Applications of Stochastic Control Theory for Portfolio Optimization

Wei Ning

Bachelor of Science and Bachelor of Education, Monash University, 2015
Master of Financial Mathematics, Monash University, 2017

A thesis submitted for the degree of

Doctor of Philosophy

at Monash University in 2021

School of Mathematics
Monash University
Melbourne, Australia
2021

Copyright notice

Abstract

The present work is devoted to investigating portfolio optimization from different perspectives. We consider continuous-time investment on a finite time horizon. In the first part, we address the problem of robust utility maximization, where we take into account the uncertainty in the drift and covariance matrix of the securities. Our model's novelty is that, instead of restricting the uncertain parameters in a fixed range, we design a convex penalty function to avoid unrealistic values. We connect this robust optimization problem with a two-player zero-sum stochastic differential game, and analyze it using stochastic optimal control techniques. In the second part, we provide a new objective function for portfolio optimization. Instead of considering the first and second moments of the return, our objective function includes the whole target distribution of the terminal wealth. We study this problem with the tools of optimal mass transport. In the numerical examples, we can successfully reach the attainable targets by controlling the portfolio allocation at each instant. We also consider consumption and cash input during the investment process in order to reach distributions that are either unattainable or sub-optimal. Our work also dedicates to proposing various numerical algorithms. In particular, we used a finite difference method, Generative Adversarial Networks, and Monte Carlo simulations to solve the robust portfolio allocation problem. They all provide accurate estimations in various numerical examples. We then designed two deep neural network-based algorithms to solve optimal transport problems. The first deep learning algorithm is based on a relaxation/penalization of the terminal constraint; in the second algorithm, we express the dual formulation of the optimal transport problem as a saddle point problem and solve it with adversarial networks. The second method is mesh-free; therefore, it is applicable to solve high-dimensional problems. The performance of this method is tested on several examples up to dimension 10.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Wei Ning
15 Feb 2021

Publications during enrolment

This thesis is comprised of three academic manuscripts that have been written during my candidature.

- Chapter 2 is based on joint work with Grégoire Loeper, Nicolas Langrené and Ivan Guo. The manuscript, titled ‘Robust utility maximization under model uncertainty via a penalization approach’, has been submitted to the journal *Mathematics and Financial Economics*. A version is currently available on arXiv under the same name.

- Chapter 3 is based on joint work with Grégoire Loeper, Nicolas Langrené and Ivan Guo. The manuscript, titled ‘Portfolio optimization with a prescribed terminal wealth distribution’, has been submitted to the journal *Quantitative Finance*. A version is currently available on arXiv under the same name.

- Chapter 4 is based on joint work with Grégoire Loeper, Nicolas Langrené and Ivan Guo. The manuscript, titled ‘Deep Semi-Martingale Optimal Transport’ is currently available on arXiv under the same name.

To all my family and friends

Acknowledgements

I would like to express my deep and sincere gratitude to my research supervisors, Prof. Grégoire Loeper, Dr. Ivan Guo, Dr. Nicolas Langrené, and Dr. Zili Zhu for giving me the opportunity to do research and providing me invaluable guidance throughout the research. It is a great honour and privilege to work and study under their guidance. I would like to thank them for their patience, empathy and wisdom.

I am extremely grateful to my parents for their love, caring and sacrifices for educating and preparing me for my future. I also want to express my thanks to my friends for their company and support. They make me feel less lonely during this challenging time in my life.

Contents

Chapter 1

Introduction

In this thesis, we investigate the problem of portfolio optimization from different perspectives. First we consider the problem of utility maximization under model uncertainty. Secondly, instead of maximizing a given utility function, we try to design an investment strategy that reaches a given final wealth distribution. All along this work, we will propose some efficient numerical algorithms to estimate the solutions, and the last part of the thesis is dedicated in particular to the use of neural networks.

This thesis is comprised of four chapters. In Chapter 1, we first provide the notation and related results to be used in the subsequent chapters (section 1.1 and section 1.2). Then we state the background of the research topic (section 1.3) and my contributions (section 1.4). Chapter 2 is based on our paper 'Robust utility maximization under model uncertainty via a penalization approach'. This work details utility maximization under uncertain drift and covariance matrix of the securities. Chapter 3 is based on our paper 'Portfolio optimization with a prescribed terminal wealth distribution'. This work details the problem of portfolio allocation with a prescribed terminal wealth density. Chapter 4 is based on our paper 'Deep Semi-Martingale Optimal Transport'. This work proposes two deep neural network-based algorithms to solve multi-dimensional optimal transport problems.

1.1 Notations

$\mathbb{R}^n$ — n-dimensional real Euclidean space.

$\mathbb{R}^{n\times m}$ — the set of all $(n \times m)$ real matrices.

$\mathbb{R}^+$ — non-negative real numbers.

$\mathbb{S}^d$ — the set of $d \times d$ symmetric positive semidefinite matrices.

$tr(A)$ — the trace of the square matrix A.

$x^\intercal$ — the transpose of the vector (or matrix) x.

$\langle \cdot, \cdot \rangle$ — the inner product

$:=$ — defined to be (see below).

$\mathcal{E} := [0, 1] \times \mathbb{R}^d$.

$\mathcal{D}$ — a Polish space equipped with its Borel σ-algebra.

$C(\mathcal{D}; \mathbb{R}^d)$ — the space of continuous functions on $\mathcal{D}$ with values in $\mathbb{R}^d$.

$C_b(\mathcal{D};\mathbb{R}^d)$ — the space of bounded continuous functions.

$C_b^*(\mathcal{D};\mathbb{R}^d)$ — the topological dual of C_b. When $\mathcal{D}$ is compact, $C_b^*(\mathcal{D};\mathbb{R}^d) = \mathcal{M}(\mathcal{D};\mathbb{R}^d)$. But when $\mathcal{D}$ is non-compact, $C_b^*(\mathcal{D};\mathbb{R}^d)$ is larger than $\mathcal{M}(\mathcal{D};\mathbb{R}^d)$.

$C_0(\mathcal{D};\mathbb{R}^d)$ — the space of continuous functions, vanishing at infinity.

$C_b^{1,2}(\mathcal{E})$ — the set of all functions $\phi : \mathcal{E} \to \mathbb{R}$ such that $\phi \in C_b(\mathcal{E})$ and $(\partial_t\phi, \partial_x\phi, \partial_{xx}\phi) \in C_0(\mathcal{E};\mathbb{R},\mathbb{R}^d,\mathbb{S}^d)$.

$\mathcal{P}(\mathcal{D})$ — the space of Borel probability measures on $\mathcal{D}$ with a finite second moment.

$\mathcal{M}(\mathcal{D};\mathbb{R})$ — the space of finite signed measures on $\mathcal{D}$ with values in $\mathbb{R}$.

$\mathcal{M}_+(\mathcal{D};\mathbb{R})$ — non-negative measures, the subset of $\mathcal{M}(\mathcal{D};\mathbb{R})$

$L^p(0,T;\mathbb{R}^n)$ — the set of Lebesgue measurable functions $\varphi : [0,T] \to \mathbb{R}^n$ such that $\int_0^T |\varphi(t)|^p dt \leq \infty \, (p \in [1,\infty])$.

$(\Omega, \mathcal{F}, \mathbb{P})$ — probability space.

$\{\mathcal{F}_t\}_{t\geq 0}$ — filtration

$(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ — filtered probability space.

$\mathbf{B}(U)$ — Borelian σ-field generated by the open subsets of the topological space U.

$X^{t,x}$ — the process X starting from x at time t.

$\mathbb{E}[X]$ — the expectation of the random variable X.

$\mathbb{E}[X \mid \mathcal{G}]$ — conditional expectation of X given $\mathcal{G}$.

$X \sim \mathcal{N}(\mu, \sigma^2)$ —the random variable X is normally distributed with mean μ and variance σ^2.

$\partial_x v$ or v_x — the first derivative of v with respect to x.

$\partial_{xx} v$ or v_{xx} — the second derivative of v with respect to x.

1.2 Elements of the general theory

Stochastic calculus serves as a fundamental tool throughout this thesis. In this chapter, we presents some concepts and results on stochastic calculus frequently used in the subsequent chapters. All the results are quoted from classical textbooks, such as Klebaner (2005), Karatzas and Shreve (1998) and Krylov (2008). As these are standard theorems from the literature, we only provide the sources of the proofs. We refer interested readers to the above books for a comprehensive treatment on these topics.

1.2.1 Stochastic processes

We consider continuous-time stochastic processes on a finite time interval in the thesis. We denote $\mathbb{T} = [0,T], 0 < T < \infty$. We fix a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{T}}, \mathbb{P})$

satisfying the usual conditions and a n-dimensional Brownian Motion $W = (W^1, ..., W^n)$ with respect to $\mathbb{F}$.

Definition 1.2.1. (Adapted process). A process $(X_t)_{t\in\mathbb{T}}$ is adapted (with respect to $\mathbb{F}$) if for all $t \in \mathbb{T}$, X_t is $\mathcal{F}_t$-measurable.

Definition 1.2.2. (Progressively measurable). A process $(X_t)_{t\in\mathbb{T}}$ is progressively measurable (with respect to $\mathbb{F}$) if for any $t \in \mathbb{T}$, the mapping $(s, \omega) \to X_s(\omega)$ is measurable on $[0, t] \times \Omega$ equipped with the product σ-field $\mathbf{B}([0, t]) \otimes \mathcal{F}_t$.

1.2.2 Stochastic differential equations (SDE)

We only introduce the concept of strong solutions of SDE. We are given functions $\mu(t, x, \omega) = (\mu_i(t, x, \omega))_{1\leq i\leq d}, \sigma(t, x, \omega) = (\sigma_{ij}(t, x, \omega))_{1\leq i\leq d, 1\leq j\leq n}$ defined on $\mathbb{T} \times \mathbb{R}^d \times \Omega$, and valued respectively in $\mathbb{R}^d$ and $\mathbb{R}^{d\times n}$. We assume that for all ω, the functions $\mu(\cdot, \cdot, \omega)$ and $\sigma(\cdot, \cdot, \omega)$ are Borelian on $\mathbb{T} \times \mathbb{R}^d$ and for all $x \in \mathbb{R}^d$, the processes $\mu(\cdot, x, \cdot)$ and $\sigma(\cdot, x, \cdot)$, written $\mu(\cdot, x)$ and $\sigma(\cdot, x)$ for simplification, are progressively measurable. We then consider the SDE valued in $\mathbb{R}^d$:

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t. \tag{1.1}$$

Theorem 1.2.1. (Existence and Uniqueness).

