RL methods have similar requirements (Berkenkamp et al., 2017; Borkar, 2002; Chow et al., 2017; García & Fernández, 2015; Gilbert & Weng, 2016; Huang & Haskell, 2017; Junges et al., 2016; Shen et al., 2014) even in a risk-sensitive scenario. However, the reward functions are usually stochastic in many practical problems. Therefore, in this chapter, we consider the return variance formula (see Section 1.3) for example, and show how the proposed SAT can solve this problem—to enable those methods, which work for processes with simple forms of reward only, to work for ones with complicated forms of reward. Furthermore, with the help of the SAT and the return variance formula, we estimate the three law-invariant risks—VaR, exponential utility, and mean-variance. In the examples in the following sections, we illustrate that, when the objective is risk-sensitive, and the reward needs to be converted to a simpler form, the SAT should be implemented instead of the model simplification.

3.1.1 Chapter Contribution

In this chapter, we estimate the three law-invariant risks presented in Section 1.2 with the return variance. Meanwhile, we consider the variance formula (See Section 1.3) as an example to show how the proposed SAT preserves (R_t), and how the model simplifications (see Section 1.1.3) lead to wrong results in terms of risk objectives. Since many RL methods require the reward function to be deterministic and state-based, the transformation is needed for the MDPs with other types of reward functions in the risk-sensitive problems. The proposed SAT presents a connection between theoretical methods and practical problems in risk-sensitive RL, and we believe that many related studies should be revisited with the proposed SAT instead of applying the common-used model simplification directly. In the last example, we illustrate how the state lumping works for an MDP with a stochastic policy.

3.1.2 Chapter Organization

This section consists of two MDP examples. The first one refers to an inventory control MDP, in which we consider VaR objectives in short, long, and infinite horizons, respectively. Since the VaR measures the return distribution, this example illustrate the error in the reward distribution

estimation from the model simplification. The second example illustrates the state lumping in an MDP with a randomized policy, and we estimate the other two law-invariant risks with the SAT and the model simplification.

3.2 Examples

In this section, we illustrate that, i). how the model simplifications change the (discounted) total reward distributions; ii). how the proposed SAT preserves the return variance and enable the variance formula; and iii). how to use the isotopic state lumping to alleviate the enlarged state space. With the return variance, we estimate VaR functions for the two VaR problems (see 1.2.3) in MDPs with short, long, and infinite horizons, and estimate the other two law-invariant risks in an MDP with an infinite horizon.

3.2.1 An MDP for Inventory Control Problems

We constructs an MDP for a single-product stochastic inventory control problem based on (Puterman, 1994, Section 3.2.1). The assumptions for problem modeling are listed in the appendix. Define the inventory capacity $M \in \mathbb{N}^+$, and the state space $S = \{0, \cdots, M\}$. Briefly, at time epoch $t \in \mathbb{N}$, denote the inventory level by $X_t \in S$ before the order, the order quantity by $K_t \in \{0, \cdots, M - X_t\}$, the demand by $D_t \in \{0, \cdots, M\}$ with a time-homogeneous probability distribution, then we have $X_{t+1} = \max\{X_t + K_t - D_t, 0\}$. For $x \in S$, denote the cost to order x units by $c(x)$, a fixed cost $W \geq 0$ for placing orders, then we have the order cost $o(x) = (W + c(x))\mathbb{1}_{[x>0]}$. Denote the revenue when x units of demand is fulfilled by $f(x)$, the maintenance fee by $m(x)$. The real reward is $r_T(X_t, K_t, X_{t+1}) = f(X_t + K_t - X_{t+1}) - o(K_t) - m(X_t)$.

We set the parameters as follows. The fixed order cost $W = 4$, the variable order cost $c(x) = 2x$, the maintenance fee $m(x) = x$, the warehouse capacity $M = 2$, and the price $f(x) = 8x$. Given the probabilities of demands $\mathbb{P}(D_t = 0) = 0.25$, $\mathbb{P}(D_t = 1) = 0.5$, $\mathbb{P}(D_t = 2) = 0.25$, respectively, we have the transition probability $p(y \mid x, a)$ for any $x, y \in S$ and $a \in A_x$. Set the initial distribution $\mu(0) = 1$. Notice that, the reward function is deterministic and transition-based in this

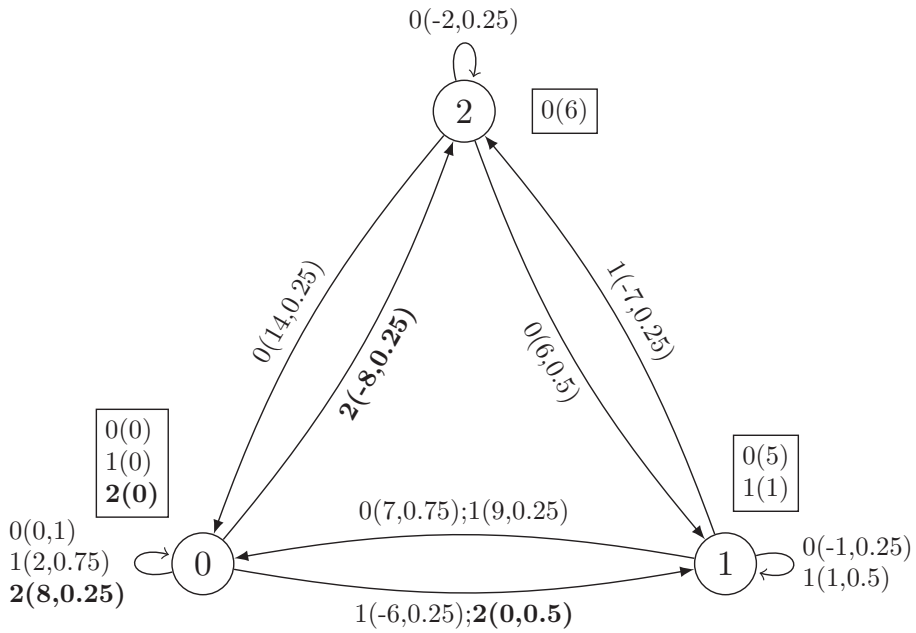

Figure 3.1: An MDP with a transition-based reward function and its counterpart with a state-based reward function following the model simplification. Labels along transitions denote $a(r_T(x,a,y), p(y|x,a))$, and labels next to states denote $\boxed{a(r_S(x,a))}$, the state-based reward function simplified with Equation (1). For example, the labels in bold are interpreted as follows: the label $2(0, 0.5)$ below the transition from 0 to 1 means that the reward $r_T(0,2,1) = 0$ and the transition probability is 0.5; the label $\boxed{2(0)}$ near state 0 means when $X_t = 0$ and $K_t = 2$, the simplified reward $r_S(0,2) = 0$.

MDP. The simplified reward r_S can be calculated by Equation (1), which is state-based. As illustrated in Figure 3.1, now we have two MDPs with different reward functions: $\langle N, S, A, r_T, p, \mu, \gamma\rangle$ and $\langle N, S, A, r_S, p, \mu, \gamma\rangle$.

3.2.2 VaR in Short-Horizon MDPs

Here, we illustrate that both VaR problems (Problem 1.1 and 1.2) refer to the VaR function in the two MDPs described above with the time horizon $N = 3$, and show how the model simplifications affect the total reward distribution and the VaR function. Firstly, we show for a given deterministic policy, how the model simplification changes the cumulative reward distribution. We consider the optimal policy for the nominal expected total reward objective. The optimal nonstationary deterministic policy for both MDPs is $\pi^* = (\pi_1, \pi_2) \in \Pi_n$, where $\pi_1(0) = \pi_2(0) = 2$ and $\pi_2(x) = 0$, for $x \in \{1, 2\}$. The expected total reward $\mathbb{E}(\Phi) = 6.5625$. As shown in Figure 3.2, with

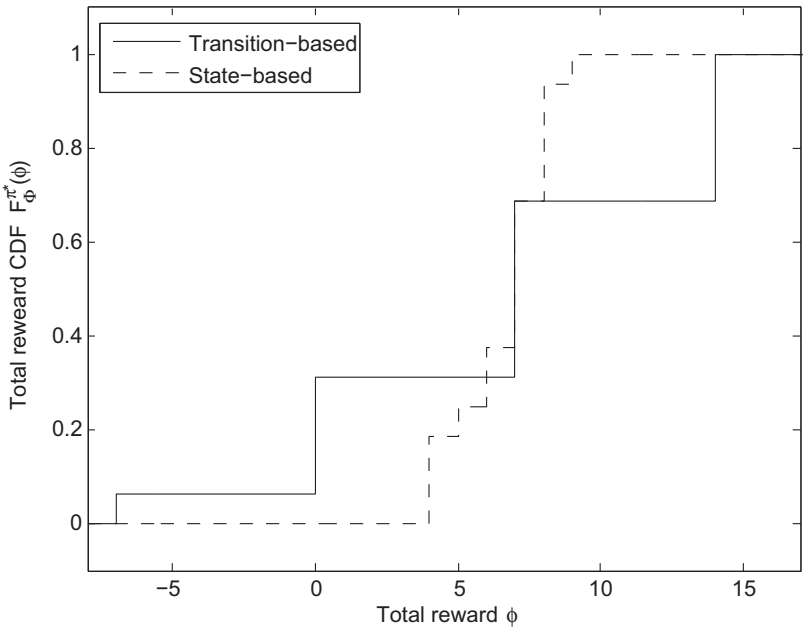

Figure 3.2: Taking the expected total reward as the objective, the two MDPs, $\langle N, S, A, r, p, \mu, v\rangle$ and $\langle N, S, A, r', p, \mu, v\rangle$, share the same optimal policy π^*, but the total reward CDFs are different.

the expected total reward objective, the simplification of transition-based reward function leads to a different total reward distribution, which leads to a different VaR function (Equation (7)) shown in Figure 3.3.

Unlike the expected total reward, VaR does not have a time-consistent property, so the backward induction cannot be implemented directly. For short-horizon MDPs, one method is the augmented-state 0-1 MDP (Xu & Mannor, 2011), which incorporates the cumulative reward space into the state space, and brings in the threshold and reorganizes the MDP components, in order to calculate the probability as expectation. The other method is to enumerate all policies to achieve the total reward CDF set, then calculate the VaR function. Here we describe and compare the two methods.

Augmented-State 0-1 MDP

Since the cumulative reward is needed for the optimality (Bouakiz & Kebir, 1995), an augmented state space is adopted to keep track of it. For short-horizon MDPs considering the VaR

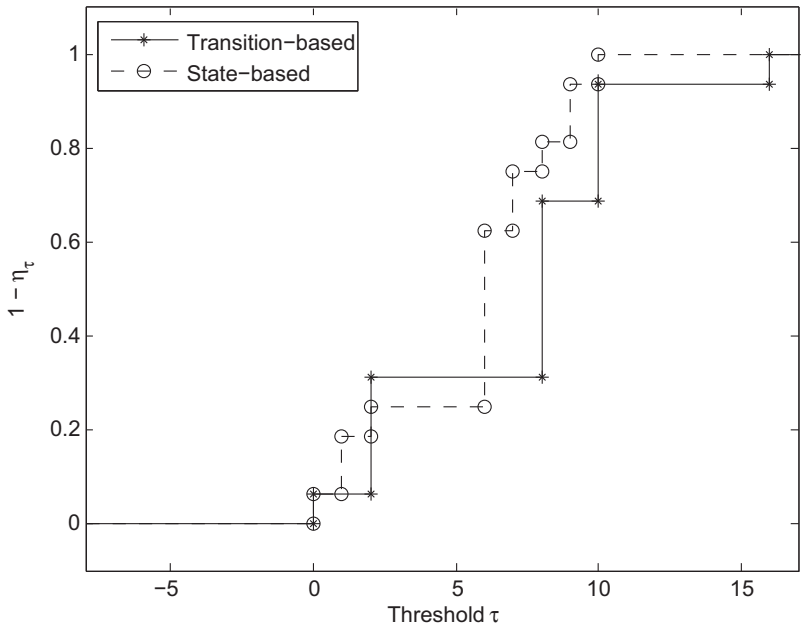

Figure 3.3: The VaR functions for the short-horizon inventory MDPs with two rewards from the model simplification and the SAT, respectively.

Problem 1.2, Xu and Mannor (2011) presented a state augmentation method to include the cumulative reward in state space. This state augmentation is also implemented in several former studies (Bouakiz & Kebir, 1995; Ohtsubo & Toyonaga, 2002; Wu & Lin, 1999; Xu & Mannor, 2011). Here we give the proof for Lemma 3.1 in the appendix, which is missed in (Xu & Mannor, 2011).

Lemma 3.1. *For every finite-horizon MDP* $\langle N, S, A, r, p, \mu, v\rangle$*, there exists an augmented-state 0-1 MDP* $\langle N, S', A, v', p', \mu'\rangle$*, in which the optimal expected total reward equals to the optimal probability* $\eta_\tau \in [0,1]$ *of the original MDP with a threshold* $\tau \in \mathbb{R}$*.*

This 0-1 MDP enables backward induction to solve VaR Problem 1.2 with calculating the probability η_τ as an expectation. The augmented-state 0-1 MDP algorithm is presented in the appendix. In the implementation of the algorithm, it is worth noting that, in most instances, it is more efficient to deal with the state space in a time-dependent way, since at each epoch, only a subspace of S' is feasible. Furthermore, since the reward function is recorded by means of the cumulative reward, the model simplification still affects the result.

Now we use the augmented-state 0-1 MDP method to solve the inventory control problem described above. We consider the VaR Problem 2 with the threshold $\tau = 9$, for instance. The optimal policy for the MDP with the transition-based reward function is $\pi^* = (\pi_1, \pi_2)$, where $\pi_1((0,0)) = 2$, $\pi_2((0,2)) = 2$, $\pi_2((0,8)) =$1 or 2, $\pi_2((1,0)) = 1$ or 2, $\pi_2((1,6)) = 0$ or 1, and $\pi_2((2,-2)) = 0$ or 1. The optimal probability is $\eta^*_\tau = 0.3125$. The optimal policy for the MDP with a simplified reward function is $\pi^* = (\pi_1, \pi_2)$, where $\pi_1((0,0)) = 2$, $\pi'_2((2,0)) = 0$. And the optimal probability is $\eta^*_\tau = 0.1875$. The conclusion which claims that the model simplification changes the VaR, is verified here.

As described above, it is clear that the augmented-state 0-1 MDP method is for VaR Problem 1.2 with a specified threshold only. In order to solve either VaR problem with any $\tau \in \mathbb{R}$ or $\alpha \in [0,1]$, we enumerate all the deterministic policies on the augmented state space S' to acquire the optimal VaR.

Policy Enumeration

In a finite-horizon MDP, the VaR function is a right-continuous step function. It is worth noting that its jump points contains the VaR information. Denote the jump set $\{x \in \mathbb{R} \mid \eta_{x^-} < \eta_x\}$.

Remark 3.1 (VaR function in a finite horizon)**.** *Given the jump set $\{x \in \mathbb{R} \mid \eta_{x^-} < \eta_x\}$ for a finite-horizon MDP, for any $\tau \in \mathbb{R}$, denote $\alpha^* = 1 - \eta_\tau$, and then $\tau^* = \rho_{(1-\alpha^*)} \in \{x \in \mathbb{R} \mid \eta_{x^-} < \eta_x\}$, then we have $(\tau^*, 1-\alpha^*) \in \{(x_i^-, \eta_{x_i^-})\} \cup \{(+\inf, 1)\}$. There exists a similar conclusion for any $\alpha \in [0,1]$.*

In other words, in a finite-horizon MDP, the set $\{(x_i^-, \eta_{x_i^-})\} \cup \{(+\inf, 1)\}$ contains all solutions to the two VaR problems defined in Section 1.2.3. In the same MDP setup, we study the effect of the model simplification on the VaR function. Given the two MDPs, $\langle N, S, A, r, p, \mu, v\rangle$ and $\langle N, S, A, r', p, \mu, v\rangle$, Figure 3.3 illustrates that the simplification of reward function changes the VaR function. Now we can verify the solution to the VaR problem with a specified threshold $\tau = 9$. Furthermore, for any threshold $\tau \in \mathbb{R}$, we can acquire $\eta_\tau \in [0,1]$ along the curves, so as for any probability $\alpha \in [0,1]$. For example, when $\tau = 7.5$, η_τ for the MDP with a transition-based reward function is 0.6875 $(1 - 0.3125)$, and η'_τ for the MDP with a state-based reward function is 0.25

$(1 - 0.75)$. The model simplification results in a nontrivial difference $(0.6875 - 0.25 = 0.4375)$, which could have an effect on a risk-sensitive decision making.