Existence and uniqueness of a strong solution to the SDE (1.1) is ensured if the Lipschitz and linear growth conditions are satisfied: there exists a constant K and a real-valued process κ such that for all $t \in \mathbb{T}$, $\omega \in \Omega$, $x, y \in \mathbb{R}^d$

(1)

$$|\mu(t, x, \omega) - \mu(t, y, \omega)| + |\sigma(t, x, \omega) - \sigma(t, y, \omega)| < K|x - y|, \tag{1.2}$$

(2)

$$|\mu(t, x, \omega)| + |\sigma(t, x, \omega)| \leq \kappa_t(\omega) + K|x|, \tag{1.3}$$

(3)

$$\mathbb{E}\left[\int_0^t |\kappa_u|^2 du\right] < \infty, \forall t \in \mathbb{T}. \tag{1.4}$$

Proof. See Pham (2009) or Klebaner (2005). □

Theorem 1.2.2. Let (1.2), (1.3) and (1.4) hold. Then for any ξ $\mathcal{F}_t$-measurable random variable valued in $\mathbb{R}^d$, such that $\mathbb{E}[|\xi|^p] < \infty$, for some $p \geq 1$, SDE (1.1) admits a unique strong solution starting from ξ at time $t \in \mathbb{T}$, i.e., $X_t = \xi$. For all $T > t$, there exists a constant C_T such that

$$\mathbb{E}\left[\sup_{t\leq s\leq T} |X_s|^p\right] \leq C_T(1 + \mathbb{E}[|\xi|^p]), \tag{1.5}$$

and a constant K_T such that

$$\mathbb{E}\left[|X_t - X_s|^p\right] \leq K_T(1 + \mathbb{E}[|\xi|^p])|t - s|^{p/2}, \quad \forall s, t \in [0, T]. \tag{1.6}$$

Moreover, if $\hat{\xi}$ is another $\mathcal{F}_t$-measurable random variable valued in $\mathbb{R}^d$ and $\hat{X}_t$ is the corresponding strong solution of (1.1) starting from $\hat{\xi}$ at time t, then for any $T > 0$, there exists a $K_T > 0$ such that

$$\mathbb{E}\left[\sup_{t\leq s\leq T} |X_s - \hat{X}_s|^p\right] \leq K_T\mathbb{E}\left[|\xi - \hat{\xi}|^p\right]. \tag{1.7}$$

Proof. This is a standard result and one can find proof in the book of Krylov (2008) or Ikeda and Watanabe (2014). □

Theorem 1.2.3. Assume the conditions (1.2), (1.3) and (1.4) hold.

There exists a constant C (depending on K in (1.2)) such that for all $t \leq \theta$ in $\mathbb{T}$ and $x \in \mathbb{R}^d$

$$\mathbb{E}\left[\sup_{t\leq s\leq\theta}|X_s^{t,x}|^2\right] \leq C|x|^2 + Ce^{C(\theta-t)}\mathbb{E}\left[\int_t^\theta |x|^2+|\kappa_u|^2du\right] \tag{1.8}$$

$$\mathbb{E}\left[\sup_{t\leq s\leq\theta}|X_s^{t,x}-x|^2\right] \leq Ce^{C(\theta-t)}\mathbb{E}\left[\int_t^\theta |x|^2+|\kappa_u|^2du\right]. \tag{1.9}$$

There exists a finite positive constant β_0 such that for all $0 \leq t \leq s$ in $\mathbb{T}$ and $x, y \in \mathbb{R}^d$

$$\mathbb{E}\left[\sup_{t\leq u\leq s}|X_u^{t,x}-X_u^{t,y}|^2\right] \leq e^{2\beta_0(s-t)}|x-y|^2. \tag{1.10}$$

Proof. A proof of these estimates can be found in Krylov (2008). □

Notice that the inequality (1.9) implies that

$$\lim_{h\downarrow 0}\mathbb{E}\left[\sup_{s\in[t,t+h]}|X_s^{t,x}-x|^2\right] = 0. \tag{1.11}$$

1.2.3 Optimal Transport

Theorem 1.2.4. (Kantorovich duality). Let X and Y be Polish spaces, let $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, and let $c : X \times Y \to \mathbb{R}^+ \cup \{\infty\}$be a lower semi-continuous cost function.

Whenever $\pi \in \mathcal{P}(X \times Y)$ and $(\varphi, \psi) \in L^1(d\mu) \times L^1(d\nu)$, define

$$I(\pi) = \int_{X\times Y} c(x,y)d\pi(x,y), \quad J(\varphi,\psi) = \int_X \varphi d\mu + \int_Y \psi d\nu. \tag{1.12}$$

Define $\Pi(\mu, \nu)$ to be the set of all Borel probability measures π on $X \times Y$ such that for all measurable subsets $A \subset X$ and $B \subset Y$,

$$\pi(A\times Y) = \mu(A), \quad \pi(X\times B) = \nu(B), \tag{1.13}$$

and define Φ_c to be the set of all measurable functions $(\varphi, \psi) \in L^1(d\mu) \times L^1(d\nu)$ satisfying

$$\varphi(x) + \psi(y) \leq c(x,y) \tag{1.14}$$

for $d\mu$-almost all $x \in X$, $d\nu$-almost all $y \in Y$.

Then

$$\inf_{\Pi(\mu,\nu)} I(\pi) = \sup_{\Phi_c} J(\varphi,\psi). \tag{1.15}$$

Moreover, the infimum in the left-hand of (1.15) is attained. Furthermore, it does not change the value of the supremum in the right-hand side of (1.15) if one restricts the definition of Φ_c to those functions (φ, ψ) which are bounded and continuous.

Proof. A proof of this theorem can be found in Villani (2003). □

1.2.4 Well known inequalities

We will make use of the following inequalities in analysis often.

(1) Cauchy–Schwarz inequality. For all vectors u and v of an inner product space it is true that

$$|\langle u, v\rangle|^2 \leq \langle u, u\rangle \cdot \langle v, v\rangle. \tag{1.16}$$

(2) Hölder's inequality. Let (S, Σ, μ) be a measure space and let $p, q \in (1, \infty)$ with $1/p + 1/q = 1$. Then, for all measurable real- or complex-valued functions f and g on S,

$$||fg||_1 \leq ||f||_p ||g||_q. \tag{1.17}$$

(3) Markov's inequality. If X is a non-negative random variable and $a > 0$, then the probability that X is at least a is at most the expectation of X divided by a:

$$P(X \geq a) \leq \frac{\mathbb{E}[X]}{a}. \tag{1.18}$$

(4) Jensen's inequality. If X is a random variable, $f : \mathbb{R} \to \mathbb{R}$ is a convex function, $f(\mathbb{E}[X])$ and $\mathbb{E}|f(X)|$ are finite, then

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]. \tag{1.19}$$

1.3 Background

Over the past decades, there has been a vast amount of research on portfolio allocation. Perhaps one of the most iconic results is the portfolio selection theory by Markowitz (1952, 1959), which is also referred to as 'modern portfolio theory (MPT)'.

Markowitz transformed two proverbs — nothing ventured, nothing gained, and do not put all your eggs into one basket — into rigorous theorems for portfolio optimization. According to Markowitz, with a given risk level, a wise investor should maximize the expected return (or minimize the risk for a given expected return). With this breakthrough achievement, Markowitz won the Nobel Prize in Economic Sciences. MPT can be expressed with the formulation

$$\begin{cases} \max & \mathbb{E}\left[R_1\alpha_1 + R_2\alpha_2 + \ldots + R_n\alpha_n\right] \\ \text{s.t.} & \operatorname{var}\left(R_1\alpha_1 + R_2\alpha_2 + \ldots + R_n\alpha_n\right) \leq \beta, \\ & \alpha_1 + \alpha_2 + \ldots + \alpha_n = 1, \\ & \alpha_i \geq 0, i = 1, 2, \ldots, n, \end{cases} \tag{1.20}$$

where α_i represent the fraction of wealth invested in security i, R_i denotes the return for the i-th security, and β is the risk level that the investor can tolerate.

Markowitz's theory only provides a starting point, as it still has some limitations. It is a static model, where the decision is only made once at the beginning of the time horizon. In the real world, the financial market is dynamic and constantly changing. Hence, continuous-time portfolio selection models have emerged, see, for example, Merton (1969, 1971), Karatzas et al. (1987) and Bajeux-Besnainou and Portait (1998).

In a landmark paper, Merton (1969) introduces a continuous-time model to maximize the expected utility within a fixed time horizon, and he found an explicit solution for the problem of optimal portfolio selection and consumption, for a constant relative risk aversion (CRRA) utility function $\frac{X^\gamma}{\gamma}$, $\gamma \in (0, 1)$ (a.k.a. power utility or isoelastic utility). He found that the optimal fraction of the wealth to be invested in the risky asset is given by $\pi^* = \frac{\mu - r}{\sigma^2(1-\gamma)}$[1], which is independent of both time and the current wealth.

Merton's problem has led to various extensions, some of which are illustrated in the textbook by Rogers (2013). In the first stream, researchers vary the time horizon. For example, the agent lives for a random time τ which follows some distribution independent of the evolution of the assets. Or the agent may choose to stop at some stopping time τ of his choice and receive an immediate reward $U(X_\tau)$. In the second stream, researchers put some practical constraints on the wealth. For example, Elie and Touzi (2008) treated drawdown constraint on the wealth, i.e. $X_t \geq c \sup\limits_{s \leq t} X_s$, $\forall t$, where c is a fixed constant between $(0, 1)$. Another kind of constraint deals with the expected losses below a threshold, i.e. expected shortfall constraint. Agents maximize their expected utility subject to $\mathbb{E}\left[(\overline{X} - X_T)^+\right] \leq \eta$, where $\overline{X}$ and η are constants. In practice, the investors are not allowed to trade at every instant of time and all trades incur transaction costs. The optimization problem without any transaction costs or trading constraints will lead to an unrealistic portfolio selection strategy. Hence, some research take transaction costs into account while maximizing the objective function, for example, Oksendal and Sulem (2002), Liu and Loewenstein (2002), Dai and Zhong (2008). In the third stream, researchers consider stochastic parameters. For example, the interest rate is no longer a constant but following a Vasicek process instead, or the drift can follow an Ornstein–Uhlenbeck process (Rogers, 2013). In the research

[1] Here, μ is the expected rate of asset returns, σ^2 is the variance of the asset returns, r is the risk-free interest rate and $1 - \gamma$ is the relative risk aversion constant.

regarding non-constant volatility, people usually consider a stochastic volatility model (see, for example, Matoussi et al. (2015)).

It is obvious that the distributions of security returns are unknown. In the implementation of portfolio optimization, investors usually estimate the mean, variance and interest rate using historical data. However, the optimal portfolios can be sensitive to errors in estimating the parameters, or even more generally assuming that the past is a reliable prediction of the future. To reduce the effect of estimation errors (aka misspecification), robust techniques are applied to construct more stable portfolios.