Briefly, the model simplification changes the VaR function. For the VaR Problem 1.2, the augmented-state 0-1 MDP method enables the backward induction algorithm. However, the method fails for long-horizon MDPs, and it works for the VaR problem 1.2 with a specified threshold only. Since both VaR problems relate to the VaR function, how to calculate it effectively in a short horizon needs further study. Next, we compare the VaR results from the SAT and the model simplification in a long horizon.

3.2.3 VaR in Long-Horizon MDPs

Since it is intractable to find the exact optimal policy for a long-horizon MDP with a VaR objective, we look for a deterministic stationary policy instead. With the aid of spectral theory and the central limit theorem, the total reward CDF set $\{F_\Phi^\pi\}$ can be estimated in an MDP with a deterministic reward by enumerating all $\pi \in \Pi_d$, in order to achieve the VaR function. For MDPs with transition-based reward functions, we use the Corollary 2.1 and present an algorithm to transform a Markov reward process with the reward function $r_T^\pi : S \times S \to \mathbb{R}$ to one with $r_S^{\pi\dagger} : S^\dagger \to \mathbb{R}$, where $S^\dagger = S \times S$, in order to keep the VaR function (which is derived from $\{F_\Phi^\pi\}_{\pi\in\Pi_d}$) intact.

Total Reward CDF Estimation

Firstly we estimate the total reward distribution for a long-horizon Markov reward process derived from an MDP with an r_S and a policy $\pi \in \Pi_d$, and denote the reward in the derived Markov process by r_S^π. Given an MDP with a state-based reward function $\langle N, S, A, r_S, p, \mu\rangle$[1] and a $\pi \in \Pi_d$, we have a Markov reward process $\langle N, S, r_S^\pi, p^\pi, \mu\rangle$. For $x, y \in S$, the reward is $r_S^\pi(x) = r_S(x, \pi(x))$, and the transition probability is $p^\pi(x, y) = p(x, \pi(x), y)$.

Kontoyiannis and Meyn (2003) proposed a method to estimate F_Φ^π. In a positive recurrent Markov reward process with invariant probability measure (stationary distribution) $\xi : S \to [0, 1]$, we have the total reward $\Phi_N^\pi = \sum_{t=0}^{N-1} r_S^\pi(X_t)$, and the averaged reward $\zeta(r_S^\pi) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}(\Phi_N)$,

[1] Though the salvage function v is ignored when the horizon is long, it can be involved if necessary.

which can be expressed as $\zeta = \xi r_S^\pi$. Define the limit $\hat{r_S^\pi} = \lim_{N\to\infty} \mathbb{E}_\mu(\Phi_N - N\zeta)$, which solves the Poisson equation

$$P\hat{r_S^\pi} = \hat{r_S^\pi} - r_S^\pi + \zeta,$$

where P is the transition matrix and $P(x, y) = p^\pi(x, y)$. Denoting the variance of the return Φ_N with the initial state x by $\mathbb{V}_x(\Phi_N)$, two assumptions (Kontoyiannis & Meyn, 2003, Section 4) are needed for the total reward CDF estimation.

Assumption 3.1. *The Markov reward process X is geometrically ergodic with a Lyapunov function $V : X \to [1, \infty)$ such that $\zeta(V^2) < \infty$.*

Assumption 3.2. *The (measurable) function $r_S^\pi : S \to [-1, 1]$ has zero mean and nontrivial asymptotic variance $\sigma^2 = \lim_{N\to\infty} \mathbb{V}_x[(\Phi_N)/\sqrt{N}]$.*

Under the two assumptions, we show the Edgeworth expansion theorem for nonlattice functionals (Kontoyiannis & Meyn, 2003, Theorem 5.1) as follows.

Theorem 3.1. *Suppose that X and the strongly nonlattice functional r_S^π satisfy Assumptions 3.1 and 3.2, and let $G_N(y)$ denote the distribution function of the normalized partial sums $(\Phi_N - N\zeta(r_S^\pi))/\sigma\sqrt{N}$:*

$$G_N(y) = \mathbb{P}\{(\Phi_N - N\zeta(r_S^\pi))/\sigma\sqrt{N} \le y\}, \text{ for all } y \in \mathbb{R}.$$

Then, denote the standard normal density by $\gamma(y)$, the corresponding distribution function by $g(y)$, and a constant related to the third moment of $\Phi_N/\sqrt{N}$ by $\kappa \in \mathbb{R}$. For any initial state $x \in S$ and as $N \to \infty$,

$$G_N(y) = g(y) + \frac{\gamma(y)}{\sigma\sqrt{N}}\left[\frac{\kappa}{6\sigma^2}(1 - y^2) - \hat{r_S^\pi}(x)\right] + o(N^{-0.5}).$$

The formulas for κ, $\hat{r_S^\pi}$ and σ^2 can be found in (Xu & Mannor, 2011).

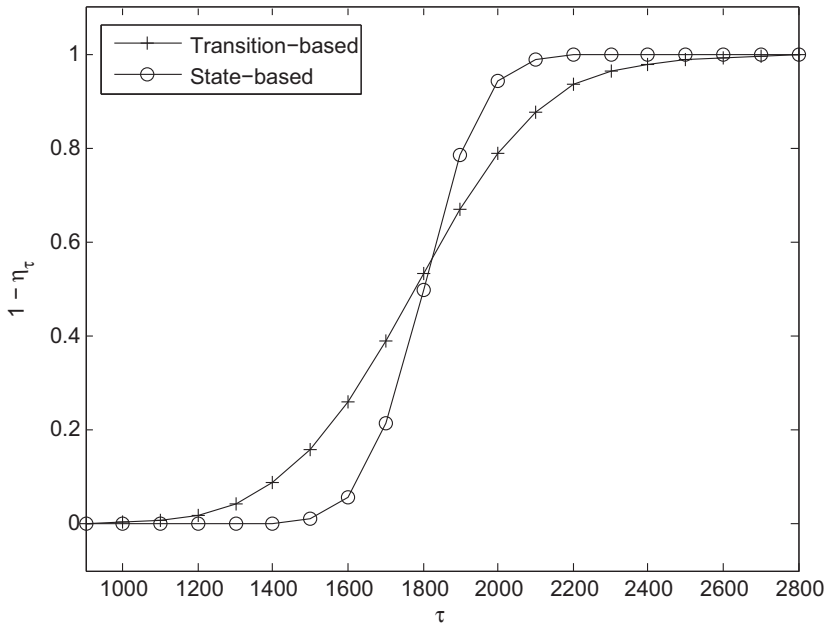

Figure 3.4: Estimated VaR functions for the long-horizon inventory problems with the transformed reward function and the simplified reward function, respectively.

Distribution Estimation with SAT

For a Markov reward process derived from an MDP with a transition-based reward function and a stationary policy π, we simplify the reward function to show the effect of the model simplification. In order to compare the result with the one from the SAT, we transform the Markov reward process with a transition-based reward to one with a state-based reward by Corollary 2.1. Comparing with the result from the model simplification, the proposed transformation can keep the distribution intact.

In a similar MDP setup outlined in the last section, we estimate the VaR function in a long horizon. We set $N = 500$ and implement the SAT for Case 1 to the MDP with every $\pi \in \Pi_d$. In order to reduce the computational complexity, we use $\hat{G}_N(y) = g(y)$ instead of Equation (3.1). In Figure 3.4, we can see that the simplification of the transition-based reward function with Equation (1) results in a distinct VaR function with different values at most nontrivial thresholds ($\tau \in [1300, 1700] \cup [1900, 2300]$ in general). For example, when τ is around 1600, the difference is around 0.2, which may have a serious impact on a risk-sensitive decision making.

Remark 3.2 (VaR in a long horizon)**.** *In a long-horizon MDP, given an estimated VaR function which is strictly increasing, for any probability* $\alpha \in [0, 1]$*, there exists a unique* $\tau \in \mathbb{R}$*, such that* $\alpha = 1 - \eta_\tau$*. In this case, every point along the estimated VaR function can be regarded as a "jump point" described in Remark 3.1. In other words, when the estimated VaR function is strictly increasing, the inversed VaR function solves the VaR Problem 1 with* $\alpha \in [0, 1]$*.*

Till now we have shown that how the SAT preserves the (discounted) total reward distribution in finite-horizon MDPs with VaR objectives, and the model simplification changes the result. Next, we show how the three law-invariant risks are estimated with the aid of the return variance formula in infinite-horizon MDPs.

3.2.4 Risks in Infinite-Horizon MDPs

In this section, we focus on the three law-invariant risks described in Section 1.2, which are functionals on the return $\Phi = \sum_{t=1}^{+\infty} \gamma^{t-1} R_t$. We estimate the three risks with the return variance in infinite horizons, and show how the model simplification leads to wrong results. In the same inventory control MDP, we illustrate how the SAT works for the Case 1 (Corollary 2.1). In a different toy MDP example, we illustrate how the SAT works for the Case 3 (Corollary 2.3), and how the state lumping works.

VaR

The two VaR problems (Problem 1.1 and Problem 1.2) both depend on the VaR function (Equation (7)), which is the infimum of the set of return CDFs. Therefore, we need to estimate the return CDF for every policy. The functional distribution estimations in Makrov reward processes have been studied for decades (Meyn & Tweedie, 2009; Woodroofe, 1992). However, there is no related central limit theorem for the discounted sum of rewards (return). To simplify the estimation, we estimate the return distribution by considering mean and variance only, which implies the assumption that the return is approximately normally distributed.

We consider the MDP described in Section 3.2.1 with an infinite horizon for example. To illustrate how the SAT works for an MDP with a deterministic policy, and show the effect of the model

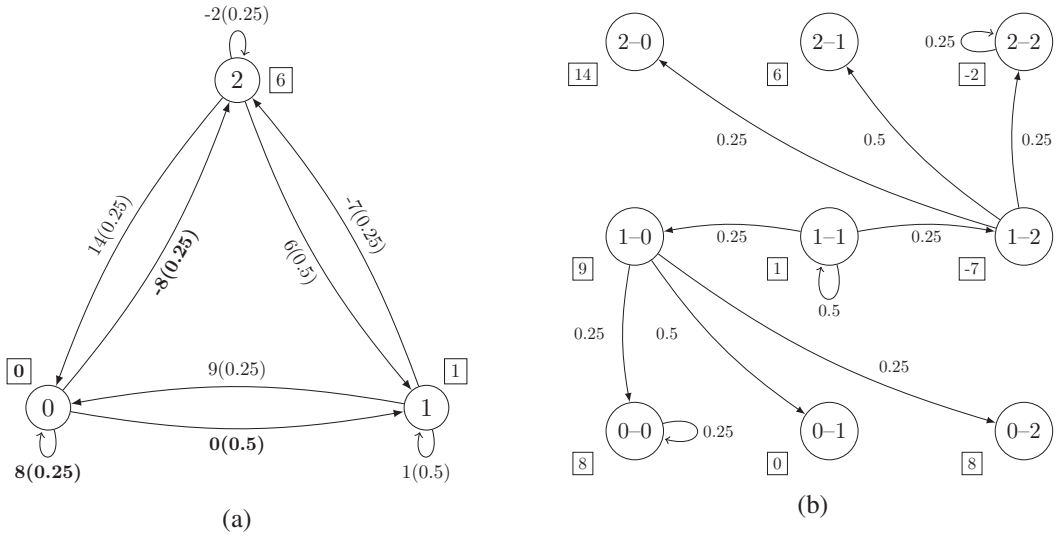

Figure 3.5: (a) The Markov reward process for the MDP in Figure 3.1 with the policy π. This Markov process has an r_T^π. The labels along transitions denote $r_T^\pi(x,y)(p^\pi(x,y))$, and the labels $\boxed{r_S^\pi(x)}$ near states denote the state-based reward simplified with Equation (1); (b) The transformed Markov reward process with a an $r_S^{\pi\dagger}$. For a Markov reward process with an r_T^π, the transformation takes transitions as states and attach each possible reward to a state, in order to preserve the reward sequence. The labels along transitions denote $p^{\pi\dagger}(x^\dagger, y^\dagger)$, and the labels $\boxed{r_S^{\pi\dagger}(x^\dagger)}$ near states denote the state-based reward function from the transformation.

simplification, we give two Markov processes from the model simplification and the SAT, respectively. We consider a deterministic policy $\pi = [2, 1, 0]$—to order 2, 1, 0 item(s) when the inventory level is 0, 1, 2, respectively—then we have a Markov reward process illustrated in Figure 3.5(a), with the simplified reward. To illustrate how the SAT works, we implement the SAT for Case 1, and the transformed Markov process is presented in Figure 3.5(b), with only some of the transitions are shown.

Now we set $\gamma = 0.95$ and compare the two return distributions—one from the SAT for Case 1, and the other from the model simplification—with the averaged empirical return distribution. With the aid of Theorem 1.1, the two distributions are estimated and shown in Figure 3.6. The averaged empirical return distribution is from a simulation repeated 50 times with a time horizon 1000, with the error region representing the standard deviations of the means along return axis.

The Kolmogorov–Smirnov statistic D_{KS} (Durbin, 1973) is used to quantify the distribution difference (error). Denote the averaged empirical return distribution by $\hat{F}_\Phi^\pi$, the estimated distribution

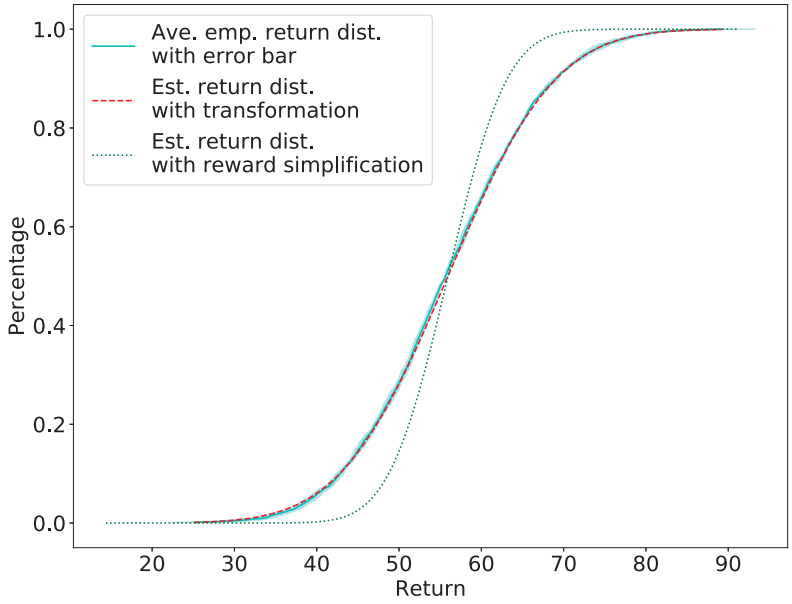

Figure 3.6: Comparison among the averaged empirical return distribution with error region, the estimated return distribution from the transformation, and the estimated return distribution from the model simplification.

for the transformed process by Q^{π}_{Φ}, and the estimated distribution for the process with the model simplification by Q'^{π}_{Φ}. For the case in Figure 3.6, $D_{KS}(Q'^{\pi}_{\Phi}, \hat{F}^{\pi}_{\Phi}) = \sup_{\phi\in\mathbb{R}} |Q'^{\pi}_{\Phi}(\phi) - \hat{F}^{\pi}_{\Phi}(\phi)| \approx 0.145$, and $D_{KS}(Q^{\pi}_{\Phi}, \hat{F}^{\pi}_{\Phi}) \approx 0.012$. The results show that, the model simplification leads to a nontrivial estimation error ($0.145 > 0.012$).

The VaR function is obtained by enumerating the deterministic policies. Figure 3.7 shows the two estimated VaR functions. Since the VaR function can also be regarded as a return distribution, we can still use D_{KS} to measure the error from the model simplification, and in this case $D_{KS} \approx 0.150$. Denote the optimal probability for the MDP with the model simplification by $\hat{\eta}_\tau$, then the error bound for the optimal probability $\sup\{|\hat{\eta}_\tau - \eta_\tau| : \tau \in \mathbb{R}\} \approx 0.150$, which is nontrivial in this risk-sensitive problem.

Remark 3.3 (Smaller variance from model simplification)**.** *In Figure 3.6 we can tell that the distribution for the process with the model simplification has a smaller variance. The reason can be*

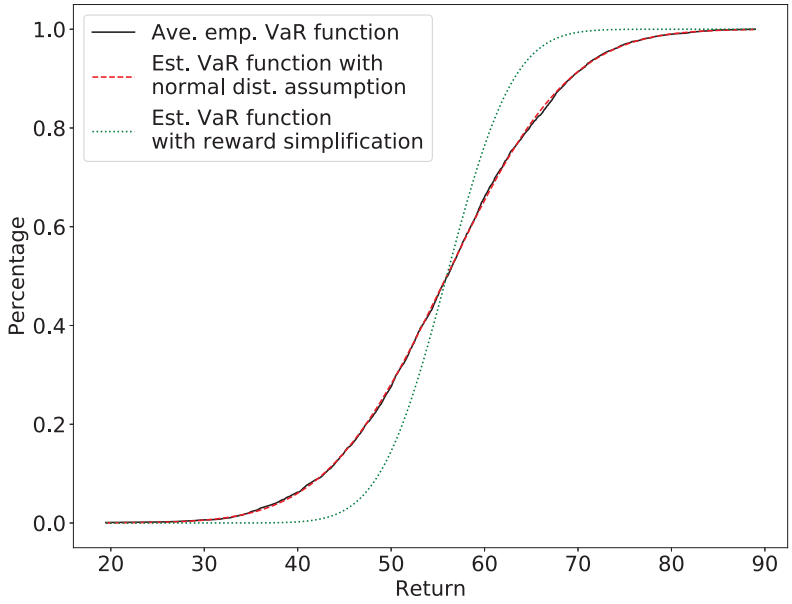

Figure 3.7: Comparison among the averaged empirical VaR function, the estimated VaR function from the transformation, and the estimated VaR function from the model simplification.

intuitively explained by the analysis of variance (Scheffé, 1999). Taking the deterministic transition-based reward function for example. Considering the possible rewards for the same current state as a group, the variance of R_t includes the variances between groups and the variances within groups. When the reward function is simplified with Equation (1), the variances within groups are removed, so the variance is smaller.

From the inventory example we can see that, some methods require Markov processes to be with deterministic and state-based reward functions only, and the model simplification changes the reward sequence as well as the return distribution. If we want to use Q-learning, or other methods for Markov processes with r_S in a risk-sensitive scenario, we should implement an appropriate SAT first. In the next two sections, we consider the other two law-invariant risks—mean-variance risk and exponential utility risk—in an MDP with d_T and a randomized policy, and illustrate how the state lumping alleviates the enlarged state space.

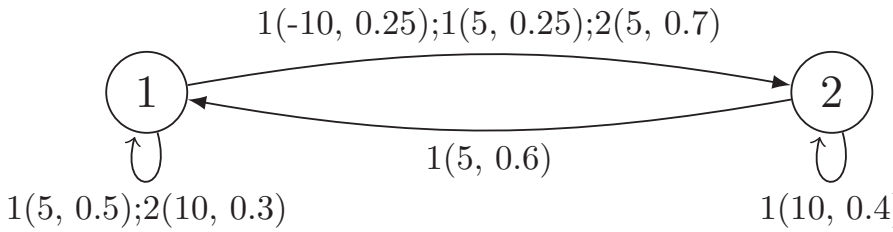

Figure 3.8: A toy example with two states and two actions. The labels $a(j, q)$ along transitions represent the action a, the immediate reward j, and the probability $q = p(y \mid x, a)d_T(j \mid x, a, y)$, where x, y represent the current and the next states, respectively.

Mean-Variance and Exponential Utility Estimations with State Lumping

Here, we illustrate how the SAT works in Case 3 and compare the results with the estimations from the model simplification. Since Case 3 refers to an MDP with a d_T and a randomized policy $\pi \in \Pi_r$, we turn to a smaller example to simplify the graphical representation. Moreover, we also illustrate how the state lumping works.

Consider the process illustrated in Figure 3.8 with two states and two actions, in which state 1 has two actions and state 2 has one. The process starts from state 1 at time $t = 1$. Let's consider the randomized policy $\pi(1) = [0.5, 0.5]$—uniformly choose an action for state 1—then we have a Markov reward process illustrated in Figure. 3.9(a). In order to calculate the return variance, the Markov process needs to have a deterministic reward. One way to achieve this is to naively simplify the reward function by calculating the expectation conditioned on state. The induced Markov process is shown in Figure 3.9(b). The other way to acquire a deterministic reward function is through

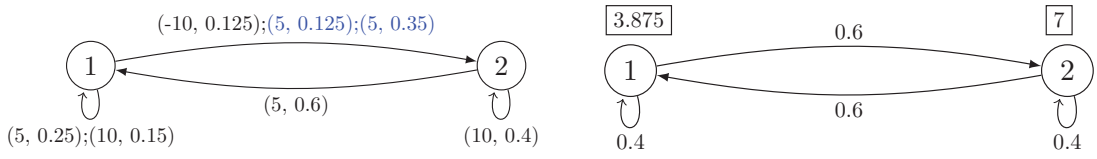

Figure 3.9: (a) A Markov process with a stochastic transition-based reward function. The labels (j, q) along transitions represent the immediate reward j and the probability $q = p^\pi(y \mid x)d_T^\pi(j \mid x, y)$, where x, y represent the current and the next states, respectively. The two blue situations are isotopic states, which can be lumped into one augmented state in the transformed Markov process; (b) The Markov process from the model simplification. The labels p along transitions represent the probability $p = p^\pi(y \mid x)$, and the labels $r(x)$ in the text boxes represent the deterministic state-based reward values from the model simplification.

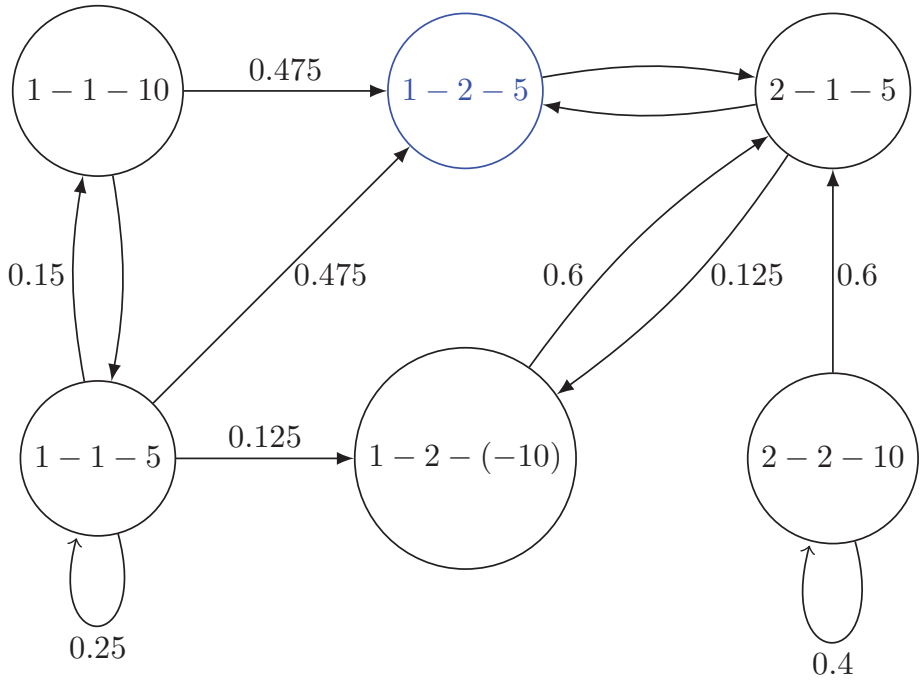

Figure 3.10: The transformed Markov process with a deterministic state-based reward function. Some transitions are hidden. The underlined state comes from lumping two situations from the original Markov process in Figure 2(a).

the SAT. A suitable SAT renders a deterministic reward function and preserve (R_t). However, the SAT also enlarges the state space from $|S|$ to $(|S|^2|A||J| + |S|)$ in general, which is illustrated in Figure 3.10 for this case.

Figure 3.10 illustrates the transformed Markov process with the two states underlined in Figure 3.9(a) lumped into the underlined state $(1-2-5)$. Furthermore, by Corollary 2.5, the two states $(1-1-5)$ and $(2-1-5)$ are isotopic as well, which can be lumped to further simplify the analysis. In summary, by lumping isotopic states into one state, we decrease the size of the augmented state space and keep (R_t) intact. Considering the structural property of the transition probability of the transformed Markov process, it has a fair chance to relieve from the enlarged state space to some degree.

Next, we estimate the two risks on the return with the aid of the SAT. we estimate the exponential utility risk and the mean-variance risk by Equation (5) and (6), respectively. The return variance is calculated by Theorem 1.1. For each risk, we compare three estimations: empirical estimation, estimation from the model simplification, and estimation from the SAT. The empirical estimation

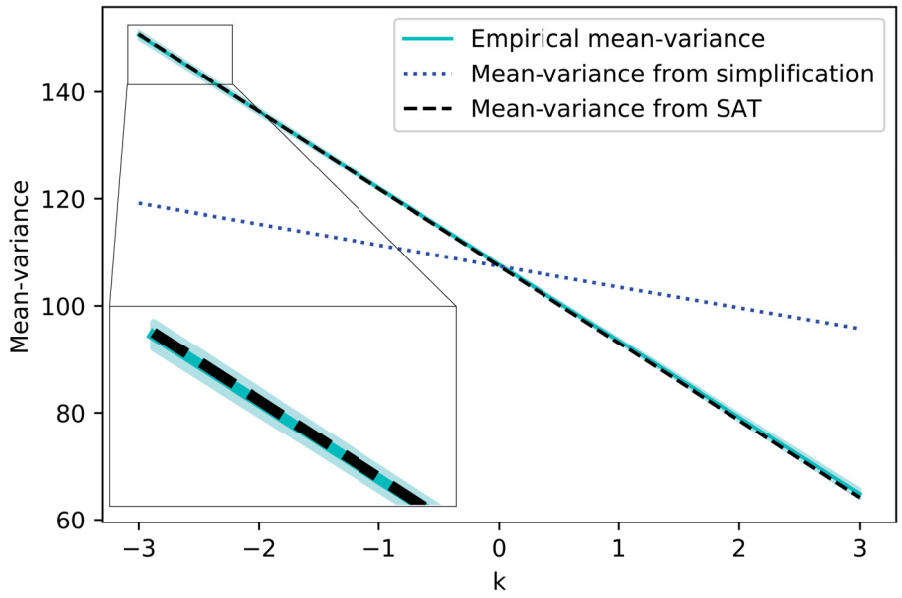

Figure 3.11: The comparison among the empirical mean-variance risk (with an error region), the estimated mean-variance risk from the model simplification and the estimated mean-variance risk from the SAT along the risk parameter $k \in (-3, 3)$.

is calculated as follows. We run $L = 50$ groups of simulations to calculate the variance of an estimation, in each group we run $M = 1000$ simulations of the Markov process, and we set the time horizon $N = 1000$ for the simulations.

Mean-variance risk estimation: The empirical estimation of mean-variance risk with a risk parameter $k \in \mathbb{R}$ is

$$\hat{\Psi}_V(\Phi, k) = \sum_{i=1}^{L} \Psi_{V,i}(k)/L,$$

in which

$$\Psi_{V,i}(k) = \sum_{j=1}^{M} [\phi_{i,j} - k(\phi_{i,j} - \sum_{j=1}^{M} \phi_{i,j}/M)^2]/M,$$

where $\phi_{i,j}$ is the j-th simulation in group $i \in \{1, \cdots, L\}$ and $j \in \{1, \cdots, M\}$. The comparison among the empirical mean-variance risk, the estimated mean-variance risk from the model simplification, and the estimated mean-variance risk from the SAT along the risk parameter $k \in (-3, 3)$ is shown in Figure 3.11. For different k, the empirical mean-variance risk $\hat{\Psi}_V(k)$ is illustrated with an error region representing the standard deviations of the means. Since the error region is so narrow that it is hardly seen, we zoom in a piece to make it clear. Based on the observation, we can see that,

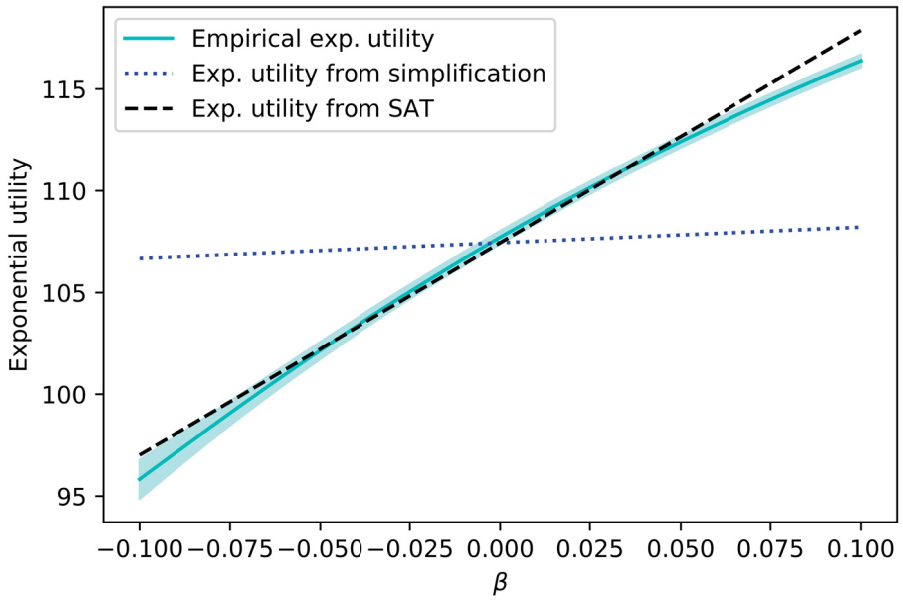

Figure 3.12: The comparison among the empirical exponential utility risk (with an error region), the estimated exponential utility risk from the model simplification, and the estimated exponential utility risk from the SAT along the risk parameter $\beta \in (-0.1, 0.1)$.

the estimated mean-variance risk from the SAT is close to the empirical mean-variance risk, but its counterpart from the model simplification is not. That is because the SAT preserved the return variance, but the model simplification does not.