There has been a substantial amount of literature on robust portfolio optimization over the last decade and the area is still developing. A comprehensive introduction can be found in the book by Fabozzi et al. (2007). Gabrel et al. (2014) provided an overview of advances in robust optimization, including, but not limited to applications in finance, where they stated that "robustifying" model calibration is one of the most important area of research that should be looked after following the 2007 financial crisis. We list below a few pieces of influential research in this direction. For instance, Elliott and Siu (2009) supposed that an agent wants to maximize the minimized utility function, over a family of probability measures. This problem was then formulated as a Markovian regime-switching model, where the market parameters are modulated by a continuous-time finite-state Markov chain. Fouque et al. (2016) studied an asset allocation problem with stochastic volatility and uncertain correlation, and derived closed-form solutions for a class of utility functions. Ismail and Pham (2019) studied a robust Markowitz portfolio selection problem under covariance uncertainty. The value function is obtained by optimizing the worst-case mean-variance functional, over the admissible investing strategies α. They then solved this problem by the McKean–Vlasov dynamic programming approach and characterized the solution with a Bellman–Isaacs partial differential equation (PDE). They also characterized the robust efficient frontier in two examples: uncertain volatilities and uncertain correlation. Last but not least, we also mention the work by Talay and Zheng (2002), which studied the robust optimization problem in the context of derivatives hedging.

In parallel to the above, there is also a rich body of literature discussing the robustness issue in economics, for example, the book by Hansen and Sargent (2008). In their papers (e.g., Hansen and Sargent 2007, Hansen et al. 2005 and Hansen and Sargent 2011), the decision maker starts with a single approximation model. He surrounds the approximation model with a set of alternative models whose relative entropies are restricted or penalized. The goal is to make a decision which can perform well for all the models of this set. Based on the framework by Hansen and Sargent, Glasserman and Xu (2013) studied robust risk measurement under model misspecification.

The systems studied in this thesis are dynamic, and their evolution is described by Itô's stochastic differential equations. Since the system is dynamic, the decision makers must make decisions based on the updated information, and the optimal decision will also evolve continuously over time. Such optimization problems are called *stochastic optimal control problems*. For basic notations, definitions and properties of stochastic processes, we refer to the books by Karatzas et al. (1998) and Karatzas and Shreve (2012).

Stochastic optimal control problems commonly appear in the fields of physics, biology, economics, etc. They also arise in various financial applications, where the investment strategy is the "control variable". In the following chapters, we will apply stochastic control techniques to solve dynamic portfolio optimization problems. We refer interested readers to ElKouri (1981), Yong and Zhou (1999), Pham (2009) and Nisio (2015) for a overview of stochastic control analysis. When solving a stochastic optimal control problem, Pontryagin's maximum principle and Bellman's dynamic programming are two commonly used tools. In this study we will only focus on Bellman's dynamic programming. This approach is based on obtained a so-called value function, which is solution to a second order PDE. This PDE is called a Hamilton–Jacobi–Bellman (HJB) equation. Another PDE we will often use in the following chapters is called a Fokker–Planck equation, aka the Kolmogorov forward equation. It describes the time evolution of the probability density function of the velocity of a particle under the influence of drag forces and random forces, as in Brownian motion. The Fokker-Planck equation is used with problems where the initial distribution is known. With the known drift and diffusion coefficients, we can compute the distribution at a latter time.

Optimization is a thriving discipline, where the idea of convexity is at the central place. Among the computational optimization techniques, we will frequently apply duality-based algorithms. Convex conjugate is a commonly used tool when we want to find the dual problem of a convex minimization problem. Good references about convex analysis already exist, such as the classic Rockafellar (1970), Rockafellar and Wets (2009), and Boyd and Vandenberghe (2004).

1.4 My contributions

1.4.1

The first part of the thesis (Chapter 2) deals with the problem of utility maximization under uncertain parameters. It led to the completion of the research article Guo et al. (2019a). Our portfolio optimization problem was first inspired by the Merton problem (Merton, 1969). In the original Merton problem, the evolution of the risky asset, although stochastic by essence, is governed by the Black-Scholes model (Black and Scholes, 1973) with *observable* parameters μ, r and σ. This assumption clearly does not match the reality that investors are facing. Therefore, in this work, we take into account the uncertainty of the drift and diffusion coefficients in the model, leading to a robust portfolio opimization problem. The concept of *robust portfolio optimization* was first introduced in the operations research literature by El Ghaoui and Lebret (1997) and Ben-Tal and Nemirovski (1998). Instead of assuming a model with a known drift, interest rate or volatility, the problem of *robust* optimal allocation assumes that they will evolve dynamically in the most unfavourable way within a given range. The resulting allocation process tends to be more stable and less vulnerable to changes and misspecifications in model parameters.

The first novelty of this work is that we do not assume a given range of parameters in the evolution of the underlying process. The first novelty of this work is that we do not define specific lower and upper bounds for parameters in the evolution of the underlying process. In other papers considering uncertain volatility (or drift), the authors assume the admissible volatility $\sigma \in [\sigma_{\min}, \sigma_{\max}]$, where $\sigma_{\min}$ and $\sigma_{\max}$ are model bounds defining the uncertainty of the amplitude of future prices fluctuations. Instead, we allow the parameters to move freely and use a penalty function $F = F(r, \mu, \sigma, \dots)$ to penalize unrealistic values of the parameters. In our problem, we consider a set $\mathcal{A}$ for the portfolio allocation strategies and a set $\mathcal{B}$ for the drift and diffusion coefficients, the value function is defined by

$$\overline{v}(t,x) = \sup_{\Gamma\in\mathcal{N}} \inf_{\theta\in\mathcal{B}} \left\{ \mathbb{E}^{t,x}\left[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta}) \right] \right\}, \tag{1.21}$$

where $\theta := (\Sigma(\cdot), \mu(\cdot))$, $\alpha \in \mathcal{A}$ is the portfolio allocation process, and the strategy Γ maps $\mathcal{B} \to \mathcal{A}$, λ_0 is the intensity of the penalty function.

We can interpret this robust utility maximization problem as a two-player zero-sum stochastic differential game (SDG). We first show the existence of a value function for the differential game in Proposition 2.3.2. In the classical papers studying two-player zero-sum SDGs, Fleming and Souganidis (1989) and Nisio (2015) made the assumptions that the utility function U is bounded and Lipschitz continuous. The present work extends these results to more general assumptions by considering an unbounded domain and an unbounded utility function U. We prove that the value function (1.21) satisfies the Dynamic Programming Principle in Theorem 2.3.1. We then prove that our value function is the unique viscosity solution (see, for example, Crandall and Lions (1983) for the introduction of viscosity solution) of an HJBI equation in Theorem 2.3.2 and Theorem 2.3.3.

We then consider a particular utility function $U(x) = \ln(x)$. With this specifically chosen utility function, we can write the terminal wealth X_T explicitly and get the analytical solution of the optimal controls, as in the following

$$0 = \hat{\sigma}_s^4 - \sigma_0 \hat{\sigma}_s^3 - \frac{(\mu - r)^2}{2\lambda_0}. \tag{1.22}$$

To simulate a more realistic scenario, we add some noise to the reference covariance matrix and simulate portfolios with robust and non-robust strategies, respectively. Figure 2.4(a) tells us how far should our reference covariance deviate from reality for the robust portfolio to outperform the non-robust portfolio. With Figure 2.4(b), we can choose the best λ_0 in (1.21) for robust portfolio allocation based on our confidence in the reference covariance. Furthermore, we test our robust approach by constructing two empirical portfolios using real market data. Figure 2.6 depicts how the robust and non-robust expected utilities change w.r.t. λ_0. For a given amount of noise, the robust portfolio may underperform for small values of λ_0, however the robust expected utility will increase gradually as λ_0 increases. Finally, as λ_0 diverges to infinity, the robust expected utility will converge to the non-robust one.

In the numerical part, we first compute the value function with a PDE method. Starting from the terminal condition $\overline{v}(T,x)$, we work backward to compute $\overline{v}(t,x)$ via an implicit finite difference method. Since the Hamilton–Jacobi–Bellman–Isaacs (HJBI) equation is non-linear, in order to use the implicit finite difference method, we introduce a novel iteration scheme based on the Legendre transform of the nonlinearity. The algorithm for this method is summarized in Algorithm 1. Figure 2.9 compares the PDE estimated $\overline{v}(t,x)$ with the analytical solution for a logarithmic utility function. Figure 2.10(a) shows the simulated value $\overline{v}(t,x)$ for the power utility function.

Another novelty of this work is that we connect Generative Adversarial Networks (GANs) with zero-sum differential games, and we devise a deep learning-based algorithm to solve this problem. In general, GANs can be interpreted as minimax games between the generator and the discriminator, whereas our problem is a minimax game between the agent who controls the portfolio allocation and the market who controls the model parameters. To our knowledge, it is the first time GANs are used to solve a robust optimization problem in the field of quantitative finance. GANs are an exciting recent innovation in machine learning. Cao et al. (2020) reviewed the minimax structures underlying GANs, and they established theoretical connections between GANs and Mean-Field Games. However, there are few applications of GANs in quantitative finance so far, except Wiese et al. (2020) and Cuchiero et al. (2020). Inspired by GANs' ability to generate images, they approximated a realistic asset price simulator using adversarial training techniques.

Our GANs are composed of two neural networks; one generates α (α-generator), the other generates σ (σ-generator). They compete against each other during training. As training advances, the game may end up at a Nash Equilibrium. A demonstration of the simplified network architecture is illustrated in Figure 2.11. We summarized this training process in Algorithm 2. We first apply it to the utility function $U(X_T) = \ln(X_T)$. Figure 2.12(a) compares the learned value functions with the true values for a range of λ_0. The value functions estimated with GANs show good accuracy compared to the true ones. Then we also consider the utility function $U(X_T) = 3X_T^{\frac{1}{4}}$, and compare the GANs estimated values with the PDE estimations in Figure 2.13(a).

Finally, we implement a Regression Monte Carlo scheme (Longstaff and Schwartz, 2001) to solve the same robust portfolio allocation problems. The obstacle in the implementation is that we are not able to simulate the paths forward, since the dynamics of the state variable depends on the uncertain controls. Following Kharroubi et al. (2014), one way to tackle this problem is an initial randomization of the controls. To be specific, we choose an arbitrary initial distribution for the controls and simulate the state variable with these dummy controls. Then, inspired by the Dynamic Programming Principle, we can start from the known terminal condition and compute the value functions backward in time recursively. In this process, we use a least-squares regression to approximate the conditional expectation. The complete process is shown in Algorithm 3. For a logarithmic utility function, Figure 2.16 compares the Monte Carlo simulation values, finite difference results and analytical values. It shows that both the PDE and Monte Carlo approach the true value in this example. Figure 2.18 shows Monte Carlo and finite difference approximations for a power utility function.