Exponential utility risk estimation: The empirical estimation of exponential utility risk with a risk parameter $\beta \in \mathbb{R} \setminus \{0\}$ is calculated by

$$\hat{\Psi}_U(\Phi, \beta) = \sum_{i=1}^{L} \Psi_{U,i}(\beta)/L,$$

in which

$$\Psi_{U,i}(\beta) = \beta^{-1} \log[\sum_{j=1}^{M} \exp(\beta \phi_{i,j})/M], \tag{8}$$

where $\phi_{i,j}$ is the j-th simulation in group $i \in \{1, \cdots, L\}$ and $j \in \{1, \cdots, M\}$. The comparison among the empirical exponential utility risk, the estimated exponential utility risk from the model simplification, and the estimated exponential utility risk from the SAT along the risk parameter $\beta \in (-0.1, 0.1)$ is shown in Figure 3.12. It is worth noting that, the utility value at $\beta = 0$ is set by the average of the two adjacent values, since as the denominator in the Equation (8), β cannot be

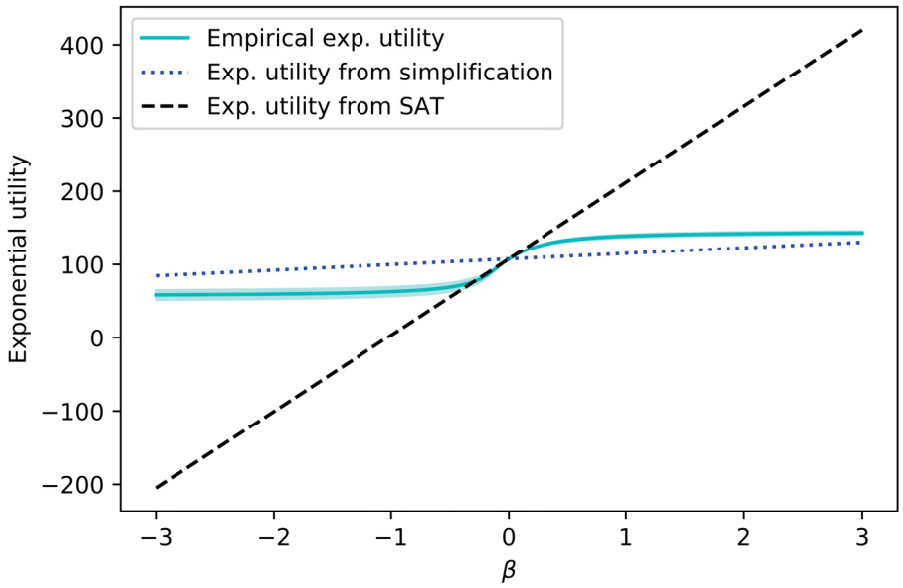

Figure 3.13: The comparison among the empirical exponential utility risk (with an error region), the estimated exponential utility risk from the model simplification, and the estimated exponential utility risk from the SAT along the risk parameter $\beta \in (-3, 3)$.

zero. Based on the observation, we can see that when $\beta \in (-0.1, 0.1)$, the estimated risk from the SAT is close to the empirical one, but its counterpart from the model simplification is not. That is again because the return variance is preserved by the SAT only. It is also found that, for the case $\beta \in (-3, 3)$ in Figure 3.13, the estimated risk from the SAT goes far from the empirical one. That is because there exists a term $\mathcal{O}(\beta^2)$ in Equation (5), and when β deviates too much from zero, this term brings a large error. Therefore, the estimation from the SAT works only when β is close to zero, and its counterpart from the model simplification has large errors in both cases.

Thus far, we have shown that, the model simplification changes the two risk estimations in different cases. There are a number of methods for MDPs and Markov processes with rewards which are not stochastic and transition-based, such as (Guo, Ye, & Yin, 2012; Xia, 2018) on mean-variance risk and (Borkar, 2002; Shen et al., 2014) on exponential utility risk. We believe that, when people apply these methods for practical problems with stochastic and transition-based rewards, they should revisit the methods with the SAT instead of the model simplification. This concern refers to the case when randomized policy is involved as well.

3.3 Related Work

The two VaR problems were defined in (Filar et al., 1995), which studied the VaR problems on average reward by separating state space into communicating and transient classes. VaR concerns the threshold-probability pair, and the optimality of one comes into conflict with the other as they are virtually non-increasing functions of each other. Bouakiz and Kebir (1995) pointed out that the cumulative reward is needed for VaR objectives, and various properties of the optimality equations were studied in both finite and infinite-horizon MDPs. In a finite-horizon MDP, Wu and Lin (1999) showed that the VaR optimal value functions are target distribution functions, and there exists a deterministic optimal policy. The structure property of optimal policy for an infinite-horizon MDP was also studied. For the VaR problem with a probability $\alpha \geq 0.5$, Delage and Mannor (2010) solved it as a convex "second order cone" programming with reward and transition uncertainties. Different from most studies, Boda and Filar (2006) and Kira, Ueno, and Fujita (2012) considered the VaR objective in multi-epoch settings, in which a risk is required to reach an appropriate level at not only the final epoch but also all intermediate epochs.

The VaR problem with a fixed threshold has been extensively studied. An augmented-state 0-1 MDP was proposed for finite-horizon MDPs with either integer or real-valued reward functions (Xu & Mannor, 2011). By including the cumulative reward space into the state space, the states which satisfied the threshold can be "tagged" by a Boolean salvage function. The general reward discretizing error was also bounded (Xu & Mannor, 2011). In a similar problem named MaxProb MDP, the goal states (in which the threshold is satisfied) were defined as absorbing states, and the problem was solved in a similar way (Kolobov, Mausam, Weld, & Geffner, 2011). Value iteration (VI) was proposed to solve the MaxProb MDP in (S. X. Yu, Lin, & Yan, 1998), and followed by some VI variants. In the topological value iteration (TVI) algorithm, states were separated into strongly-connected groups, and efficiency was improved by solving the state groups sequentially (Dai, Weld, & Goldsmith, 2011). Two methods were presented to separate the states efficiently by integrating depth-first search (TVI-DFS) and dynamic programming (TVI-DP) (Hou, Yeoh, & Varakantham, 2014). For both exact and approximated algorithms for VaR with a threshold, the state of the art can be found in (Steinmetz, Hoffmann, & Buffet, 2016). VaR was also considered in practical problems,

Ermon, Gomes, and Vladimirsky (2012) solved a time-sensitive stochastic shortest path problem, in which the route of a delivery truck was planned in order to reach the destination before a strict deadline. They defined a utility function with a worst case constraint, and solved the problem with dynamic programming.

Central limit theorem (CLT) for Markov chain is studied for decades. Most works in this field are for the partial sum of rewards. Under different conditions, the distribution of the partial sum can be estimated (Jones, 2004; Meyn & Tweedie, 2009). Gerber (1971) studied the discounted sum of an infinite sequence of i.i.d. random variables. The first three moments of each random variable were assumed to be bounded, and it was shown that the normalized discounted sum is asymptotically normal ($N(0,(1+v)^{-1})$) when the discount factor v tends to be 1. The difference between the two CDFs were also studied. Stein (1972) presented a bound for the error in the normal approximation to the distribution of the sum of dependent random variables. In the review part it also mentioned several error bounds under different conditions. For example, if every single term in the sum is bounded with exponentially decreasing dependence, then the error is roughly $O(n^{1/4})$. Several studies used different methods to present bounds under different conditions. Woodroofe (1992) developed a sufficient condition for partial sums of a function of a stationary, ergodic Markov chain to be asymptotically normal.

Q-learning has been studied in risk-sensitive RL for decades[2]. Many risk-sensitive Q-learning studies are for MDPs with an r_S. Borkar (2002) proved the convergence of Q-learning for an exponential utility cost objective with an ordinary differential equation method. A trajectory-based algorithm which combines policy gradient and actor-critic methods was presented to solve a CVaR-constrained problem (Chow et al., 2017). For robust MDP problems, with considering a set of general uncertainties (random action, unknown cost and safety hazards), Junges et al. (2016) provided an approach to compute safe and optimal strategies iteratively. Q-learning has also been used to provide risk-sensitive analysis on the fMRI signals, which provides a better interpretation of the human behavior in a sequential decision task (Shen et al., 2014).

For large-scale MDPs with time-consistent risks, P. Yu, Haskell, and Xu (2018) proposed an approximate dynamic programming approach and developed a family of simulation-based algorithms.

[2] As well as methods based on value function, (Borkar & Meyn, 2002) for example.

A unified convergence and sample complexity analysis technique were also given for this set of computationally tractable and simulation-based algorithms. For the whole class of coherent risks, Tamar, Chow, Ghavamzadeh, and Mannor (2015) extended the policy gradient method for both static and time-consistent dynamic risks. For static risk measures, they combined the policy gradient with a standard sampling approach with convex programming. For dynamic risk measures, they proposed an actor-critic algorithm which involved explicit approximation of value function. This unified approach to RL with coherent risks generalized previous related studies. The expectation-based worst case risk might not need the proposed SATs. For example, the minimax risks studied in (Huang & Haskell, 2017). For a comprehensive survey on safe RL, see (García & Fernández, 2015).

3.4 Conclusion and Future Research

With the aid of the SAT, we show that, the three law-invariant risks can be estimated in MDPs with a stochastic transition-based reward and a randomized policy. To relieve the enlarged state space, a novel definition of isotopic states is proposed for the state lumping, considering the special structure of the transformed transition probability. In the numerical experiment, we illustrate the state lumping in the SAT, errors from a naive model simplification, and the validity of the SAT for the three risk estimations. It is worth noting that, the SATs also works for some risks that are not law invariant, such as dynamic risks (Ruszczyński, 2010), since (R_t) is preserved.

Quite a few techniques operate on MDPs with deterministic reward functions, such as Q-learning and the estimation of total reward distribution. However, in many practical scenarios, the reward is stochastic, and the model simplification will change the total reward distribution. The SAT enables the transitions to have properties of states, in order to implement the techniques and keep the distribution intact. We believe that some practical problems with respect to VaR should be revisited using our proposed transformation approach when the reward is complicated.

The essence of the SATs is to attach each possible reward value to an augmented state to preserve the reward sequence. This attachment is crucial in risk-sensitive RL, considering the wide application of value function and Q-function. Without the proposed SATs, when the MDP is with

a reward which is not an r_S type, the direct use of Q-learning also implies such a model simplification. Now with the SATs, the Q-function (value function) can be considered as an approximation of the "real" value of a given augmented state-action pair (state). In other words, the proposed transformations "transform" the uncertainties from the transition, action, and the stochasticity of the reward to the augmented state space. One future work is to efficiently deal with the enlarged state space in the transformed process. Since the state space is augmented, the SATs has an effect on the computational complexity. Denote the complexity for an algorithm by $T(|S|, |A|)$, it becomes $T(|S^2 \times A \times J| + |S|, |A|)$ when the SAT for Case 4 is implemented.

The second future study is to estimate the VaR function without enumerating all the policies. A special case is that there exists an optimal policy π^* for all $\tau \in \mathbb{R}$, i.e., $F_\Phi^{\pi^*}(\tau) = \inf\{F_\Phi^{\pi}(\tau) \mid \pi \in \Pi_n\}$. Ohtsubo and Toyonaga (2002) gave two sufficient conditions for the existence of an optimal policy in infinite-horizon discounted MDPs, and another condition for the unique solution on a transient state set. Another idea is to consider it as a dual-objective optimization. Zheng (2009) studied the dual-objective MDP concerning variance and CVaR, which might provide some insight. A survey on multi-objective MDPs, which concern more than risks, can be found in (Roijers, Vamplew, Whiteson, & Dazeley, 2013). Furthermore, multiple quantile objectives can also be considered as constraints (Randour, Raskin, & Sankur, 2015). A study on multiple long-run average objectives can be found in (Chatterjee, 2007).

The final work could refers to distributional RL. Distributional RL aims to model the distribution over returns in order to evaluate and improve a policy. A successful algorithm, categorical DQN (Bellemare, Dabney, & Munos, 2017), combined a categorical distribution and the cross-entropy loss with the Cramer-minimizing projection. Categorical DQN outperformed all previous improvements to DQN on a set of 57 Atari 2600 games in the Arcade Learning Environment, which we refer to as the Atari-57 benchmark.

Distributional RL algorithms are restricted to assigning probabilities to an a priori fixed, discrete set of possible returns. Dabney, Ostrovski, Silver, and Munos (2018) proposed an alternate pair of choices, parameterizing the distribution by a uniform mixture of Diracs whose locations are adjusted using quantile regression. Their algorithm restricted to a discrete set of quantiles, and automatically adapted return quantiles to minimize the Wasserstein distance between the Bellman updated and

current return distributions.

Dabney et al. (2018) built on recent advances in distributional RL to give a generally applicable, flexible, and state-of-the-art distributional variant of DQN. They achieved this by using quantile regression to approximate the full quantile function for the state-action return distribution. By reparameterizing a distribution over the sample space, this yields an implicitly defined return distribution and gives rise to a large class of risk-sensitive policies.

Zhang and Yao (2019) proposed the Quantile Option Architecture (QUOTA) for exploration based on recent advances in distributional RL. QUOTA provides a new dimension for exploration via making use of both optimism and pessimism of a value distribution. However, the decision making in prevailing distributional RL methods is still based on the mean (Bellemare et al., 2017). They focused on making decisions based on the full distribution and whether an agent can benefit for better exploration. They proposed to select an action greedily w.r.t. certain quantile of the action value distribution. A high quantile represents an optimistic estimation of the action value, and action selection based on a high quantile indicates an optimistic exploration strategy. QUOTA adaptively selects a pessimistic and optimistic exploration strategy, resulting in improved exploration consistently across different tasks.

Chapter 4

A Scheme for Dynamic Risk-Sensitive Sequential Decision Makings with Constraints

4.1 Introduction

Risk-sensitive criteria have been drawing more attention in practical problems, especially in which the small probability events have serious consequences. When various goals are targeted in practice, one solution is to optimize one of them as objective and take others as constraints. Furthermore, the environment might vary over time, which makes it more complicated. In this chapter, we consider the sequential decision making (SDM) problems with three concerns: risk, constraint, and dynamic environment. In order to solve this problem, we give a constrained and risk-sensitive model, sample the variables involved within specified intervals, and propose a scheme for generating a synthetic dataset and training an approximator (here we use a neural network for example).

The motivation for the proposed scheme arises from practical problems in which exact analytic risk analysis may be excessively complicated or prohibitively expensive, especially in a dynamic environment. Though historical data can be used for neural network (NN) training, in many practical

situations, the recorded decisions are not optimal. Furthermore, in risk-sensitive cases, the criteria based on which the decisions were made are not clarified. Therefore, we generate a synthetic dataset for problems with a specified risk-sensitive objective and constraints. For a given parameter set, we construct an MDP, calculate the return variance for any deterministic policy, evaluate (or estimate) the involved risks to check constraint satisfaction, and record the optimal policy.

A practical inventory control problem is considered for example, in which the risk objective and constraints can be evaluated or estimated with return variance, as shown in the previous chapter. The environment of an inventory control problem is dynamic in the long run. For example, the wholesale prices are changing all the time, as well as the uncertainties of the market demands and supplier reliabilities. How to adapt to these dynamics efficiently from a risk perspective is not only an academic concern, but also a business one.

4.1.1 Chapter Contribution

In this chapter, we propose a scheme for solving risk-sensitive SDM problems with constraints in a dynamic environment. we focus on law-invariant risks, as well as other functions of the return variance $\mathbb{V}(\Phi)$. As presented in the previous chapter, the three law-invariant risks can be either calculated (mean-variance risk), estimated (utility risk), or estimated with assumption (quantile-based risk) with $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$. By virtue of the SAT, $\mathbb{V}(\Phi)$ can be calculated in a Markov process with a stochastic reward, which allows a risk evaluation in practical SDM problems. For a given risk-sensitive problem, parameters defining a targeted process might be dynamic, i.e., they might vary over time, so we sample them within specified intervals to deal with these dynamics. As for the risk-sensitive criterion, considering the objective and constraints are, or can be estimated by, functions of the mean and variance of return, we enumerate all $\pi \in \Pi_d$ to estimate the risk objective and check the risk constraints. Thus, we can now generate a synthetic dataset as training data by virtue of the proposed SAT.

We use MDPs to model SDM problems for example, and it enables us to evaluate risks as shown in previous chapters. An NN is trained as an approximator of the mapping from risk and environment parameter spaces to space of risk and policy spaces with respect to risk-sensitive constraints. We show that, i) practical problems modeled by MDPs with stochastic rewards can be solved in a

risk-sensitive case; and ii). the proposed scheme is validated by a numerical experiment.

4.1.2 Chapter Organization

This chapter is structured as follows. In Section 4.2, we give a constrained and risk-sensitive decision-making model, and propose a scheme for solving it in a dynamic environment with NN as the approximator for example. In Section 4.3, we present a practical inventory control problem for illustrating the validity of the scheme. In Section 4.4, we give the literature review on constrained and risk-sensitive SDMs in dynamic environments. In Section 4.5, we make conclusions and discuss the future research.

4.2 Sequential Decision Making Scheme and Approximator

In this section, we propose a scheme for solving risk-sensitive SDM problems with constraints in a dynamic environment. We use MDP to model it and NN as the approximator in the scheme for example.

4.2.1 Sequential Decision Making Scheme

Taking advantage of the growing computational power, "big data" techniques have been drawing attentions for decades. People from all walks of life value their historical data and use it to automate decision-making processes. However, there are several problems in the risk-sensitive decision-making automation.

Question 4.1. *Is the risk criterion in the decision making fixed all the time?*

The answer to this question can be negative in many cases, since a decision maker can be fickle, and there could be multiple decision makers accounting for the historical data.

Question 4.2. *Is the risk criterion in the decision making well defined?*

The answer to this question can be negative in most cases, since a criterion description (a mathematical one or not) is usually missing in historical data.