1.4.2

In the second part of this thesis (Chapter 3), we consider a portfolio allocation problem, where the goal is to reach a prescribed wealth distribution at the terminal time. In this problem, the prices of the risky assets are modelled by a semimartingale, with prescribed drift (μ_t) and diffusion (Σ_t) coefficients. By controlling the portfolio allocation between a risky and a risk-free asset, the investor wants the distribution of the terminal wealth to match, or be close to, a given target distribution. This work led to the article Guo et al. (2020a), currently under review for publication.

The novelty of this work is to provide a new perspective on portfolio optimization and connect it with optimal transport. The classical objective function in a portfolio optimization problem is to maximize the expected return given variance level. However, the first and second moments of the return of a portfolio is only a simplified two parameters description of the wealth. In our problem setting, the whole distribution of the portfolio wealth would provide investors complete information. The challenge of finding a continuous semimartingale with prescribed distributions at given times can be addressed using the optimal transport (OT) theory.

We denote by $\rho_t := \mathbb{P} \circ X_t^{-1} \in \mathcal{P}(\mathbb{R})$ the distribution of X_t. In this problem, we know the initial distribution of the portfolio wealth $\rho_0 \in \mathcal{P}(\mathbb{R})$, and we are given a prescribed terminal distribution $\overline{\rho}_1 \in \mathcal{P}(\mathbb{R})$. With ρ_0 and a suitable portfolio allocation process α, the realized terminal distribution of the portfolio wealth is $\rho_1 := \mathbb{P} \circ X_1^{-1}$. We also introduce a functional $C(\rho_1, \overline{\rho}_1)$ to penalize the deviation of ρ_1 from $\overline{\rho}_1$. To make sure the feasible set is convex, in Notation 3.2.1, we define the maps $\tilde{B}(t,x) := \alpha_t^\intercal \mu_t x$, $\tilde{A}(t,x) := \alpha_t^\intercal \Sigma_t \alpha_t x^2$, and we let $B(t,x) := \tilde{B}\rho$, $A(t,x) := \tilde{A}\rho$. Then the dynamics of the portfolio wealth is given by

$$dX_t = \tilde{B}(t,X_t)dt + \tilde{A}^{\frac{1}{2}}(t,X_t)dW_t, \tag{1.23}$$

$$X_0 = x_0. \tag{1.24}$$

In Proposition 3.2.1, we state the necessary and sufficient condition for the existence of $\alpha_t \in \mathbb{R}^d$. Then based on Proposition 3.2.1, we constrain A, B in a feasible set as $\mathit{\Pi} := \{(\rho, B, A) : A \geq \frac{(B^+)^2}{\|\nu_t\|^2 \rho}\}$ ($B^+ := \max(0,B)$, $\nu_t := \Sigma_t^{-\frac{1}{2}}\mu_t$). In this problem, we use a convex cost function $F : \mathcal{E} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \cup \{+\infty\}$ to penalize measures out of the set $\mathit{\Pi}$, and we want to solve the infimum of the functional

$$V(\rho_0, \overline{\rho}_1) = \inf_{\rho,B,A} \int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho + C(\rho_1, \overline{\rho}_1) \tag{1.25}$$

over all $(\rho, B, A) \in \mathit{\Pi}$ satisfying the Fokker–Planck equation and the initial distribution

$$\partial_t \rho(t,x) + \partial_x B(t,x) - \frac{1}{2}\partial_{xx} A(t,x) = 0 \quad \forall (t,x) \in \mathcal{E}, \tag{1.26}$$

$$\rho(0,x) = \rho_0(x) \quad \forall x \in \mathbb{R}. \tag{1.27}$$

While solving this problem, we first compute the convex conjugate of the cost functional, given in (3.18). Then, in Theorem 3.3.1, we introduce the dual problem. This proof is an application of the Fenchel–Rockafellar duality theorem, e.g., Brezis (2010, Theorem 1.12).

In Corollary 3.3.1, we further simplify the form of the dual problem, given in

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ -C^*(-\phi_1) - \int_{\mathbb{R}} \phi_0 d\rho_0 \right\}. \tag{1.28}$$

Instead of looking for the supremum over all functions $\phi = (\phi_t)_{t\in[0,1]}$, we only need to find the optimal ϕ_1, which is ϕ evaluated at the terminal time. Then the function $\phi(t,x)$ is a viscosity solution of the Hamilton–Jacobi–Bellman equation

$$\begin{cases} -\phi_t - \sup_{\tilde{A} \geq \frac{(\tilde{B}^+)^2}{\|\nu_t\|^2}} \left[\phi_x \tilde{B} + \frac{1}{2}\phi_{xx}\tilde{A} - F(\tilde{B}, \tilde{A}) \right] = 0, & \text{in } [0,1) \times \mathbb{R}, \\ \phi(1,x) = \phi_1(x), & \text{on } [1] \times \mathbb{R}. \end{cases} \tag{1.29}$$

Because the minimal objective function (1.25) is a trade-off between the cost function and the penalty functional, the optimal ϕ_1 in the dual problem (3.29) will not in general ensure that ρ_1 reaches $\overline{\rho}_1$, unless the penalty functional is infinity for $\rho_1 \neq \overline{\rho}_1$. When $\overline{\rho}_1$ is attainable, it can be realized by choosing the penalty functional as an indicator function

$$C(\rho_1, \overline{\rho}_1) = \begin{cases} 0 & \text{if } \rho_1 = \overline{\rho}_1, \\ +\infty & \text{if } \rho_1 \neq \overline{\rho}_1. \end{cases} \tag{1.30}$$

Using the indicator penalty functional (1.30) is equivalent to adding the terminal constraint $\rho_1 = \overline{\rho}_1, \forall x \in \mathbb{R}$. This coincides with the ‘usual’ optimal transport problem. When $C(\rho_1, \overline{\rho}_1)$ is defined as (1.30), we provide the dual problem in Corollary 3.3.2.

We then solve the dual problem using numerical methods. In particular, we use a gradient descent-based method to find the optimal ϕ_1. A crucial ingredient in the gradient descent method is to characterize optimality condition. By providing a gradient, the computation is faster and more accurate. We show that the optimal terminal function ϕ_1 should satisfy the optimality condition (3.45). When $C(\overline{\rho}_1, \rho_1)$ is defined as (1.30), the corresponding optimality condition can be expressed as $-\overline{\rho}_1 + \rho_1 = 0, \quad \forall x \in \mathbb{R}$. We know $\phi(t,x)$ is the solution of the HJB equation (1.29). For a given terminal function ϕ_1, we can calculate ϕ_0 by solving the HJB equation backward with a finite difference method. In Algorithm 4, we state the gradient descent-based algorithm to look for the optimal ϕ_1 in (3.29). It includes solving the HJB equation and the Fokker–Planck equation with a finite difference method combined with a fixed-point iteration.

In the numerical results, we give examples for general target distributions with various penalty functionals. In the first example, we use $L2$ norm as the penalty functional and $\overline{\rho}_1 = \mathcal{N}(6,1)$. As shown in Figure 3.1, ρ_1 moves closer to the target $\overline{\rho}_1$ as we increase the intensity of the penalty. Compared to other research (e.g., Chen et al. 2018, Chen et al. 2019b) where the prescribed distributions are restricted to Gaussian, our method applies to a large choice of $\overline{\rho}_1$, including heavy-tailed and asymmetric distributions. In Figure 3.3, we illustrate that we can attain the target when $\overline{\rho}_1$ is a mixture of two Normal distributions. Then we use the Kullback–Leibler (K–L) divergence as the penalty functional, and the results are presented in Figure 3.4.

Next, we relax the problem by allowing the investor to inject or withdraw cash during the investment process. As stated in Proposition 3.2.1, we always have $(\tilde{B}^+)^2 \leq \|\nu_t\|^2 \tilde{A}$ for a self-financing portfolio. When the prescribed terminal distribution is not ambitious enough, it is possible that the optimal drift $\tilde{B}(t,x)$ and diffusion $\tilde{A}(t,x)$ lie in the interior of the feasible set, i.e., $\tilde{B}(t,x)^2 < \|\nu_t\|^2 \tilde{A}(t,x)$. In this case, instead of using this unsaturated drift $\tilde{B}(t,x)$ to reach the prescribed terminal distribution, we can choose to use the drift $\tilde{B}(t,x) = \|\nu_t\| \sqrt{\tilde{A}}$ to attain a more ambitious distribution. We define the concept of *cash saving* at time t as $c_t := \|\nu_t\| \sqrt{\tilde{A}(t,x)} - \tilde{B}(t,x)$. We show that we can reach a better terminal distribution, in the sense that the terminal wealth has a higher expected value, when we take *cash saving* into account. As shown in Figure 3.5(a), with a relatively conservative target, although ρ_1 has attained the target, the distribution for the *wealth with cash saving* attains a higher value than the target distribution.

Finally, by allowing the investor to inject cash during the process, we can remove the constraint $(\tilde{B}^+)^2 \le \|\nu_t\|^2 \tilde{A}$ and we allow $\tilde{B} \in \mathbb{R}$ instead. In the cost function, we use the term $K(\tilde{B}^2 - \|\nu_t\|^2 \tilde{A})$ to penalize the amount of cash input. By varying K, we can control the strength of penalty and hence control the cash input flow. To see the effect of cash input, we demonstrate two examples with unattainable targets in Figure 3.7. It shows that we can steer the empirical terminal density ρ_1 to $\overline{\rho}_1$ by inputting cash wisely.

1.4.3

The third and last part of the thesis (Chapter 4) focuses on the numerical solution of optimal transport problems. This time, we consider a general setting, where the state variable is a multi-dimensional semimartingale.

In the field of finance, optimal transport has been applied for robust hedging (Dolinsky and Soner 2014, Henry-Labordère 2017) and volatility calibration (Guo et al. 2019b). Although we have gained tremendous theoretical insight, the numerical solution of the problem remains challenging. When the dimension is less or equal to three, many state-of-art approaches are able to compute the global solution effectively; see, for example, the review by Zhang et al. (2020). However, when the distributions live in spaces of dimension four or more, these traditional methods face the curse of dimensionality. Our contribution in this work is to propose two deep neural network-based algorithms for solving optimal transport problems. Both methods are mesh-free and therefore mitigate the curse of dimensionality. These methods can be widely applied to solving optimal transport as well as stochastic optimal control problems.

In this part, we consider a stochastic process $(X_t)_{t\in[0,1]}$ valued in $\mathbb{R}^d$, which solves the SDE

$$dX_t = B(t, X_t)dt + A(t, X_t)dW_t, \tag{1.31}$$
$$X_0 = x_0, \tag{1.32}$$

where $B : \mathcal{E} \to \mathbb{R}^d$ and $A : \mathcal{E} \to \mathbb{R}^{d\times d}$ is defined such that $AA^\intercal = \mathcal{A}$. As in the previous work, the process X_t has two predefined marginal distributions, ρ_0 at time $t = 0$ and $\overline{\rho}_1$ at time $t = 1$.

The first algorithm is based on a relaxation/penalization of the terminal constraint, and then the primal problem is solved using deep neural networks. In particular, we let $F(B_t, \mathcal{A}_t)$ denote the cost function, and we use the functional $C(\rho_1, \overline{\rho}_1)$ to penalize the deviation of ρ_1 from $\overline{\rho}_1$. We are interested in solving the infimum of the functional

$$V(\rho_0, \overline{\rho}_1) = \inf_{\rho,B,\mathcal{A}} \left\{ \int_{\mathcal{E}} F(B_t, \mathcal{A}_t) d\rho(t,x) + C(\rho_1, \overline{\rho}_1) \right\}, \tag{1.33}$$

over all $(\rho, B, \mathcal{A}) \in \mathcal{P}(\mathbb{R}^d) \times \mathbb{R}^d \times \mathbb{S}^d_+$ satisfying the initial distribution

$$\rho(0, x) = \rho_0(x) \quad \forall x \in \mathbb{R}^d, \tag{1.34}$$

and the Fokker–Planck equation

$$\partial_t \rho(t,x) + \nabla_x \cdot (B(t,x)\rho(t,x)) - \frac{1}{2} \sum_{i,j} \partial_{ij}(\mathcal{A}_{i,j}(t,x)\rho(t,x)) = 0 \quad \forall (t,x) \in \mathcal{E}. \tag{1.35}$$

In this method, we discretize the period $[0,1]$ into N constant time steps and construct a neural network θ_n for each time step. Then we use neural networks to look for the optimal $(B_t)_{t\in[0,1]}$ and $(A_t)_{t\in[0,1]}$, namely $(B_n, A_n) \approx \theta_n(X_n)$. With B_n, A_n and X_n, we can compute the state variable at the next time step from the dynamics

$$X_{n+1} = X_n + B_n \Delta t + A_n \Delta W_n, \quad \forall n \in [0, N-1].$$

We denote by M the number of Monte Carlo paths. At the final time step N, we can use the terminal wealth samples $X_N^m, \forall m \in [1, M]$ to estimate ρ_1 as $\tilde{\rho}_1$ using kernel density estimation (KDE) (Hastie et al., 2009). Because KDE is not an unbiased estimator of the true density, we also estimate $\tilde{\overline{\rho}}_1$ from $\overline{\rho}_1$ with the same KDE. Using gradient descent-based algorithms, the training will search for the optimal neurons $\hat{\theta}_{n,n\in[0,N-1]}$ which minimize the loss function:

$$\hat{\theta}_{n,n\in[0,N-1]} = \arg\inf_{\theta} \left\{ \frac{1}{M} \sum_{m=1}^{M} \left\{ \sum_{n=0}^{N-1} F(\theta_n(X_n^m))\Delta t \right\} + C(\tilde{\rho}_1, \tilde{\overline{\rho}}_1) \right\}.$$

The algorithm is summarized in Algorithm 5. We illustrate the numerical result for a 2-dimensional example, where $\overline{\rho}_1$ is a bivariate normal distribution. The contours of the empirical and target distributions are presented in Figure 4.2.

In the second method, we consider the indicator penalty function

$$C(\rho_1, \overline{\rho}_1) = \begin{cases} 0 & \text{if } \rho_1 = \overline{\rho}_1, \\ +\infty & \text{if } \rho_1 \neq \overline{\rho}_1, \end{cases} \tag{1.36}$$

and one recovers the "usual" semi-martingale optimal transport problem. In this case, we provide the dual formulation in Theorem 4.4.1 and then express the dual problem as as a saddle point problem:

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \inf_{B\in\mathbb{R}^d, \mathcal{A}\in\mathbb{S}_+^d} \left\{ \int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1 - \mathbb{E}\left[\phi_1(X_1) - \int_0^1 F(B, \mathcal{A}) dt | X_0 = x_0 \right] \right\}. \tag{1.37}$$

This dual saddle point formulation of the optimal transport problem is reminiscent of GANs: recall that GANs can be interpreted as minimax games between the generator and the discriminator, whereas our problem is a minimax game between ϕ_1 and $(\mathcal{A}, B)$. Inspired by this connection, we use adversarial networks, as described in Chapter 2, to estimate the value (1.37). The algorithm is stated in Algorithm 7.

This second method is free from the *curse of dimensionality*. Hence it is a promising quantitative techniques for solving high-dimensional optimal transport problems. We test the performance and accuracy of Algorithm 7 on 5-d and 10-d examples. The empirical marginal distributions of the trained sample sets are shown in Figure 4.8 and Figure 4.13, respectively. The mean and covariance matrix of the empirical distributions are also provided. As high dimensional distributions are difficult to visualize, in addition to the graphical check, we also introduce a loss metric to measure the quality of the solution. Firstly, we will apply a number of random affine transformations on the dataset. Then, we propose a loss metric adapted from the Wasserstein distance. For each affine transformation, we compute the loss metric between the empirical and theoretical distribution. Figure 4.11 and 4.16 show the histogram of the loss metrics for the 5-d case and 10-d case, respectively.

We illustrate the method with an application in finance, where we implement Algorithm 5 to the problem of optimal portfolio selection with a prescribed terminal density studied in Chapter 3. The goal is, given an initial wealth distribution ρ_0, to find an investment strategy that attains a prescribed terminal wealth distribution $\overline{\rho}_1$. The objective function in this application is

$$\inf_{\alpha,\rho} \left\{ \int_{\mathcal{E}} F(\alpha_t) d\rho(t,x) + C(\rho_1, \overline{\rho}_1) \right\}, \tag{1.38}$$

where $F(\alpha_t)$ is a convex cost function in the portfolio allocation α_t. We can apply Algorithm 5 to solve for the value (1.38). In the numerical results, we present several examples with various penalty functionals, including $L2$ norm (Figure 4.18), Kullback–Leibler divergence (Figure 4.20) and 2-Wasserstein distance (Figure 4.21). Note that our method is not restricted to Gaussian target distributions; for example, we use a mixture of two normal distributions as the target $\overline{\rho}_1 = 0.5\mathcal{N}(4,1) + 0.5\mathcal{N}(7,1)$ in Figure 4.20 and the result is satisfactory.

Chapter 2

Robust utility maximization under model uncertainty via a penalization approach

This chapter addresses the problem of utility maximization under uncertain parameters. In contrast with the classical approach, where the parameters of the model evolve freely within a given range, we constrain them via a penalty function. In addition, this work dedicates in proposing various numerical algorithms to solve for the value function, including finite difference method, Generative Adversarial Networks and Monte Carlo simulation. These methods contribute to the quantitative techniques for solving robust portfolio optimization problems. We show that this robust optimization process can be interpreted as a two-player zero-sum stochastic differential game. We prove that the value function satisfies the Dynamic Programming Principle and that it is the unique viscosity solution of an associated Hamilton–Jacobi–Bellman–Isaacs (HJBI) equation. By testing this robust algorithm on real market data, we show that robust portfolios generally have higher expected utilities and are more stable under strong market downturns.

2.1 Introduction

In the problems of continuous-time utility maximization, besides the choice of utility function, a key element in the formulation is the a priori knowledge assumed for the evolution of the underlying assets (e.g., the expected returns and the quadratic covariation of the diffusion process). In the original Merton problem (Merton, 1969), the evolution of the risky asset, although stochastic by essence, is governed by the Black-Scholes model (Black and Scholes, 1973) with fixed parameters μ, r and σ. This is a very simplistic model for the underlying asset price. Stochastic models (for the volatility and interest rates) that describe the price evolution more realistically have later emerged. Several papers have addressed the problem in this context: Matoussi et al. (2015) examined the case of stochastic volatility, while Noh and Kim (2011) addressed the case of stochastic interest rates. The expected return (or drift) μ plays an essential role in the optimal allocation; even when it is considered stochastic, it is still assumed to be an *observable* input of the problem. This assumption clearly does not match the reality that investors are facing. Several works by Lakner (1995) and then Bel Hadj Ayed et al. (2017) addressed the utility maximization problem with an uncertain drift, although it was assumed to follow some form of prescribed dynamics or prior distribution.

Two decades ago, the concept of *robust portfolio optimization* had emerged. It was first introduced in the operations research literature by El Ghaoui and Lebret (1997) and Ben-Tal and Nemirovski (1998). Instead of assuming a model with a known drift, interest rate or volatility, the problem of *robust* optimal allocation assumes that they will evolve dynamically in the most unfavourable way within a given range. The resulting allocation process tends to be more stable and less vulnerable to changes and misspecifications in model parameters.

A robust investment process can be interpreted as a two-player game. On one hand, the market can be thought of as an adversarial player controlling the volatility (or the drift) in order to minimize the gains of an investor; on the other hand, the investor, who controls the allocation of the portfolio, is trying to maximize her gains under the worst possible behaviour of the market. The two controllers have conflicting interests, with the gain of one player being a loss for the other. Hence we call this competition between the investor and the market a two-player zero-sum stochastic differential game (SDG).

Differential games were first introduced by Isaacs (1965). When it comes to a continuous-time context, where a continuum of moves for each player is envisioned, a profound difficulty arises to make the formulation precise because the control choices can be changed instantaneously and the value of the game may not exist. To overcome the difficulties, Fleming (1961) & Fleming (1964) naturally started with discretizing time and approximating a dynamic game by a sequence of 'one-shot' games. Afterward, Friedman (1971) defined upper and lower δ-strategy and made use of "noise". In a similar approach, Elliott and Kalton (1972) introduced a pseudo-strategy for each player and proved that if the 'Isaacs condition' is satisfied, then the game has a value. Thanks to the pioneering work in viscosity solution by Crandall and Lions (1983) and Lions (1983), a direct proof of the existence of the value of stochastic differential games became possible. The book by Fleming and Soner (2006) serves as a concise introduction to the theory of viscosity solutions and deterministic zero-sum differential games. The first complete theory for two-player zero-sum SDG was developed by Fleming and Souganidis (1989), where they proved the existence of value functions of the games. Buckdahn and Li (2008) generalized the results of Fleming and Souganidis (1989) by considering the gain functional as a solution of a Backward Stochastic Differential Equation (BSDE). With the help of BSDE methods, they proved the Dynamic Programming Principle (DPP) for the value functions in a more straightforward approach. Some more recent works on zero-sum SDG include Hernández-Hernández and Sîrbu (2018), Baltas et al. (2019) and Cosso and Pham (2019).