Question 4.3. *Are the decisions optimal?*

Unfortunately, the answer to this question can be often negative as well. Consider an inventory control problem for example. The manager could simply adopt the EOQ (economic order quantity) strategy since it is easy to use, though the situation does not meet the assumptions of the EOQ model. To solve this problem for automated risk-sensitive decision making, we propose a scheme for solving risk-sensitive SDM problems with constraints in a dynamic environment.

A dynamic, constrained, and risk-sensitive SDM problem can be considered as a process (environment) with a criterion (objective with constraints). Since the future environment is unknown, we consider the problem in a "static" environment with the current related parameters to evaluate the criterion at any epoch, and thus it becomes a one-step decision-making problem. We represent the process of the problem by a vector of $n \in \mathbb{N}^+$ environmental features $\mathbf{q_e} \in Q_e = F_1 \times \cdots \times F_n$. Similarly, denote the parameters involved in the criterion by a vector of $\mathbf{q_p} \in Q_p$, and it may contain β in Equation 5 for exponential utility risk, k in Equation 6 for mean-variance risk, α in Problem 1.1, τ in Problem 1.2 for VaR, and other coefficients and factors. It worth noting that, we allow $\mathbf{q_p}$ to change within the prespecified Q_p for dynamic concern as well as $\mathbf{q_e} \in Q_e$.

With all the parameters defined, we can model this one-step decision-making problem into an infinite-horizon MDP (from $\mathbf{q_e}$) with a criterion $\langle z_0, \mathbf{z} \rangle = \langle z_0, \mathbf{z} \rangle(\mathbf{q_p}) = \langle z_0(\cdot, \cdot; \mathbf{q_p}), \mathbf{z}(\cdot, \cdot; \mathbf{q_p}) \rangle$. $z_0(\cdot, \cdot; \mathbf{q_p})$ and $\mathbf{z}(\cdot, \cdot; \mathbf{q_p})$ are functions of expectation and variance of return, which are denoted by $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$, respectively. Recall that Π_d is the deterministic policy space, the return (or the discounted total reward) in an infinite-horizon MDP with a policy $\pi \in \Pi_d$ is defined by $\Phi^\pi = \sum_{t=1}^{+\infty} \gamma^{t-1} R_t$ (for details see Section 1.1), and here we denote it by Φ for brevity. Also note that the immediate reward R_t is not the one in the dynamic SDM process, but the one in the MDP modeled with $\mathbf{q_e}$ at any epoch.

For any given $\mathbf{q_p} \in Q_p$, we have $z_0(\cdot, \cdot; \mathbf{q_p}) : \mathbb{R}^2 \to \mathbb{R}$ as an objective function on $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$, and $\mathbf{z}(\cdot, \cdot; \mathbf{q_p}) : \mathbb{R}^2 \to \mathbb{R}^m$ as a m-dimensional vector of constraint functions on $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$, where $m \in \mathbb{N}^+$. Hence, a constrained and risk-sensitive SDM problem at any given epoch

can be formulated as

$$\begin{aligned} \min_{\pi \in \Pi_d} \quad & z_0 \\ \text{s.t.} \quad & \mathbf{z} \leq \mathbf{0}. \end{aligned}$$

Denote the allowable policy space by $\Pi_c = \{\pi \in \Pi_d \mid \mathbf{z}(\mathbb{E}(\Phi^\pi), \mathbb{V}(\Phi^\pi); \mathbf{q_p}) \leq \mathbf{0}\}$, and then the optimal policy

$$\pi^* \in \operatorname*{argmin}_{\pi \in \Pi_c} z_0(\mathbb{E}(\Phi^\pi), \mathbb{V}(\Phi^\pi); \mathbf{q_p}),$$

with the optimal risk $q_o = z_0(\mathbb{E}(\Phi^{\pi^*}), \mathbb{V}(\Phi^{\pi^*}); \mathbf{q_p}) \in Q_o$. Thus, a decision maker can be mathematically denoted by

$$\nu : Q_e \times Q_p \to Q_o \times \Pi_d,$$

i.e., a theoretically perfect decision maker consider all the environmental parameters, the objective, and possible constraints, and choose the optimal policy with respect to a specified risk-sensitive criterion. The goal is to train an NN to approximate ν.

Usually, historical data is used in training an approximator. However, as we describe above, at least two problems should be considered in practice. The first problem refers to information incompleteness. Most historical data contains no information on $\mathbf{q_p}$ and (or) π^*, and in many cases, even the information on $\mathbf{q_e}$ is incomplete. The second problem is about optimality. In practice, the decision makers might prefer an easy-to-use policy than an optimal one, which is hard to determine since the practical problems could be different from the theoretical model diversely and subtly. We therefore propose that, for any $\mathbf{q_e} \in Q_e$ and $\mathbf{q_p} \in Q_p$, we should evaluate or estimate involved risks and calculate q_o and π^* to generate a synthetic dataset, and then use it to train the NN.

We present a scheme in Figure 4.1 to solve a constrained risk-sensitive SDM problem in a dynamic environment. In the feature representation module, we define the SDM process with $\mathbf{q_e}$. In the SDM modeling module, a mathematic model is chosen to represent the process and offer a platform for risk evaluation, and here we use MDP for example. In the risk evaluation module, we calculate the optimal policy by evaluating all deterministic policies with a specified risk-sensitive criterion. For a given MDP with a risk-sensitive criterion, we firstly enumerate all deterministic policies to generate Markov processes. Secondly, we implement the SAT to transform every Markov process to one with a deterministic reward, and calculate the return variance. Thirdly, we estimate

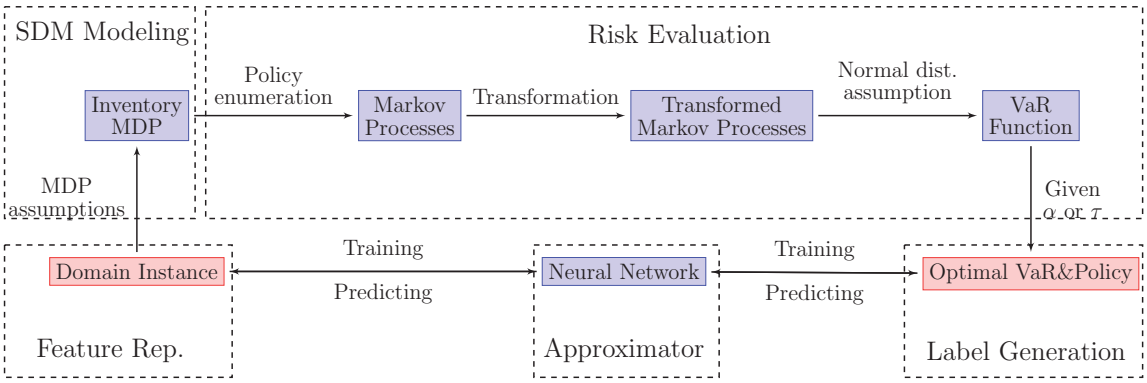

Figure 4.1: A dynamic risk evaluation scheme with NN and RL methods for optimal risk and policy in a sequential decision making problem.

the related risk(s) with the return variance, and here we consider VaR as the objective and a function of return mean and variance as a constraint only. In the label generation module, the optimal VaR and policy is recorded and attached to the input as a label. Finally, a functional approximator, such as an NN or a linear combination of polynomial basis or Fourier basis, is trained with the labeled feature dataset. In this chapter, we consider an inventory control problem for example.

Briefly, the upper level of the scheme is for synthetic dataset generation (modeling and risk evaluation), and the lower level is for approximator training. After each decision-making epoch, some parameters will be updated with the outcome (such as market demand distribution) and the involved external variables (such as wholesale price). Suppose the updated parameters are still within the predefined intervals, the trained NN is able to output the estimated optimal policy and risk at the next decision-making epoch. We present Algorithm 5 for the synthetic dataset generation (upper level of the scheme) as follows.

4.2.2 Approximator

A functional approximator needs training with a dataset to approximate the mapping from parameter space to risk and policy space. In this study, we choose to use an NN, which is a universal function approximator (Hornik, 1991). Other approximators, such as a linear combination of polynomial basis or Fourier basis, are not considered here. The definition of NN is as follows.

Definition 4.1 (Feed forward neural network)**.** *Let $Q \in \mathbb{N}^+$ be the number of layers, and*

Algorithm 5 Synthetic Dataset Generation Algorithm

Input: the size of training dataset $|G|$, environmental feature space Q_e, risk parameter space Q_p, and the criterion $\langle z_0, \mathbf{z}\rangle$.

Output: the training dataset G.

for all $i \in \{1, \cdots, |G|\}$ **do**
  Randomly generate a environmental feature vector $\mathbf{q_e} \in Q_e$ and a risk parameter vector $\mathbf{q_p} \in Q_p$;
  Construct an MDP $\mathcal{M}$ from $\mathbf{q_e}$;
  for all $\pi \in \Pi_d$ **do**
    Get the Markov process $\mathcal{M}^\pi$ from $\mathcal{M}$ and π; ▷ $\mathcal{M}^\pi$ has a stochastic transition-based reward
    Get the transformed Markov process $\mathcal{M}^{\pi\dagger}$ using the SAT; ▷ $\mathcal{M}^{\pi\dagger}$ has a deterministic state-based reward
    Calculate the return variance in $\mathcal{M}^{\pi\dagger}$;
    Evaluate the criterion $\langle z_0, \mathbf{z}\rangle(\mathbf{q_p})$;
    if π does not satisfy the constraint(s) $\mathbf{z}(\mathbf{q_p})$ **then**
      continue;
    else
      record the risk value $z_0(\mathbf{q_p})$ and policy π;
    end if
  end for
  Get the optimal risk value ρ^* and the corresponding optimal policy π^*;
  Record the i-th sample point as (k_i, l_i), with $k_i = (\mathbf{q_e}, \mathbf{q_p})$ and $l_i = (\rho^*, \pi^*)$);
end for

$N_0, N_1, \cdots, N_Q \in \mathbb{N}^+$ be the numbers of nodes of different layers. Denote the activation function by $g : \mathbb{R} \to \mathbb{R}$. For any $q \in \{1, \cdots, Q\}, x \in \mathbb{R}$, denote the affine function by $W_q(x) = A_q x + b_q$ for some $A_q \in \mathbb{R}^{N_q \times N_{q-1}}$ and $b_q \in \mathbb{R}^{N_q}$. For any $i \in \{1, \cdots, N_q\}, j \in \{1, \cdots, N_{q-1}\}$, and $q \in \{1, \cdots, Q\}$, the entry $A_{q,i,j}$ of A_q denotes the weight of the edge from the i-th node in the $(q-1)$-th layer to the j-th node in the q-th layer. For any layer $q \in \{1, \cdots, Q\}$, let $W_q : \mathbb{R}^{N_{q-1}} \to \mathbb{R}^{N_q}$ be an affine function. With $H_q = g \circ W_q$ for $q = 1, \cdots, Q-1$, a function $H : \mathbb{R}^{N_0} \to \mathbb{R}^{N_Q}$ defined as

$$H = W_Q \circ H_{Q-1} \circ \cdots \circ H_1$$

is called a feed forward NN. Since we use NN as an estimator of the decision maker ν, we denote the NN by $\hat{\nu}$, and the set of NNs mapping from $\mathbb{R}^{N_0}$ to $\mathbb{R}^{N_Q}$ with g by $\{\hat{\nu}^g_{\infty,N_0,N_Q}\}$.

In this study, for any given $\mathbf{q_e}$, we construct an MDP model, and by virtue of the SAT, we can preserve the variance for a Markov process with a stochastic reward. As claimed in Section 1.2, most inherent risks can be evaluated or estimated by return variance with or without some assumption. We calculate q_o and π^* and attached them to $\mathbf{q_e}$ and $\mathbf{q_p}$ as labels. Finally, we train an NN with the synthetic dataset. In the next section, we present an inventory management problem as an example

to show the implementation details.

4.3 Numerical Experiment

Figure 4.1 illustrates the scheme with an NN and RL methods for a dynamic risk-sensitive SDM problem with a VaR objective and risk-sensitive constraints. The scheme includes synthetic dataset generation (upper layer), approximator training and predicting (lower layer). To generate the dataset for training and testing according to Algorithm 5, we define and randomly generate a domain set. For each domain instance, we estimate the optimal VaR and the corresponding policy as follows. Firstly, construct an inventory MDP under some predefined assumptions. Secondly, enumerate all deterministic policies to achieve a set of Markov reward processes $\{\langle S, J^\pi, p^\pi, d_T^\pi, \mu\rangle\}_{\pi\in\Pi_d}$. Thirdly, acquire the set of transformed Markov process $\{\langle S^\dagger, r_S^{\pi\dagger}, p^{\pi\dagger}, \mu^\dagger\rangle\}_{\pi\in\Pi_d}$ by the SAT, with $r_S^{\pi\dagger}$ being deterministic and state-based. Fourthly, estimate the mean and variance vectors υ, ψ by Theorem 1.1. Assuming the return is approximately normally distributed, we have the estimated return distributions for all $\langle S^\dagger, r_S^{\pi\dagger}, p^{\pi\dagger}, \mu^\dagger\rangle$ to calculate the VaR function for the inventory MDP. Finally, for the risk sensitivity parameters in the domain instance, check the constraints, calculate risk-sensitive objective, and record π^*. With the training dataset, we tune, train and validate an NN properly, and use it to predict q_o and π^* for any $\mathbf{q_e} \in Q_e$ and $\mathbf{q_p} \in Q_p$.

4.3.1 An Inventory Management Example

As a proof of principle for the applicability of the proposed scheme, we apply it with an NN to a practical inventory control process for example, in which three factors are modeled: multiple sourcing, supplier reliability, and backlog. It is worth noting that MDP is a discrete-time process, so the output optimal policy is a periodic-review strategy, which has its limitations comparing with its aperiodic-review counterparts. Furthermore, some assumptions are made to keep the model small. Hence, this inventory control SDM model is theoretical, and in many cases a more accurate estimation can be achieved through simulation with respect to complicated cases.

Multiple sourcing

Multiple sourcing (multisourcing) is a strategy that blends services from the optimal set of internal and external suppliers in the pursuit of business goals (Cohen & Young, 2006). We consider two suppliers U_1 and U_2, and a retailer R. The scenario involves a multinational corporation (R) which functions in the supply chain as a retailer. For a targeted product, this corporation has two suppliers U_1 and U_2 (in US and in India, for example). The supplier U_1 in US is close and reliable, the other supplier U_2 is far and unreliable but with a lower wholesale price.

Supplier reliability

Supplier (sourcing) reliability can be expressed in terms of quantity, quality or timing of orders due to multiple factors, such as equipment breakdowns, material shortages, warehouse capacity constraints, price inflations, strikes, embargoes, and political crises (Mohebbi, 2004). In the framework of MDP, one way to model these uncertainties is to take the status of a supplier as a Boolean variable, which represents whether the supplier is "available" or "unavailable", and it can be modeled as a two-state Markov process (Ahiska, Appaji, King, & Warsing Jr, 2013). In order to simplify the state representation, we assume that the unreliable supplier U_2 is available with a probability $\beta_1 \in [0, 1]$, and the retailer will be compensated with an amount $p_s \in \mathbb{R}^+$ if the supplier with an order is unreliable at the current epoch. The probability β_1 can be estimated using historical data.

Transportation can be another source of unreliability. It emphasizes the long delivery time which mainly results from a long physical distance. This uncertainty can be also a critical factor at times. In our model, we assume U_1 is reliable and its transportation lead time can be ignored (for example, a day when one epoch is a month). Assume U_2 offers a lower price but unreliable. Since it is far from R, its lead time is one period. To simplify the model, we consider the transportation factor within the supplier reliability.

Backlog and lost sale

For each unsatisfied unit of demand, a backlog or a lost sale occurs. Suppose the probabilities of backlog and lost sale are $\beta_2 \in [0, 1]$ and $1 - \beta_2$, respectively. The probability β_2 can be estimated

using historical data. We assume that when a backlog happens, the product will be sold at a lower price $p_r - c_b$ (which can be due to from fast delivery or other reasons), where $p_r \in \mathbb{R}^+$ is the retail price and $c_b \in (0, p_r)$. A lost sale induces a cost $c_l \in (0, p_r)$. Rewards for both cases will be calculated at the current epoch, in order to keep the state representation small. The backlog and lost sale instances are associated with market demand. To keep the stochastic reward function simple, finite possible demands are considered (i.e., a finite support for the reward function). For example, when the warehouse capacity is four units and the maximum demand is six units, then the possible maximum amount of backlogs or lost sales is two units. In the MDP model, the reward and cost from these two units will be received at a discount at the current epoch.