The main novelty of this work is threefold. Firstly, we do not assume a given range of parameters in the evolution of the underlying process. In other papers considering uncertain volatility (or drift), the authors assume the admissible $\sigma \in [\sigma_{\min}, \sigma_{\max}]$, where $\sigma_{\min}$ and $\sigma_{\max}$ are model bounds in accordance with the uncertainty about future fluctuations. Instead, we allow the parameters to move freely and use a penalty function $F = F(r, \mu, \sigma, \dots)$ to penalize unrealistic values of the parameters. Mathematically speaking, the penalty function gives some *coercivity* to the problem so that an optimal solution can be found. This approach has been used for robust derivatives pricing in Tan et al. (2013) and Guo et al. (2017). Note that one can asymptotically recover the aforementioned approaches that involve a *fixed parameter range*, by taking the penalty function F to be 0 over a given set and $+\infty$ outside.

Secondly, compared to other papers studying robust portfolio optimization, we consider errors in parameter estimation, and analyse the problem from game theory and optimal control theory perspectives. Our formulation allows straightforward calculation of the value function and we focus on approximating the value function with numerical methods. We devise two innovative algorithms, which are Generative Adversarial Networks (GANs) and control randomization. These methods enrich the quantitative techniques for solving robust portfolio optimization problems. In particular, it is, to our knowledge, the first time GANs are used to solve a robust optimization problem in the field of quantitative finance. It is also the first application of the control randomization method (see Kharroubi et al. 2014) in the context of a robust portfolio optimization problem.

GANs are an exciting recent innovation in machine learning. The fundamental principle of GANs is to use two different neural networks as two opponents with conflicting goals, and its solution is a Nash equilibrium. Hence, GANs training is closely related to game theory. Cao et al. (2020) reviewed the minimax structures underlying GANs, and they established theoretical connections between GANs and Mean-Field Games. However, there are few applications of GANs in quantitative finance so far. The only relevant work is by Wiese et al. (2020). Being inspired by GANs' ability to generate images, they approximated a realistic asset price simulator using adversarial training techniques.

Last but not least, in the classical papers studying two-player zero-sum SDGs, Fleming and Souganidis (1989) and Nisio (2015) made the assumptions that the utility function U is bounded and Lipschitz continuous. The present work extends these results to more general assumptions by considering an unbounded domain and an unbounded utility function U. Moreover, we prove that the lower- and upper-value of the SDG (2.2)-(2.3) in fact coincide.

The rest of this chapter is organized as follows. In section 2.2, we formulate a portfolio optimization problem in a robust setting and introduce the uncertain drift and uncertain volatility processes. In section 2.3, we define the value functions for static games and two-player zero-sum SDGs. In section 2.3.1 we show that the differential game has a saddle point and as a consequence, the lower- and upper-values of the SDG coincide. We prove that the value function satisfies the DPP in section 2.3.2 and that our value function is the unique viscosity solution of an HJBI equation in section 2.3.3. In section 2.4, we derive a closed-form solution for the logarithmic utility. In section 2.5.1, we add some noise to the covariance matrix and simulate portfolios with robust and non-robust strategies, respectively. Then, in section 2.5.2, we test our robust mechanism by constructing two empirical portfolios using market data. In section 2.6.1, we provide numerical results for general utility functions using PDE techniques via finite difference methods. Finally, in section 2.6.2 and section 2.6.3, we present the algorithms and results of solving a robust portfolio optimization problem with GANs and Monte Carlo simulations via control randomization.

2.2 Problem formulation

We consider a portfolio with d risky assets and one risk-free asset compounding at a constant interest rate $r \in \mathbb{R}$. The price process of the risky assets is denoted by $S_u \in \mathbb{R}^d$ $(0 \leq u \leq T)$, and the ith element of S_u follows the dynamics

$$\frac{dS_u^i}{S_u^i} = \mu_u^i du + \sum_{j=1}^{d} \sigma_u^{ij} dW_u^j, \quad 1 \leq i \leq d, \tag{2.1}$$

with the drift μ_u takes values in $\mathbb{R}^d$, the invertible matrix σ_u takes values in $\mathbb{R}^{d\times d}$ and the covariance matrix $\Sigma_u := \sigma_u \sigma_u^\intercal$ takes value in $\mathbb{S}^d$.

We will follow the framework set in Fleming and Souganidis (1989) and Talay and Zheng (2002). We first introduce the canonical sample spaces for the underlying Brownian motion in (2.1). For each $t \in [0, T]$, we set

$$\Omega_t := (\omega \in C([t, T]; \mathbb{R}^d) : \omega_t = 0).$$

We denote by $\mathbb{F} = \mathcal{F}_{t,s}$ ($s \in [t, T]$), the complete filtration generated by the canonical process from time t to time s. Equipped with the Wiener measure $\mathbb{P}_t$ on $\mathcal{F}_{t,T}$, the filtered probability space $(\Omega_t, \mathcal{F}_{t,T}, \mathbb{P}_t, \mathbb{F})$ is the canonical sample space.

The processes $\mu(\cdot), \sigma(\cdot)$ are progressively measurable with respect to the $\mathbb{P}_t$-augmented filtration of the d-dimensional Brownian motion W_u.

Let $X_u \in \mathbb{R}$ be the value of the portfolio at time u. A portfolio allocation strategy α_u takes value in $\mathbb{R}^d$ represents the proportion of total wealth the agent invests in the d risky assets at time u, and $1 - \sum_{i=1}^d \alpha_u^i$ is the proportion invested in the risk-free asset.

Assuming the strategy is self-financed, the wealth process evolves as follows

$$\frac{dX_u}{X_u} = \sum_{i=1}^{d} \alpha_u^i \frac{dS_u^i}{S_u^i} + \left(1 - \sum_{i=1}^{d} \alpha_u^i\right) r du.$$

We define $\mathbf{r} := r \times \mathbf{1}$ with $\mathbf{1} \in \mathbb{R}^d$ being a d-dimensional ones vector. The wealth evolution can be rewritten as

$$dX_u = X_u(\alpha_u^\intercal(\mu_u - \mathbf{r}) + r)du + X_u \alpha_u^\intercal \sigma_u dW_u. \tag{2.2}$$

Now, we introduce the concept of *admissible controls.*

Definition 2.2.1. An admissible control process $\theta := (\sigma(\cdot), \mu(\cdot))$ for the market on $[t, T]$ is an $\mathcal{F}_{t,s}$ progressively measurable process taking values in a compact convex set $B \subset \mathbb{R}^{d \times d} \times \mathbb{R}^d$. The set of all admissible θ on $[t, T]$ is denoted by $\mathcal{B}(t)$.

Definition 2.2.2. An admissible control process α for the investor on $[t, T]$ is an $\mathcal{F}_{t,s}$ progressively measurable process taking values in a compact convex set $A \subset \mathbb{R}^d$. The set of all admissible α is denoted by $\mathcal{A}(t)$.

Note that although the sets for the value of the controls are compact, in practice, A and B are arbitrarily large. Next, let us define the payoff function as the expectation of a terminal utility function U plus a penalty function F:

$$J(t, x, \alpha, \theta) = \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s) ds\right], \tag{2.3}$$

where $\mathbb{E}^{t,x}(\cdot)$ denotes the expectation given the initial time and wealth $(t, x) \in [0, T] \times \mathbb{R}$ (we assume the initial wealth x is finite), and $\lambda_0 \in \mathbb{R}$ is a positive constant. Throughout the paper, we will often include α and θ in the superscript of X to indicate the dependency of the wealth process on the allocation, drift and volatility processes. Our objective is to find the optimal portfolio allocation process α that maximizes the worst-case payoff function given by the drift process μ and the covariance process Σ. Throughout the paper, F will be a convex function in θ_s.

2.2.1 Robust value functions

We are now ready to define the value functions. In our problem, the covariance and drift are unknown. We want to find the optimal portfolio allocation process that maximizes the worst-case situation given by the covariance and drift. Then, given an initial condition $(t, x) \in [0, T] \times \mathbb{R}$, this value is given by

$$\underline{u}(t, x) = \sup_{\alpha \in \mathcal{A}} \inf_{\theta \in \mathcal{B}} \left\{\mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s) ds\right]\right\}. \tag{2.4}$$

Here, $\mathcal{A}, \mathcal{B}$ depend on both t and x. We say $\hat{\alpha}$ and $\hat{\theta}$ are optimal controls if $\underline{u}(t,x) = J(t,x,\hat{\alpha},\hat{\theta}) = \inf_{\theta\in\mathcal{B}} J(t,x,\hat{\alpha},\theta)$.

This problem is known as a *static game*, and the function $\underline{u}(t,x)$ is called the *lower value of the static game*. If we reverse the moving order of the two players, we obtain the *upper value of the static game*, which is

$$\overline{u}(t,x) = \inf_{\theta\in\mathcal{B}} \sup_{\alpha\in\mathcal{A}} \left\{ \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds \right] \right\}. \tag{2.5}$$

Note that $X_s^{\alpha,\theta}, \forall s \in [t,T]$ denotes a process controlled by processes α, θ. When $X_s^{\alpha,\theta}$ starts from an initial condition (t,x), we write the expectation of $f(X_s^{\alpha,\theta})$ as $\mathbb{E}^{t,x}\left[f(X_s^{\alpha,\theta})\right]$.

2.2.2 Assumptions

In this section, we make the following assumptions which will be used in the theoretical proofs.

Assumption 2.2.1. The utility function $U : \mathbb{R} \to \mathbb{R}$ is a continuous, increasing and concave function such that

$$\left| U(x) - U(\overline{x}) \right| \leq Q(|x|, |\overline{x}|)\, |x - \overline{x}|, \tag{2.6}$$

where $Q(|x|, |\overline{x}|)$ is a positive polynomial function.

Assumption 2.2.2. The set B has non-empty interior. The penalty function $F : B \to \mathbb{R}$ is a continuous convex function, and F attains its minimum in the interior of B.

2.3 Value functions of two-player zero-sum stochastic differential games

In order to complete the description of the game, we need to clarify what information is available to the controllers at each time s. For multi-stage discrete time games this can be formulated inductively. However, this is problematic in continuous time, because control choices can be changed instantaneously (Fleming and Soner, 2006, Chapter 11). To address this issue, Fleming and Souganidis (1989) adopted the idea of a progressive strategy in a two-player zero-sum SDG, which is defined as follows:

Definition 2.3.1. An admissible strategy Γ (resp. Δ) for the investor (resp. market) on $[t,T]$ is a mapping $\Gamma : \mathcal{B} \to \mathcal{A}$ (resp. $\Delta : \mathcal{A} \to \mathcal{B}$) such that, for any $s \in [t,T]$ and $\theta, \tilde{\theta} \in \mathcal{B}$ (resp. $\alpha, \tilde{\alpha} \in \mathcal{A}$), $\theta(u) = \tilde{\theta}(u)$ (resp. $\alpha(u) = \tilde{\alpha}(u)$) for all $u \in [t,s]$ implies $\Gamma(\theta)(u) = \Gamma(\tilde{\theta})(u)$ (resp. $\Delta(\alpha)(u) = \Delta(\tilde{\alpha})(u)$) for all $u \in [t,s]$. The set of all admissible strategies for the investor (resp. market) on $[t,T]$ is denoted by $\mathcal{N}$ (resp. $\mathcal{M}$).

In the two-player zero-sum SDG, one player is allowed to strategically adapt his control according to the control of his opponent in a non-anticipative fashion. This is in contrast to the static game, in which the player must choose his control without any knowledge of the opponent's choice. Then, we may define another set of value functions using these admissible strategies: the *upper value function* of the two-player zero-sum SDG is defined by

$$\overline{v}(t,x) = \sup_{\Gamma\in\mathcal{N}} \inf_{\theta\in\mathcal{B}} \Big\{ \mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta}) \Big] \Big\}, \tag{2.7}$$

and the corresponding *lower value function* is

$$\underline{v}(t,x) = \inf_{\Delta \in \mathcal{M}} \sup_{\alpha \in \mathcal{A}} \Big\{ \mathbb{E}^{t,x} \Big[\lambda_0 \int_t^T F(\Delta_s) ds + U(X_T^{\alpha,\Delta}) \Big] \Big\}. \tag{2.8}$$

The terms "lower" and "upper" are not obvious at first glance, one might first guess the opposite because $\inf\sup \geq \sup\inf$. We will justify $\underline{v} \leq \overline{v}$ in Corollary 2.3.2 using the comparison principle.

2.3.1 Existence of a value for the differential games

In this section, we prove that the four value functions defined in the previous sections all coincide, i.e., $\underline{u}(t,x) = \underline{v}(t,x) = \overline{v}(t,x) = \overline{u}(t,x)$. This is established via the following propositions.

Proposition 2.3.1. The four value functions defined in section 2.2 and section 2.3 satisfy the following inequalities:

$$\underline{u}(t,x) \leq \underline{v}(t,x) \leq \overline{v}(t,x) \leq \overline{u}(t,x). \tag{2.9}$$

Proof. The inequality $\underline{v}(t,x) \leq \overline{u}(t,x)$ holds because $\mathcal{M}$ contains constant mappings, i.e., $\Delta(\alpha) = \theta$ for any $\alpha \in \mathcal{A}$ and fixed $\theta \in \mathcal{B}$. Similarly, $\underline{u}(t,x) \leq \overline{v}(t,x)$ holds because $\mathcal{N}$ contains a copy of $\mathcal{A}$. Then for all $\alpha \in \mathcal{A}$ and $\epsilon > 0$, there exists some $\overline{\Delta}$ such that

$$\inf_{\Delta \in \mathcal{M}} \sup_{\alpha \in \mathcal{A}} J\left(t,x,\alpha,\Delta(\alpha)\right) + \epsilon \geq \sup_{\alpha \in \mathcal{A}} J\left(t,x,\alpha,\overline{\Delta}(\alpha)\right) \geq J\left(t,x,\alpha,\overline{\Delta}(\alpha)\right) \geq \inf_{\theta \in \mathcal{B}} J(t,x,\alpha,\theta).$$

So $\underline{u}(t,x) \leq \underline{v}(t,x)$. A similar argument gives us $\overline{v}(t,x) \leq \overline{u}(t,x)$. Hence we have

$$\underline{u} \leq \underline{v} \leq \overline{u}, \qquad \underline{u} \leq \overline{v} \leq \overline{u}.$$

In order to complete the proof, it suffices to show that $\underline{v}(t,x) \leq \overline{v}(t,x)$. This is proven in Corollary 2.3.2. □

Proposition 2.3.2. Let U be a continuous, increasing and concave utility function on $\mathbb{R}$, suppose that Assumption 2.2.2 holds, then $\underline{u}(t,x) = \underline{v}(t,x) = \overline{v}(t,x) = \overline{u}(t,x)$.

Proof. See Appendix A.1. □

Using Proposition 2.3.2, we can conclude that there exists a value for the two-player zero-sum SDG, i.e., $\underline{v} = \overline{v}$. We focus on the analysis of $\overline{v}(t,x)$ in the following sections.

2.3.2 Dynamic programming principle

If the drift and volatility functions of dynamics (2.2) and the utility function U were bounded and U was Lipschitz continuous, we could apply the results of Fleming and Souganidis (1989) directly. However, in our model, the drift and volatility functions are unbounded and U is only locally Lipschitz continuous. So we must extend the classical results and use localization techniques to prove that the value function $\overline{v}(t,x)$ defined in (2.7) satisfies the Dynamic Programming Principle (DPP). The DPP is widely used in numerical methods, such as the least squares Monte Carlo method.

Before presenting the main result, we require the following important property of the value function.

Proposition 2.3.3. Suppose that Assumption 2.2.1 holds true. Then the value function $\overline{v}(t,x)$ (2.7) is locally Lipchitz continuous w.r.t x. There exists a positive polynomial function Φ such that

$$\left|\overline{v}(t,x)-\overline{v}(t,\overline{x})\right| \leq \Phi(|x|,|\overline{x}|)\,|x-\overline{x}|, \quad \forall (t,x)\in[0,T]\times\mathbb{R}. \tag{2.10}$$

Proof. See Appendix A.2. □

We are now in the position to present a main result in this paper.

Theorem 2.3.1 (Dynamic Programming Principle)**.** Suppose that Assumption 2.2.1 and 2.2.2 hold true. Define the value function $\overline{v}(t,x)$ by (2.7) for $(t,x)\in[0,T]\times\mathbb{R}$. Let $t+h$ be a stopping time, then, for $t\leq t+h\leq T$, we have

$$\overline{v}(t,x)=\sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}}\Big\{\mathbb{E}^{t,x}\Big[\lambda_0\int_t^{t+h}F(\theta_s)ds+\overline{v}(t+h,X^{\Gamma,\theta}_{t+h})\Big]\Big\}. \tag{2.11}$$

Proof. See Appendix A.3. □

As a consequence of the DPP, the value function $\overline{v}(t,x)$ satisfies the following property.

Corollary 2.3.1. Suppose that Assumption 2.2.1, 2.2.2 hold true. Then the value function $\overline{v}(t,x)$ defined in (2.7) is Hölder continuous in t on $[0,T]$, with exponent $\frac{1}{2}$.

Proof. See Appendix A.4. □

2.3.3 Viscosity solution of the HJBI equation

In this section, we prove that the value function is the unique viscosity solution of a Hamilton–Jacobi–Bellman–Isaacs equation. First, we prove the existence of the viscosity solution, and then we state the uniqueness of this viscosity solution.

Existence of a viscosity solution of the HJBI Equation

Now we state another main result in this paper; the proof is a modification of Talay and Zheng (2002).

Theorem 2.3.2. Suppose that Assumption 2.2.1 and 2.2.2 hold true. Then the value function $\overline{v}(t,x)$ defined in (2.7) is a viscosity solution of the HJBI equation

$$\begin{cases}\frac{\partial v}{\partial t}(t,x)+H(t,x,\frac{\partial v}{\partial x}(t,x),\frac{\partial^2 v}{\partial x^2}(t,x))=0 & \text{in } [0,T)\times\mathbb{R}\\ v(T,x)=U(x) & \text{on } [T]\times\mathbb{R},\end{cases} \tag{2.12}$$

where

$$H(t,x,p,M)=\inf_{\Theta\in B}\sup_{\mathbf{a}\in A}\left\{\lambda_0F(\Theta)+(\mathbf{a}^\intercal(\mu-\mathbf{r})+r)xp+\frac{1}{2}tr\left(\mathbf{a}^\intercal\Sigma\mathbf{a}x^2M\right)\right\}, \tag{2.13}$$

for $(t,x,p,M)\in[0,T]\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}$.

Proof. See Appendix A.5. □

Comparison principle for the HJBI Equation

In this subsection, we present the comparison principle for equation (2.12), which implies the uniqueness of the viscosity solution of the HJBI equation. We can adapt the proof from Pham (2009, Theorem 4.4.4) for an HJB equation and straightforwardly extend it to HJBI equations with two controls.

Theorem 2.3.3. Comparison Principle.
Let Assumption 2.2.1, 2.2.2 hold true. Define the HJBI equation as

$$-\frac{\partial v}{\partial t}(t,x)+\lambda v(t,x)-\inf_{\Theta\in B}\sup_{\mathbf{a}\in A}\left\{\lambda_0 F(\Theta)+(\mathbf{a}^\intercal(\mu-\mathbf{r})+r)x\frac{\partial v}{\partial x}(t,x)+\frac{1}{2}tr\left(\mathbf{a}^\intercal\Sigma\mathbf{a}x^2\frac{\partial^2 v}{\partial x^2}(t,x)\right)\right\}=0,$$
$$\text{for } (t,x)\in[0,T)\times\mathbb{R}, \lambda\in\mathbb{R}. \quad (2.14)$$

Let U (resp. V) be a upper semi-continuous viscosity subsolution (resp. lower semi-continuous supersolution) with polynomial growth condition to equation (2.14). If $U(T,\cdot)\leq V(T,\cdot)$ on $\mathbb{R}$, then $U\leq V$ on $[0,T]\times\mathbb{R}$.

Proof. See Appendix A.6. □

As a consequence of the comparison principle, the function $\overline{v}(t,x)$ (2.7) is in fact the unique viscosity solution of the HJBI equation (2.12).