To simplify the model, other factors, such as fast delivery, order replacement and foreign currency exchange, have not been considered. The inventory control process is illustrated in Figure 4.2, in which the solid arrows represent the product flow, and the dashed arrows represent the order flow. Next, we construct an MDP for this inventory control problem.

4.3.2 An Inventory MDP

Here we use an infinite-horizon MDP with finite state and action space to model the inventory control process. The related parameters are given in Table 4.1. For a set of specified parameters, we construct the MDP as follows. Given the warehouse capacity $M \in \mathbb{N}^+$, for any epoch $t \in \mathbb{N}^+$, denote the inventory level by $I_{1,t} \in \{0, \cdots, M\}$, and the number of items in transit $I_{2,t} \in \{0, \cdots, M - I_{1,t}\}$, then the state is $X_t = (I_{1,t}, I_{2,t}) \in S$, where $S = \{(i_1, i_2) \in \{0, \cdots, M\}^2 \mid i_1 + i_2 \leq M\}$. The action $K_t = (K_{1,t}, K_{2,t}) \in A_{X_t}$ represents the order quantities with the two

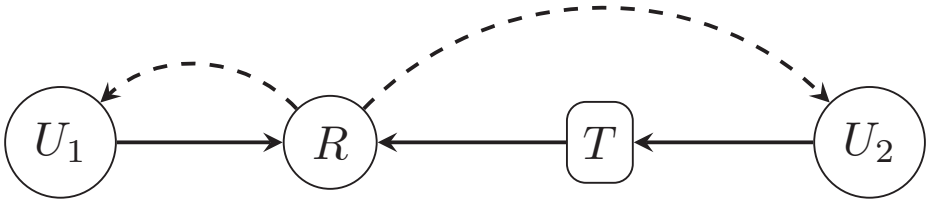

Figure 4.2: The product (solid) and order (dashed) flows between the retailer R and the two suppliers U_1 and U_2. The letter T denotes transportation. The transportation lead time of U_1 is negligible, hence ignored.

Table 4.1: Inventory control parameter list

$M \in \mathbb{N}^+$	warehouse capacity
$\gamma \in [0,1]$	Discount factor
$\mu : S \to [0,1]$	Initial state distribution
$p_r \in \mathbb{R}^+$	Retail price
$p_s \in \mathbb{R}^+$	Penalty on supplier's unavailability
$c_1, c_2 \in \mathbb{R}^+$	Wholesale price from U_1 and U_2
$c_b \in \mathbb{R}^+$	Backlog cost
$c_l \in (0, p_r)$	Lost sale cost, which could be a dependent variable
$c_f \in \mathbb{R}^+$	Fixed ordering cost
$c_h \in \mathbb{R}^+$	Holding cost
$f_D : \mathbb{N} \to [0,1]$	Demand distribution function (PMF)
$\beta_1 \in [0,1]$	Supplier availability probability
$\beta_2 \in [0,1]$	Backlog probability
$\alpha \in [0,1]$	Risk sensitivity parameter (for VaR)

suppliers, where $A_{X_t} = \{(k_1, k_2) \in \{0, \cdots, M\}^2 \mid k_1 + k_2 \leq M - I_{1,t} - I_{2,t}\}$. Assume that the maximum demand could be double of the warehouse capacity, i.e., $\text{supp}(f_D) = \{0, \cdots, 2M\}$, and the demand is $D_t \in \text{supp}(f_D)$ with probability $f_D(D_t)$. Given $\beta_1, \beta_2 \in [0,1]$ to represent supplier availability and backlog probabilities, respectively, the reward and transition probability can be formulated.

The reward and transition probability formulations are based on "situation" (check Remark 2.1). For every $X_t \in S$ and $K_t \in A_{X_t}$, we consider two factors. Firstly, is U_2 reliable? Secondly, can the demand be satisfied? If not, how many of them is backlogged? By enumerating all possible situations for all $X_t \in S$ and $K_t \in A_{X_t}$, we construct stochastic transition-based reward and the transition probability. For example, given the current state $X_t = (I_{1,t}, I_{2,t}) \in S$, the action $K_t = (K_{1,t}, K_{2,t}) \in A_{X_t}$, one situation could be that, the demand is $D_t \leq I_{1,t} + I_{2,t} + K_{1,t}$, and U_2 is available currently. In this situation, the reward is

$$R_t = p_r \cdot D_t - c_f \cdot (\mathbb{1}_{[K_{1,t}>0]} + \mathbb{1}_{[K_{2,t}>0]}) - (c_1 \cdot K_{1,t} + c_2 \cdot K_{2,t}) - c_h \cdot I_{1,t}.$$

The probability of this situation is given by

$$q = \mathbb{P}(X_{t+1},\ \text{`}U_2 \text{ is available'} \mid X_t, K_t) = f_D(D_t) \cdot \beta_1,$$

and the next state $X_{t+1} = (I'_{1,t}, K_{2,t})$, where $I'_{1,t} = I_{1,t} + I_{2,t} + K_{1,t} - D_t$.

For the same X_t and K_t, consider the situation that U_2 is unavailable, $D_t = I_{1,t}+I_{2,t}+K_{1,t}+x$, $x \in \mathbb{N}^+$, i.e., the demand is larger than the supply, and $y \in [0, x]$ units are backlogged. In this case, the reward is

$$R_t = p_r \cdot (D_t - x + y) - c_f \cdot (\mathbb{1}_{[K_{1,t}>0]} + \mathbb{1}_{[K_{2,t}>0]}) - (c_1 \cdot K_{1,t} + c_2 \cdot K_{2,t}) - c_h \cdot I_{1,t} + p_s \cdot K_{2,t} - y \cdot c_b - (x-y) \cdot c_l,$$

and the probability of this situation is

$$q = \mathbb{P}(X_{t+1},\ \text{`}U_2 \text{ is unavailable', `y backlog', `(x-y) lost sale'} \mid X_t, K_t) = f_D(D_t) \cdot (1-\beta_1) \cdot \beta_2^y \cdot (1-\beta_2)^{x-y},$$

and the next state $X_{t+1} = (0, K_{2,t})$. For any $X_t \in S$ and $K_t \in A_{X_t}$, denote the number of possible situations by N_{X_t,K_t}. For the z-th situation, where $z \in \{1, \cdots, N_{X_t,K_t}\}$, we denote the related results by $(R_{t,z}, q_z, X_{t+1,z})$ as the reward, the situation probability and the next state, respectively. Then for any $X_t \in S$ and $K_t \in A_{X_t}$, by enumerating all possible situations, we have the set of results $\{(R_{t,z}, q_z, X_{t+1,z}) \mid z \in \{1, \cdots, N_{X_t,K_t}\}\}$. The transition probability conditioned on X_t and K_t is

$$p(X_{t+1} \mid X_t, K_t) = \sum_{z \in \{1,\cdots,N_{X_t,K_t}\}} q_z \cdot \mathbb{1}_{[X_{t+1,z}=X_{t+1}]},$$

and the stochastic transition-based reward is

$$d(j \mid X_t, K_t, X_{t+1}) = \sum_{z \in \{1,\cdots,N_{X_t,K_t}\}} q_z \cdot \mathbb{1}_{[X_{t+1,z}=X_{t+1}, j=R_{t,z}]}.$$

Next, we consider a VaR objective with a risk-sensitive constraint, and generate the synthetic training data for NN.

4.3.3 Dataset Generation

In this study, the risk-sensitive criteria can be composed of any functions of $\mathbb{E}(\Phi)$ and $\mathbb{V}(\Phi)$. Here, we consider the VaR Problem 1.1 as the objective with a constraint $\mathbb{E}(\Phi)/\mathbb{V}(\Phi) > q$ for example, where $q \in Q_p$. The intuitive meaning of this constraint is that the earning per unit of risk (variance) should be larger than a threshold q. To simplify the problem, we set $q = 5$ as a constant.

Next, we artificially generate a dataset for the inventory control process with this criteria, in which $\mathbf{q_p} = \alpha \in Q_p$, and Q_p is a prespecified set.

There are multiple ways to define a mapping from parameter space to risk and policy space. Here we define it as follows. Define the domain set $\mathcal{K}$, with each instance $k \in \mathcal{K}$ refering to a vector of inventory control parameters (retail price, fixed ordering cost, etc.). Define the label set $\mathcal{L}$, with each instance $l \in \mathcal{L}$ refering to a vector of labels (optimal VaR, optimal policy, etc.). A dataset $G = \{(k_1, l_1), \cdots, (k_m, l_m)\}$ is a finite set of pairs in $\mathcal{K} \times \mathcal{L}$, i.e., a set of labeled domain instances.

It worth noting that, to construct an NN one needs to predefine certain parameters, including but not limit to Q, $\{N_i\}$ and g (see Section 4.2.2 for details). The predefined parameters, which determine the network structure and how the NN is trained, are called hyperparameters. A fine-tuned NN with a large enough dataset should be able to make a satisfactory prediction. For our inventory control problem, all related parameters are set and sampled as follows.

Hyperparameters: A domain instance includes all variable values except for warehouse capacity $M \in \mathbb{N}^+$, discount factor $\gamma \in [0, 1]$, and initial inventory distribution $\mu : S \to [0, 1]$, which are considered as hyperparameters. We set $M = 2$, $\gamma = 0.95$, and $\mu((0, 0)) = 1$, i.e., at the beginning the inventory level and the item amount in transit are both zero. The latter two parameters are fixed only to simplify the problem.

Revenue related parameters: Set the retail price $p_r \in P_r = [6, 10]$, and for generating the dataset, we take it as a random variable, whose distribution is $U(6, 10)$ [1]. Similarly, set the penalty on supplier's unavailability $p_s \in P_s = [0, 2]$ with $p_s \sim U(0, 2)$, the wholesale prices from U_1 and U_2 are $c_1 \in C_1 = [4, 6]$ with $c_1 \sim U(4, 6)$ and $c_2 \in C_2 = [1, 4]$ with $c_2 \sim U(1, 4)$, the backlog cost $c_b \in C_b = [0, 2]$ with $c_b \sim U(0, 2)$, the fixed ordering cost (for both supplier) $c_f \in C_f = [0, 2]$ with $c_f \sim U(0, 2)$, and the holding cost $c_h \in C_h = [0, 2]$ with $c_h \sim U(0, 2)$. For brevity, we set the lost sale cost $c_l = p_r - c_2$.

For probabilistic parameters, we set the supplier availability probability $\beta_1 \in [0, 1]$ with $\beta_1 \sim U(0.8, 1)$, the backlog probability $\beta_2 \in [0, 1]$ with $\beta_2 \sim U(0, 1)$, and the risk sensitivity parameter $\alpha \in [0, 1]$ with $\alpha \sim U(0, 1)$. For the demand distribution vector $v_D \in [0, 1]^{2M+1}$, we sample it uniformly from a $2M$-simplex. For the criterion parameter α, we set $Q_p = [0.01, 0.99]$. After

[1] A uniform distribution with a support $[6, 10]$.

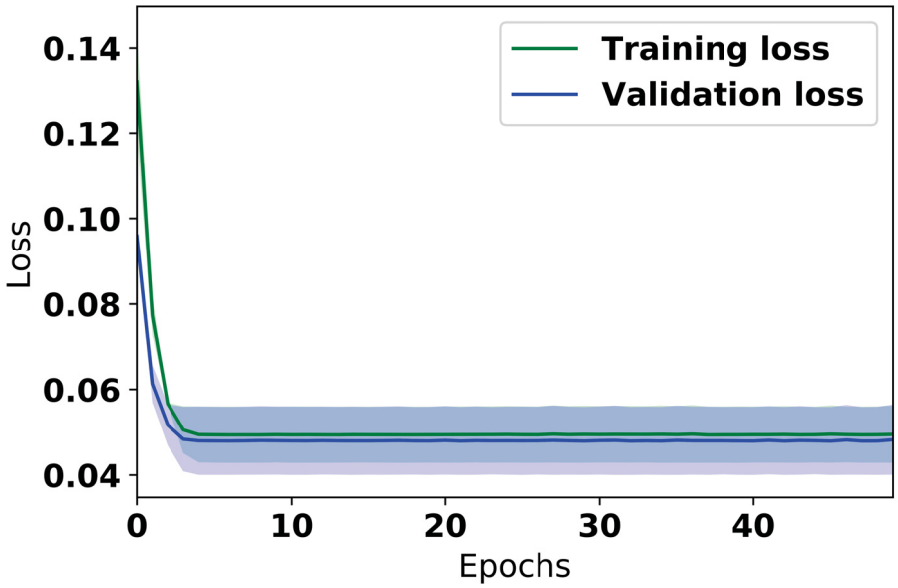

Figure 4.3: The loss for training/validating a 3-layer network in 50 epochs.

defining the domain features, labels are calculated, which include the optimal risk and policy.

We denote a policy list (vector) by L_p, in which each item represents an allowable policy. An NN will be trained as an approximator of the function, whose inputs represent the inventory parameters and α, and outputs represent ρ_α and the corresponding policy (denoted by a Boolean vector or an index), i.e., the domain set $\mathcal{K} = P_r \times P_s \times C_1 \times C_2 \times C_b \times C_f \times C_h \times [0.8, 1] \times [0, 1]^{2M+3} \times Q_p$, and the label set $\mathcal{L} = \mathbb{R} \times \{0, 1\}^{|L_p|}$ in general. Next, we train an NN with the synthetic dataset and show the validity of the proposed scheme.

4.3.4 Numerical Result

Since VaR is numeric and (deterministic) inventory policy is categorical, we consider both of them to be numeric. Considering the setting given in Section 4.3.3, we set a network $\hat{\nu}^g_{w,16,13} : \mathcal{K} \to \mathbb{R}^{13}$. Now we use Keras to construct, train and validate an NN for this small inventory problem. Firstly, we try a 3-layer NN. We set $Q = 3$, the numbers of nodes $N_0 = 16, N_1 = 12, N_2 = 8, N_3 = 13$. We set the *relu* function and the linear function as the activation functions for the hidden and output layers, respectively. Set the *mean squared error* as the loss function, and *adam* as the optimizer. Given that all hyperparameters are determined, the network is trained with a training

dataset $G_t \subset G$. We train the NN 50 times for the means and variances of the loss values at different epochs. We set the batch size to be 50 for the batch gradient descent.

Since it is a regression problem, we measure the loss by the regression metrics, such as mean absolute error or mean squared error (MSE). In Figure 4.3, it shows that the result with the training data size 10^5 converges within 10 epochs, with a hit rate around 95% in both training and validation phases. The error regions represent the standard deviations of means along the epoch axis.

4.4 Related Work

Constrained probabilistic MDPs take VaR as a constraint. The mean-VaR portfolio optimization problem was solved with the Lagrange multiplier for the VaR constraint over a continuous time span (Yiu, Wang, & Mak, 2004). Bonami and Lejeune (2009) solved the mean-variance portfolio optimization problem, and used variants of Chebychev's inequality to derive convex approximations of the VaR function. Randour et al. (2015) converted the total discounted reward to an almost-sure quantile, and proposed an algorithm based on linear programming to solve the weighted multi-constraint quantile problem. It is also pointed out that randomized policy is necessary when VaR is considered in the constraint, and an example can be found in (Defourny et al., 2008).

The SDM problems considering risk, dynamic environment, and constraint are usually studied separately. Besides the works reviewed in Section 3.3, Shen (2015) generalized risks to the valuation functions. The author applied a set of valuation functions, derived some model-free risk-sensitive reinforcement learning algorithms, and presented a risk control example in simulated algorithmic trading of stocks. For SDMs in dynamic environments, Hadoux (2015) proposed a new model named Hidden Semi-Markov-Mode Markov Decision Process (HS3MDP), which represented non-stationary problems whose dynamics evolved among a finite set of contexts. The author adapted the Partially Observable Monte-Carlo Planning (POMCP) algorithm to HS3MDPs in order to efficiently solve those problems. The POMCP algorithm used a black-box environment simulator and a particle filter to approximate a belief state. The simulator relaxed the model-based requirement, and each filter particle represented a state of the POMDP being solved. For different types of dynamic environment, the author compared a regret-based method with its Markov counterpart (S. P.-M. Choi,

Zhang, & Yeung, 2001). In the regret-based method, the agent was involved in a two-players repeated game, where two agents (the player and the opponent, which can be the environment) chose an action to play, got a feedback, and repeated the game.

For SDMs with constraint(s), Chow (2017) investigated the variability of stochastic rewards and the robustness to modeling errors. The author analyzed a unifying planning framework with coherent risks (such as CVaR), which was robust to inherent uncertainties and modeling errors, and output a time-consistent policy. A scalable approximate value-iteration algorithm on an augmented state space was developed for large scale problems with a data-driven setup. The author proposed novel policy gradient and actor-critic algorithms for CVaR-constrained and chance-constrained optimization in MDPs. Furthermore, the author proposed a framework for risk-averse model predictive control, where the risk was time-consistent and Markovian. In the framework of CMDP (Altman, 1999), Chow, Nachum, Duenez-Guzman, and Ghavamzadeh (2018) derived two LP-based algorithms—safe policy iteration and safe value iteration—for problems with constraints on expected cumulative costs. The algorithms hinged on a novel Lyapunov method, which constructed Lyapunov functions to provide an effective way to guarantee the global safety of a policy during training via a set of local, linear constraints. For unknown environment models and large state/action spaces, the authors proposed two scalable safe RL algorithms: safe DQN, an off-policy fitted Qiteration method, and safe DPI, an approximate policy iteration method. Another way to consider risk in an SDM problem is to avoid some dangerous system states. Geibel and Wysotzki (2005) defined the risk with respect to a policy as the probability of entering such states, and set the constraint as the probability being smaller than a predefined threshold. The authors presented a model-free RL algorithm with a deterministic policy space. The algorithm was based on weighting the original value function and the risk. The weight parameter was adapted to find a feasible solution for the constrained problem, and the probability defining the risk was expressed as the expectation of a cumulative return.

4.5 Conclusions and Future Research

We propose a scheme using functional approximator and RL methods to solve SDM problems with risk-sensitive constraints in a dynamic scenario. We consider risks as functions of mean and

variance of the return in an induced Markov process. As shown in Section 1.2, most, if not all, law-invariant risks can be evaluated or estimated with return variance. Considering that rewards in practical problems are often stochastic, we implement SAT to enable the variance formula and preserve the reward sequence. In an inventory control problem with practical factors, an NN is trained and validated with a synthetic training dataset in the numerical experiment.

The current work can be extended in two ways. The first one has to do with the "no free lunch" theorem. In our inventory control model, the more factors considered, the larger size the policy space has. When the warehouse capacity is 3, there are 8748 deterministic policies, when it is 4 and 5, the size goes to around 2.4×10^7 and 1.4×10^{12}, respectively. How to deal with a policy space of such an astronomical magnitude in a risk-sensitive case is still an unsettled problem. The second one is how to deal with two different output data types in an NN efficiently. In our setting, the outputs include optimal risk value (numeric) and policy (categorical). In this case, a crucial task is to properly set activation functions, normalization, and loss function for network training.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we proposed the SAT for risk estimation in MDPs and Markov processes. Three law-invariant risks have been estimated with the aid of the return variance formula. In the framework of MDP, we proposed a scheme for constrained risk-sensitive SDMs in dynamic environments.

In Chapter 1, we gave notations of MDPs with four reward forms. In the section of risks, we categorized common-used risks in RL, and claimed that most, if not all, inherent risks depend on the reward sequence $(R_t : t \in \{1, \cdots, N\})$. With respect to risk-sensitive criteria, we pointed out the model difference between theoretical methods and practical problems with respect to forms of reward, which is implied by Question 1.1 and 1.2. The problem is that, on the one hand, theoretical methods often work for MDPs or Markov processes with simple forms of reward. On the other hand, practical problems often occupy complicated forms of reward. In those cases, how to solve the practical problems with the theoretical methods for processes with simple rewards only? A naive way to work around this problem is to simplify the reward. We described three model simplifications, which work perfectly when the criteria refer to the total reward expectation, but fail in risk-sensitive cases, since they change (R_t). To illustrate how the proposed SAT preserves (R_t), we considered the return variance formula as a representative of a family of theoretical methods, and focused on the three law-invariant risks, i.e., exponential utility risk, mean-variance risk, and VaR, all of which can be estimated with the return variance.

We proposed the SAT in Chapter 2 for four different cases. The goal of the SAT is to transform processes with complicated rewards to ones with simple rewards, and preserve (R_t) at the same time. The essence of the SAT is to construct a bijective mapping from an "augmented" state space to a possible "situation" space, in which each situation determines an immediate reward. Furthermore, there exists a surjective mapping from the situation space to the reward space as well. Therefore, we now have a chance to analyze the process of immediate rewards directly as we analyze the process of state. However, nothing is free. The transformed processes have disadvantages comparing with the original ones. One problem is that the new state space is augmented. In Case 4, the size of the new state space is $(|S^2 \times A \times J| + |S|)$. Comparing with the original size $|S|$, a lot more computational power is needed. To relief this problem to some degree, we proposed the state lumping theorem based on the definition of isotopic states. Considering the special structural property of the transformed transition probability, transformed processes have a fair chance to decrease the size of the state space.

In Chapter 3, we estimated the three risks. The goal of the part is three-fold. First, we illustrated how the model simplification changes return (or total reward) distribution, as well as (R_t). The essence of the model simplification is to preserve the expectation of return (or total reward) by preserving conditional expectations of immediate rewards only. Therefore, it changes (R_t) and the return distribution. In the examples, we consider the return variance as a measure to show the change, since return variance is crucial in risk estimation. Second, we illustrated how the proposed SAT works in different cases. The first three cases are for policy evaluation, which plays an important part in RL. The most general case for an MDP as a whole, which could be deployed in model-free methods, such as Q-learning. Third, we showed that the three law-invariant risks can be estimated with the return variance. As the second central moment, the variance has a decisive effect on the risks. As shown in Section 1.2, many different risks are functions of return variance. Therefore, in our study, we focused on the return variance calculation, and reckon it as a representative of the family of theoretical methods to show the power of the proposed SAT.

After such a large amount of theoretical work, we dealt with a more realistic problem in Chapter 4. In SDM problems, we consider three factors at the same time: risk, constraint, and dynamic environment. Usually, these concerns are studied separately in SDM, as reviewed in Section 4.4.

Now by virtue of the proposed SAT, we can estimate risks in MDPs of practical problems and generate a synthetic dataset by ourselves. The reason why we need a synthetic dataset instead using a historical dataset has been explained in Section 4.2. In the proposed scheme, for any vector of environmental features, the problem is modeled for risk estimation (such as the approach we proposed in Chapter 3). By enumerating all deterministic policies, we can find the optimal risk value and policy. By doing so, we can generate a synthetic dataset for training an approximator. In our study, we chose an NN as the approximator. A numerical experiment illustrated the validity of the proposed scheme.

5.2 Future Work

We finally conclude this thesis with two additional important future research directions with respect to the SAT.

5.2.1 SAT with Deep Q-Network

One future work is to implement the SAT in Deep Q-Network (DQN) (Mnih et al., 2015). The goal to combine the SAT with DQN is two-fold. First, we can deal with a large state space by using NN. Second, we can learn the real immediate rewards by combining Q-learning with the SAT, which may offer us a platform for model-free risk estimation methods. A solution from a traditional risk-sensitive Q-learning could be like that, for a given state, an optimal action generates a preferred immediate reward distribution, since some measure on the distribution is optimal. By comparison, a solution from a risk-sensitive Q-learning with the proposed SAT could be like that, for a given state, an optimal action generates a preferred immediate reward value. By so doing, we divide a finite return distribution into a set of possible reward values, which lowers the uncertainty from the output of an algorithm.

5.2.2 SAT with Distributional RL

The other future work is to combine the SAT with distributional RL. Two points are worth noting when we consider the SAT in distributional RL. First, most distributional RL methods, such

as (Bellemare et al., 2017), outputs the optimal policy based on expectations with respect to an estimated return distribution conditioned on states. By virtue of the proposed SAT, a finite return distribution could be separated by considering each possible outcome as a state. By combining the SAT with distributional RL, we might be able to update the policy from a real risk-sensitive perspective, instead of considering the traditional expectation value in Bellman operator. Second, for the SAT to solve problems with continuous space, we may borrow the idea of quantile actors (Zhang & Yao, 2019). Each quantile actor is responsible for proposing an action that maximizes one specific quantile of a state-action value distribution.

Appendix A

A.1 Proof of Theorem 2.1

Proof. The proof has two steps. Step 1 constructs a second MDP and shows that, for every possible sample path in the first MDP, there exists a corresponding sample path in the second MDP. Step 2 proves that, the probability of any possible sample path in first MDP equals to the probability of its counterpart in the second MDP.

Step 1: Define $S = \{1, \cdots, |S|\}$. Define $J \coloneqq \bigcup_{x,y \in S, a \in A_x} \text{supp}(d_S(\cdot \mid x, a))$. Define $S^\ddagger = S^2 \times A \times J$. In order to remove the dependency of the initial state distribution on policy, define a null state space $S_n = \{s_{n,1}, \cdots, s_{n,|S|}\}$, and $s_n \cap S^\ddagger = \emptyset$. Define the state space $S^\dagger = S^\ddagger \cup s_n$.

For all $x^\dagger = (x, a, y, j), y^\dagger = (y, a', y', j') \in S^\dagger$, where $y' \in S, a' \in A_y, j, j' \in J$, define the state-based reward function $r^\dagger(j \mid x^\dagger, \cdot) = 1$, and $r^\dagger(0 \mid s_{n,\cdot}, \cdot) = 1$; define the transition kernel $p^\dagger(y^\dagger \mid x^\dagger, a') = p(y' \mid y, a')r(j' \mid y, a', y')$, and $p^\dagger(x^\dagger \mid s_{n,x}, a) = p(y \mid x, a)r(j \mid x, a, y)$; the initial state distribution $\mu^\dagger(s_{n,x}) = \mu(x)$.

Now we have two MDPs. Let $MDP_1 = \langle S, A, r, p, \mu \rangle$, and $MDP_2 = \langle S^\dagger, A, r^\dagger, p^\dagger, \mu^\dagger \rangle$. For any sample path $(x_1, a_1, j_1, x_2, a_2, j_2, x_3, a_3, j_3, x_4 \cdots)$ in MDP_1, there exists a sample path

$$(w_{x_1}, a_1, 0, (x_1, a_1, j_1, x_2), a_2, j_1, (x_2, a_2, j_2, x_3), a_3,$$
$$j_2, (x_3, a_3, j_3, x_4), \cdots)$$

in MDP_2. Therefore, we proved that for every possible sample path for the first MDP, there exists a corresponding sample path for the second MDP.

Step 2: Next we prove the probabilities for the two sample paths are equal. Here we prove it by mathematical induction. Set time epoch to n after the first x_{n+1} in MDP_1, and after the first (x_n, a_n, j_n, x_{n+1}) in MDP_2.

Denote the partial sample path till epoch i by ι_i in MDP_1. Given any $\pi \in \Pi_r$ for MDP_1, the probability for the sample path before epoch 1 in MDP_1 is

$$\mathbb{P}(\iota_1 = (x_1, a_1, j_1, x_2)) =$$
$$\mu(x_1)\pi(a_1 \mid x_1)p(x_2 \mid x_1, a_1)r(j_1 \mid x_1, a_1, x_2).$$

There exists a policy $\pi^\dagger$ for MDP_2, with $\pi^\dagger(a \mid s_{n,x}) = \pi(a \mid x)$, and $\pi^\dagger(a \mid x^\dagger) = \pi(a \mid y)$. The probability for the sample path before epoch 1 in MDP_2 is

$$\begin{aligned}
&\mathbb{P}(\iota_1^\dagger = (w_{x_1}, a_1, 0, (x_1, a_1, j_1, x_2))) \\
&= \mu(s_{n,x_1})\pi^\dagger(a_1 \mid s_{n,x_1})p((x_1, a_1, j_1, x_2) \mid s_{n,x_1}, a_1) \\
&= \mathbb{P}(\iota_1 = (x_1, a_1, j_1, x_2)).
\end{aligned}$$

Therefore, at epoch 1, the two partial sample paths share the same probability.

Assuming that the two partial sample paths share the same probability at epoch n, then the probability for the sample path before epoch $n+1$ in MDP_1 is

$$\begin{aligned}
&\mathbb{P}(\iota_{n+1} = (\iota_n, x_{n+1}, a_{n+1}, j_{n+1}, x_{n+2})) \\
&= \mathbb{P}(\iota_n = (x_1, \cdots, x_n, a_n, j_n, x_{n+1}))\pi(a_{n+1} \mid x_{n+1})\times \\
&p(x_{n+2} \mid x_{n+1}, a_{n+1})r(j_{n+1} \mid x_{n+1}, a_{n+1}, x_{n+2}).
\end{aligned}$$

The probability for the sample path before epoch $n+1$ in MDP_2 is

$$\begin{aligned}\mathbb{P}(\iota^{\dagger}_{n+1} = (\iota^{\dagger}_{n}, a_{n+1}, j_n, (x_{n+1}, a_{n+1}, j_{n+1}, x_{n+2}))) = \\ \mathbb{P}(\iota^{\dagger}_{n} = (s_{n,x_1}, \cdots, (x_n, a_n, j_n, x_{n+1})))\times \\ \pi^{\dagger}(a_{n+1} \mid (x_n, a_n, j_n, x_{n+1}))\times \\ p((x_{n+1}, a_{n+1}, j_{n+1}, x_{n+2}) \mid (x_n, a_n, j_n, x_{n+1}), a_{n+1})\times \\ r(j_n \mid (x_n, a_n, j_n, x_{n+1}), a_{n+1}) \\ = \mathbb{P}(\iota_{n+1} = (\iota_n, x_{n+1}, a_{n+1}, j_{n+1}, x_{n+2})).\end{aligned}$$

By induction we proved that, the probability of any possible sample path in $\langle S, A, r, p, \mu\rangle$ equals to the probability of its counterpart in $\langle S^{\dagger}, A, r^{\dagger}, p^{\dagger}, \mu^{\dagger}\rangle$. As a subsequence of a sample path, the reward sequence $\{R_t\}$ is preserved with a null reward at $t = 1$. Given the discount factor γ, we can compensate this time drift effect simply by setting $r^{\dagger}(x^{\dagger}) = r^{\dagger}(x^{\dagger})/\gamma$ to preserve the return distribution. Theorem 2.1 is proved.

□

A.2 Proof of Theorem 2.2

Proof. Denote the reward sequence for $\mathcal{M}'$ by $(R'_t : t \in \{1, \cdots, N\})$. Since for $x \in S' \setminus \{x_i\}$, $\mu'(x) = \mu(x)$, $\mu'(x_i) = \mu(x_i) + \mu(x_j)$, we have R_1 and R'_1 share the same distribution. Since the outcome x_j is replaced by x_i in $\mathcal{M}'$, the two events $X_t \in \{x_i, x_j\}$ and $X'_t = x_i$ share the same probability conditioned on X_{t-1}; and since $p^{\pi}(y \mid x_i) = p^{\pi}(y \mid x_j)$ for $y \in S \setminus \{x_i, x_j\}$, the replacement does not change the probability of $X'_{t+1} = x \in S'$ conditioned on X_t. Then, for $t = 2, \cdots, N$, R_t and R'_t share the same probability conditioned on X_{t-1}. Therefore, the two Markov process $\mathcal{M}$ and $\mathcal{M}'$ share the same $(R_t : t \in \{1, \cdots, N\})$. □

A.3 Assumptions for Inventory Control Model Formulation

At each epoch (every first day of a month, for example), the warehouse manager needs to determine how many units to order based on the current inventory and the demand prediction. Based on the information, he (or she) decides whether to order from a supplier. He is faced with a tradeoff between the cost of keeping inventory and the lost sales. The goal is to maximize profit over a decision-making horizon. Here we suppose that the demand follows a known probability distribution.

To simplify the problem, some assumptions are made as follows:

(1) The order decision is made at the beginning of each epoch and delivery occurs instantaneously.

(2) All demand orders are filled at the end of an epoch.

(3) If demand exceeds inventory, the customer will go elsewhere to buy the product, which means no backlogging of unfilled orders, so that excess demand is lost.

(4) The revenues, costs, and demand distribution do not vary from epoch to epoch.

(5) The warehouse has a capacity of M units.

A.4 Proof of Lemma 3.1

Proof. Given an MDP $\langle N, S, A, r, p, \mu, v\rangle$, define C as the cumulative reward space, m_a and M_a as the minimum and maximum of the rewards for action a. Then C can be $\{0\} \bigcup_{t=1}^{N} \bigcup_{a\in A}[t \cdot m_a, t \cdot M_a]$, or we can acquire C by enumerating all possibile cumulative reward within the short horizon. Define the augmented state space $S' = S \times C$ for the new MDP. For all $x, y \in S$ and $x', y' \in S'$, define the action space $A'_{x'} = A_x$ where $x' = (x, \cdot)$; define the transition kernel $p'(y' \mid x', a) = p(y \mid x, a)$ where $y' = (y, c_i), x' = (x, c_j)$, $c_i - c_j = r(x, a, y)$, $c_i, c_j \in C$ and $a \in A_x$; define the initial distribution $\mu'((x, 0)) = \mu(x)$. Set the salvage reward $v' = \mathbb{1}_{[\Phi\geq\tau-v]}$, where τ is the threshold and Φ is the cumulative reward at the final epoch. Now we have an augmented-state 0-1 MDP $\langle N, S', A, v', p', \mu'\rangle$.

Given an augmented-state 0-1 MDP $\langle N, S', A, v', p', \mu' \rangle$, for all $x' \in S'$, implement the backward induction as follows.

Step 1: Set $t = N$ and

$$u_N^*(x') = r'_N(x') = \mathbb{1}_{[\Phi \geq \tau - v]}.$$

Step 2: Set $t = t - 1$, and compute $u_t^*(x')$ by

$$u_t^*(x') = \max_{a \in A_{x'}} \{r'(x', a) + \sum_{y' \in S'} p'(y' \mid x', a) u_{t+1}^*(y')\},$$

where $r'(x', a) = 0$, therefore,

$$u_t^*(x') = \max_{a \in A_{x'}} \{ \sum_{y' \in S'} p'(y' \mid x', a) u_{t+1}^*(y')\}.$$

Since the only rewards are $r'_N = \mathbb{1}_{[\Phi \geq \tau - v]}$, we have $u_t(x') = P(\Phi \geq \tau \mid X'_t = x')$, i.e., the probability that the total reward $\Phi \geq \tau$ given any state at any epoch.

Step 3: If $t = 1$, stop. Otherwise return to Step 2. The optimal policy derived from

$$A^*_{x',t} = \underset{a \in A_{x'}}{\operatorname{argmax}} \{P(\Phi \geq \tau \mid X'_t = x')\}$$

gives the highest probability to reach the threshold. □

A.5 Augmented-State 0-1 MDP Algorithm

For policy conversion, see Section 5.2 in Xu and Mannor (2011).

Algorithm 6 Augmented-State 0-1 MDP

Input: a finite-horizon MDP $\langle N, S, A, r, p, \mu, v \rangle$ and a threshold $\tau \in \mathbb{R}$.
Output: a deterministic policy π^* and the optimal VaR η_τ.

1: Generate the cumulative reward set

$$C = \{0\} \bigcup_{t=1}^{N} \bigcup_{a \in A} \cdot [t \cdot m_a, t \cdot M_a],$$

or enumerating all possibilities;
2: Generate the augmented state space $S' = S \times C$;
3: Generate the salvage reward $v' = \mathbb{1}_{[\Phi \geq \tau - v]}$;
4: **for all** $x^\dagger = (x, y, j) \in S^\dagger$ **do**
5: Construct the transition kernel

$$p'(y' \mid x', a) = p(y \mid x, a);$$

6: **end for**
7: Calculate $\mu_0'(x') = \mu(x)\mathbb{1}_{[x'=(x,0)]}$;
8: Solve the MDP $\langle N, S', A, v', p', \mu_0' \rangle$ with the expected total reward objective, and output the policy.

References

Ahiska, S. S., Appaji, S. R., King, R. E., & Warsing Jr, D. P. (2013). A Markov decision process-based policy characterization approach for a stochastic inventory control problem with unreliable sourcing. *International Journal of Production Economics*, *144*(2), 485–496.

Altman, E. (1999). *Constrained Markov decision processes*. CRC Press, ISBN: 9780849303821.

Artzner, P., Delbaen, F., Eber, J., & Heath, D. (1998). Coherent measures of risk. *Mathematical Finance*, *9*(3), 1–24.

Bellemare, M., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)* (pp. 449–458).

Berkenkamp, F., Turchetta, M., Schoellig, A., & Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. In *Proceedings of the 31st Advances in Neural Information Processing Systems (NeurIPS)* (pp. 908–918).

Boda, K., & Filar, J. A. (2006). Time consistent dynamic risk measures. *Mathematical Methods of Operations Research*, *63*(1), 169–186.

Bonami, P., & Lejeune, M. A. (2009). An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations Research*, *57*(3), 650–670.

Borkar, V. S. (2002). Q-learning for risk-sensitive control. *Mathematics of Operations Research*, *27*(2), 294–311.

Borkar, V. S., & Meyn, S. P. (2002). Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, *27*(1), 192–209.

Bouakiz, M., & Kebir, Y. (1995). Target-level criterion in Markov decision processes. *Journal of*

Optimization Theory and Applications, *86*(1), 1–15.

Burke, C., & Rosenblatt, M. (1958). A Markovian function of a Markov chain. *The Annals of Mathematical Statistics*, *29*(4), 1112–1122.

Chatterjee, K. (2007). Markov decision processes with multiple long-run average objectives. In *Proceedings of the International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)* (pp. 473–484).

Chiu, C.-H., & Choi, T.-M. (2016). Supply chain risk analysis with mean-variance models: A technical review. *Annals of Operations Research*, *240*(2), 489–507.

Choi, S. P.-M., Zhang, N. L., & Yeung, D.-Y. (2001). Solving hidden-mode Markov decision problems. In *Proceedings of the 8th international workshop on artificial intelligence and statistics (AISTATS)*.

Choi, T.-M., Li, D., & Yan, H. (2008). Mean–variance analysis for the newsvendor problem. *IEEE Transactions on Systems, Man, and Cybernetics (SMC)-Part A: Systems and Humans*, *38*(5), 1169–1180.

Chow, Y. (2017). *Risk-sensitive and data-driven sequential decision making* (PhD dissertation). Stanford University.

Chow, Y., Ghavamzadeh, M., Janson, L., & Pavone, M. (2017). Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, *18*(1), 6070–6120.

Chow, Y., Nachum, O., Duenez-Guzman, E., & Ghavamzadeh, M. (2018). A Lyapunov-based approach to safe reinforcement learning. In *Proceedings of the 32th neural information processing systems (NeurIPS)* (pp. 8092–8101).

Chung, K.-J., & Sobel, M. J. (1987). Discounted MDPs: Distribution functions and exponential utility maximization. *SIAM journal on Control and Optimization*, *25*(1), 49–62.

Cohen, L., & Young, A. (2006). *Multisourcing: Moving beyond outsourcing to achieve growth and agility*. Harvard Business Press, ISBN: 9781591397977.

Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit quantile networks for distributional reinforcement learning. *arXiv:1806.06923*.

Dai, P., Weld, D. S., & Goldsmith, J. (2011). Topological value iteration algorithms. *Journal of*

Artificial Intelligence Research, *42*, 181–209.

Defourny, B., Ernst, D., & Wehenkel, L. (2008). Risk-aware decision making and dynamic programming. In *Proceedings of the 22nd neural information processing systems (NeurIPS) workshop on model uncertainty and risk in reinforcement learning* (pp. 1–8).

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

Durbin, J. (1973). *Distribution theory for tests based on the sample distribution function*. SIAM, ISBN: 9780898710076.

Ermon, S., Gomes, C., & Vladimirsky, A. (2012). Probabilistic planning with non-linear utility functions and worst-case guarantees. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (pp. 4–8).

Filar, J. A., Krass, D., Ross, K. W., & Member, S. (1995). Percentile performance criteria for limiting average Markov decision processes. *IEEE Transactions on Automatic Control*, *40*(I), 2–10.

García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*(1), 1437–1480.

Geibel, P., & Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, *24*, 81–108.

Gerber, H. U. (1971). The discounted central limit theorem and its Berry-Esséen analogue. *The Annals of Mathematical Statistics*, *42*(1), 389–392.

Gilbert, H., & Weng, P. (2016). Quantile reinforcement learning. *arXiv:1611.00862*.

Guo, X., Ye, L., & Yin, G. (2012). A mean-variance optimization problem for discounted Markov decision processes. *European Journal of Operational Research*, *220*(2), 423–429.

Hadoux, E. (2015). *Markovian sequential decision-making in non-stationary environments: application to argumentative debates* (PhD dissertation). UPMC, Sorbonne Universites CNRS.

Harrison, P. G., & Patel, N. M. (1992). *Performance modelling of communication networks and computer architectures*. Addison-Wesley Longman Publishing Co., Inc., ISBN: 9780201544190.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, *4*(2), 251–257.

Hou, P., Yeoh, W., & Varakantham, P. (2014). Revisiting risk-sensitive MDPs: New algorithms and results. In *Proceedings of the 24th international conference on automated planning and scheduling (ICAPS)* (pp. 136–144).

Howard, R. A. (1964). *Dynamic programming and Markov processes*. Wiley for The Massachusetts Institute of Technology, ISBN: 9780262080095.

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

Huang, W., & Haskell, W. B. (2017). Risk-aware Q-learning for Markov decision processes. In *Proceedings of the 56th IEEE conference on decision and control (CDC)* (pp. 4928–4933).

Jones, G. L. (2004). On the Markov chain central limit theorem. *Probability Surveys*, *1*, 5–1.

Junges, S., Jansen, N., Dehnert, C., Topcu, U., & Katoen, J.-P. (2016). Safety-constrained reinforcement learning for MDPs. In *Proceedings of the 22nd international conference on tools and algorithms for the construction and analysis of systems (TACAS)* (pp. 130–146).

Kemeny, J. G., & Snell, J. L. (1976). *Finite Markov chains*. Springer-Verlag, New York, ISBN: 9780387901923.

Kira, A., Ueno, T., & Fujita, T. (2012). Threshold probability of non-terminal type in finite horizon Markov decision processes. *Journal of Mathematical Analysis and Applications*, *386*(1), 461–472.

Kolobov, A., Mausam, Weld, D. S., & Geffner, H. (2011). Heuristic search for generalized stochastic shortest path MDPs. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS)* (pp. 130–137).

Kontoyiannis, I., & Meyn, S. P. (2003). Spectral theory and limit theorems for geometrically ergodic Markov processes. *Annals of Applied Probability*, *13*, 304–362.

Kusuoka, S. (2001). On law invariant coherent risk measures. In *Advances in mathematical economics* (Vol. 3, pp. 83–95). Springer, ISBN: 9784431659372.

Lau, H.-S. (1980). The newsboy problem under alternative optimization objectives. *Journal of the Operational Research Society*, *31*(6), 525–535.

Ma, S., & Yu, J. Y. (2017). Transition-based versus state-based reward functions for MDPs with value-at-risk. In *Proceedings of the 55th annual Allerton conference on communication, control, and computing (Allerton)* (pp. 974–981).

Ma, S., & Yu, J. Y. (2019a). State-augmentation transformations for risk-sensitive reinforcement learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)* (Vol. 33, pp. 4512–4519).

Ma, S., & Yu, J. Y. (2019b). Variance-based risk estimations in Markov processes via transformation with state lumping. In *Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC), preprint arxiv:1907.04269.*

Mannor, S., & Tsitsiklis, J. (2011). Mean-variance optimization in Markov decision processes. In *Proceedings of the 28th international conference on machine learning (ICML)* (pp. 1–22).

Meyn, S. P., & Tweedie, R. L. (2009). *Markov chains and stochastic stability*. Springer Science & Business Media, ISBN: 9780521731829.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mohebbi, E. (2004). A replenishment model for the supply-uncertainty problem. *International Journal of Production Economics*, *87*, 25–37.

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

Ohtsubo, Y., & Toyonaga, K. (2002). Optimal policy for minimizing risk models in Markov decision processes. *Journal of Mathematical Analysis and Applications*, *271(1)*, 66–81.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley, ISBN: 9780471727828.

Randour, M., Raskin, J., & Sankur, O. (2015). Percentile queries in multi-dimensional Markov decision processes. *Computer Aided Verification*, *9206*, 123–139.

Ravindran, B., & Barto, A. G. (2002). Model minimization in hierarchical reinforcement learning. In *Proceedings of the International Symposium on Abstraction, Reformulation, and Approximation (SARA)* (pp. 196–211).

Riedel, F. (2004). Dynamic coherent risk measures. *Stochastic Processes and their Applications*, *112*(2), 185–200.

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, *48*, 67–113.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Pearson Education Limited, ISBN: 9780136042594.

Ruszczyński, A. (2010). Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, *125*(2), 235–261.

Scheffé, H. (1999). *The analysis of variance*. John Wiley & Sons, ISBN: 9780471345053.

Shen, Y. (2015). *Risk sensitive Markov decision processes* (PhD dissertation). Technischen Universität Berlin.

Shen, Y., Tobia, M. J., Sommer, T., & Obermayer, K. (2014). Risk-sensitive reinforcement learning. *Neural Computation*, *26*(7), 1298–1328.

Sobel, M. J. (1982). The variance of discounted Markov decision processes. *Journal of Applied Probability*, *19*(4), 794–802.

Sobel, M. J. (1994). Mean-variance tradeoffs in an undiscounted MDP. *Operations Research*, *42*(1), 175–183.

Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory.*

Steinmetz, M., Hoffmann, J., & Buffet, O. (2016). Goal probability analysis in mdp probabilistic planning: Exploring and enhancing the state of the art. *Journal of Artificial Intelligence Research*, *57*, 229–271.

Tamar, A., Chow, Y., Ghavamzadeh, M., & Mannor, S. (2015). Policy gradient for coherent risk measures. In *Proceedings of the 29th Advances in Neural Information Processing Systems (NeurIPS)* (pp. 1468–1476).

Taniguchi, E., Thompson, R. G., & Yamada, T. (2012). Emerging techniques for enhancing the practical application of city logistics models. *Procedia-Social and Behavioral Sciences*, *39*, 3–18.

White, D. J. (1988). Mean , variance , and probabilistic criteria in finite Markov decision processes : A review. *Journal of Optimization Theory and Applications*, *56*(1), 1–29.

Woodroofe, M. (1992). A central limit theorem for functions of a Markov chain with applications to shifts. *Stochastic Processes and their Applications*, *41*(1), 33–44.

Wu, C., & Lin, Y. (1999). Minimizing risk models in Markov decision process with policies depending on target values. *Journal of Mathematical Analysis and Applications*, *23*(1), 47–67.

Xia, L. (2018). Mean-variance optimization of discrete time discounted Markov decision processes. *Automatica*, *88*, 76–82.

Xu, H., & Mannor, S. (2011). Probabilistic goal Markov decision processes. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2046–2052).

Yiu, K. F. C., Wang, S. Y., & Mak, K. L. (2004). Optimal portfolios under a value-at-risk constraint. *Journal of Economic Dynamics and Control*, *28*(7), 1317–1334.

Yu, P., Haskell, W. B., & Xu, H. (2018). Approximate value iteration for risk-aware Markov decision processes. *IEEE Transactions on Automatic Control*, *63*(9), 3135–3142.

Yu, S. X., Lin, Y., & Yan, P. (1998). Optimization models for the first arrival target distribution function in discrete time. *Journal of Mathematical Analysis and Applications*, *225*(1), 193–223.

Zhang, S., & Yao, H. (2019). QUOTA: The quantile option architecture for reinforcement learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)* (Vol. 33, pp. 5797–5804).

Zheng, H. (2009). Efficient frontier of utility and CVaR. *Mathematical Methods of Operations Research*, *70*(1), 129–148.