Corollary 2.3.2. Let Assumption 2.2.1, 2.2.2 hold true. Define the lower and upper value functions of the two-player zero-sum SDG by (2.8) and (2.7). Then

$$\underline{v}(t,x)\leq\overline{v}(t,x) \qquad \text{for } (t,x)\in[0,T]\times\mathbb{R}.$$

Proof. From Theorem 2.3.2, $\overline{v}(t,x)$ is a viscosity solution of the HJBI equation (2.12). Let $\phi\in C^\infty([0,T)\times\mathbb{R})$ be a test function such that $(t_0,x_0)\in[0,T)\times\mathbb{R}$ is a local minimum of $\overline{v}-\phi$. Using the viscosity supersolution property of $\overline{v}(t,x)$, we have

$$-\frac{\partial\phi}{\partial t}(t_0,x_0)-H(t_0,x_0,\frac{\partial\phi}{\partial x}(t_0,x_0),\frac{\partial^2\phi}{\partial x^2}(t_0,x_0))\geq 0,$$

where $H(t,x,p,M)$ is defined by (2.13). Define

$$\tilde{H}(t,x,p,M)=\sup_{\mathbf{a}\in A}\inf_{\Theta\in B}\left\{\lambda_0F(\Theta)+(\mathbf{a}^\intercal(\mu-\mathbf{r})+r)xp+\frac{1}{2}tr\left(\mathbf{a}^\intercal\Sigma\mathbf{a}x^2M\right)\right\}. \quad (2.15)$$

It is obvious that $H\geq\tilde{H}$, so

$$-\frac{\partial\phi}{\partial t}(t_0,x_0)-\tilde{H}(t_0,x_0,\frac{\partial\phi}{\partial x}(t_0,x_0),\frac{\partial^2\phi}{\partial x^2}(t_0,x_0))\geq 0 \quad \text{in } [0,T)\times\mathbb{R}.$$

Thus $\overline{v}(t,x)$ is a supersolution of the HJBI equation

$$\frac{\partial v}{\partial t}(t,x)+\tilde{H}(t,x,\frac{\partial v}{\partial x}(t,x),\frac{\partial^2 v}{\partial x^2}(t,x))=0,\ (t,x)\in[0,T)\times\mathbb{R}.$$

Using the results of Fleming and Souganidis (1989) and a similar argument, we can prove the lower value function $\underline{v}(t,x)$ (2.8) is the unique viscosity solution of the HJBI equation

$$\begin{cases}\frac{\partial v}{\partial t}(t,x)+\tilde{H}(t,x,\frac{\partial v}{\partial x}(t,x),\frac{\partial^2 v}{\partial x^2}(t,x))=0 & \text{in } [0,T)\times\mathbb{R}\\ v(T,x)=U(x) & \text{on } [T]\times\mathbb{R}.\end{cases} \quad (2.16)$$

Finally, by the comparison principle, we have $\underline{v}(t,x)\leq\overline{v}(t,x)$, as required. □

2.4 Logarithmic utility case

In this section, we consider a particular utility function $U(x) = \ln(x)$. With this specifically chosen utility function, we can write X_T explicitly and optimal controls can be solved easily. The value function reads:

$$\begin{aligned}\overline{v}(t,x) &= \sup_{\alpha\in\mathcal{A}} \inf_{(\mu(\cdot),\Sigma(\cdot))\in\mathcal{B}} \Bigg\{ \mathbb{E}^{t,x}\Big[\ln(x) + \int_t^T (\alpha_s^\intercal \mu_s + r - \alpha_s^\intercal \mathbf{r} - \frac{1}{2}\alpha_s^\intercal \Sigma_s \alpha_s) ds \\ &\quad + \int_t^T \alpha_s \sigma_s dW_s + \int_t^T \lambda_0 F(\mu_s, \sigma_s) ds\Big]\Bigg\} \\ &= \ln(x) + \sup_{\alpha\in\mathcal{A}} \inf_{(\mu(\cdot),\Sigma(\cdot))\in\mathcal{B}} \Bigg\{ \mathbb{E}^{t,x}\Big[\int_t^T (\alpha_s^\intercal \mu_s + r - \alpha_s^\intercal \mathbf{r} - \frac{1}{2}\alpha_s^\intercal \Sigma_s \alpha_s) ds \\ &\quad + \int_t^T \lambda_0 F(\mu_s, \sigma_s) ds\Big]\Bigg\}.\end{aligned} \tag{2.17}$$

We can solve for the optimal variables with the optimality conditions:

$$\mu_s - \mathbf{r} - \Sigma_s \alpha_s = 0, \tag{2.18}$$

$$\alpha_s + \lambda_0 \frac{\partial F(\mu_s, \sigma_s)}{\partial \mu_s} = 0, \tag{2.19}$$

$$-\frac{1}{2}\alpha_s \alpha_s^\intercal + \lambda_0 \frac{\partial F(\mu_s, \sigma_s)}{\partial \sigma_s} = 0. \tag{2.20}$$

But we emphasize that logarithmic function is just one of the utility functions we consider in the chapter. When using other more general utility functions, the problem can not be solved as simple as this.

As a one-dimensional toy example, we let μ to be a known constant and $F(\mu_s, \sigma_s) = \lambda_0(\sigma_s - \sigma_0)^2$. Then we obtain the following optimality conditions:

$$\hat{\alpha}_s = \frac{\mu - r}{\hat{\sigma}_s^2}, \tag{2.21}$$

$$-\frac{1}{2}\hat{\alpha}_s^2 + \lambda_0(1 - \frac{\sigma_0}{\hat{\sigma}_s}) = 0, \tag{2.22}$$

which leads to a quartic equation

$$0 = \hat{\sigma}_s^4 - \sigma_0 \hat{\sigma}_s^3 - \frac{(\mu - r)^2}{2\lambda_0}. \tag{2.23}$$

The optimal $\hat{\sigma}_s$ and $\hat{\alpha}_s$ can be solved from equation (2.23) explicitly. Equation (2.23) always has a real positive root and we provide the solution in the Appendix A.7. By substituting the optimal controls into (2.17), we obtain the analytical solution of the value function.

From equations (2.21)–(2.22), we observe that the optimal volatility and investment strategy are both constants, being independent of the wealth X_s and the time s. The classical optimal portfolio strategy given by Merton is also a constant, where $\alpha^* = \frac{\mu - r}{\sigma^2(1-\gamma)}$[1] for CRRA utility functions. However, in our problem, it is not possible to find an analytical solution for a power utility function. We will use numerical methods to estimate the values

[1] $1 - \gamma$ is the relative risk aversion constant.

in the next subsection. It is worth mentioning that the reference volatility σ_0 is not necessarily a constant, it can be a local volatility depending on time and stock price. However, for multiple assets, it would increase the dimension of the problem.

We use λ_0 to govern the degree to robustness, and we illustrate how the value of $\hat{\sigma}_s^2$ varies with λ_0 in Figure 2.1. As shown in the figure, by increasing λ_0, the optimal $\hat{\sigma}_s^2$ approaches to σ_0^2. Figure 2.2 compares a robust value function $\sup_\alpha \inf_\sigma \mathbb{E}\left[U(X_T)\right]$ with Merton's non-robust value function $\sup_\alpha \mathbb{E}\left[U(X_T)\right]$. We use the optimal controls $\hat{\alpha}, \hat{\sigma}$ calculated from (2.23) for the robust value, and an estimated constant volatility σ_0 for Merton's value. It is not surprising to see that the robust value is always lower than Merton's value, since we sacrifice some performance for extra robustness. Figure 2.2(a) shows that as the estimated σ_0 increases, the robust value approaches to the non-robust value, because $\hat{\sigma}_t \to \sigma_0$ as $\sigma_0 \to +\infty$. For a similar reason, in Figure 2.2(b), as the penalization gets stronger, the robust value asymptotically approaches Merton's value. When $\lambda_0 \to 0$, the worst volatility $\hat{\sigma}_t \to +\infty$, which means it is very risky to invest in stocks and it is wiser to save money in the saving account. As $\hat{\alpha}_t \to 0$, $U(X_T) \to U(xe^{rT})$. We set $x = 1, r = 0.015, T = 1$, then this value goes to 0.015 as shown at the lower left end of Figure 2.2(b).

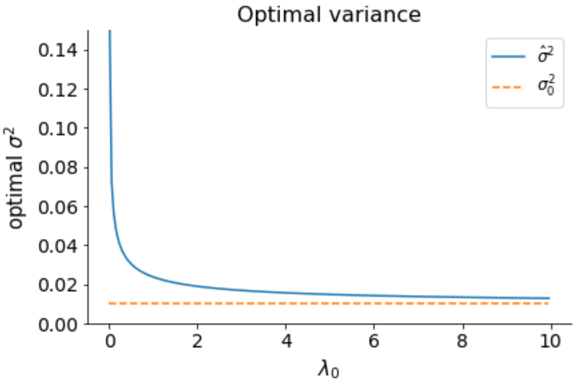
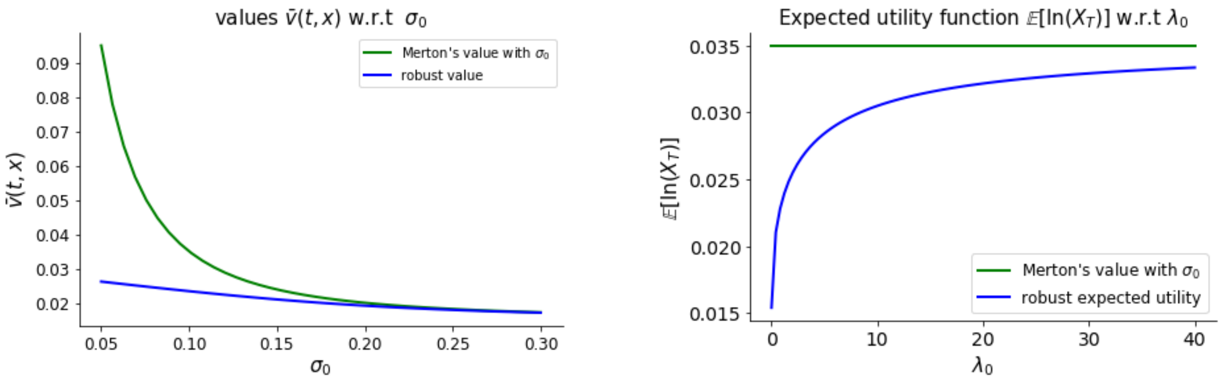


Figure 2.1: Optimal variance w.r.t. λ_0

Hereafter, we assume the drift $\mu(\cdot)$ is a known constant for simplicity, and the uncertainty about the market only comes from the covariance process. With some modifications in the implementation, all the algorithms proposed in the subsequent sections can be applied to the case where both $\mu(\cdot)$ and $\Sigma(\cdot)$ are uncertain.



(a)

(b)

Figure 2.2: Compare robust value with Merton's value, $U = \log(x), F = (\sigma_t - \sigma_0)^2$

2.5 Comparison of robust and non-robust portfolios

2.5.1 Monte Carlo simulation

In this section, we implement our robust strategy using Monte Carlo simulations, and compare the performance of robust and non-robust portfolios.

As we know, in the real world volatility estimates are noisy and biased, though likely to oscillate around a reference value in the long run. In the first experiment, we have a reference covariance matrix Σ_0, which is estimated according to historical data. We assume that the real-world covariance is *the reference covariance* Σ_0 *plus some noise*. We construct robust and non-robust portfolios consisting of two risky assets and one risk-free asset. For the robust portfolio, we use $\lambda_0 F(\Sigma_s) = \lambda_0 \|\Sigma_s - \Sigma_0\|_2^2$ ($\|\cdot\|_2$ denotes the usual Frobenius norm) as the penalty function, then the analytical robust investment strategy $(\hat{\alpha}_s^1, \hat{\alpha}_s^2)$ can be calculated in a similar method to the one in section 2.4. For the non-robust one, we use Σ_0 as the covariance, then calculate the non-robust strategy (α_s^1, α_s^2) accordingly. Assuming the real covariance matrix during the investment process is $\Sigma_{\text{real}} = \Sigma_0 + \varepsilon \times \text{noise}$, where the noise follows a standard normal distribution $\mathcal{N}(0,1)$[2] and ε is the magnitude of the noise, we use Monte Carlo simulations to estimate the expected utility function

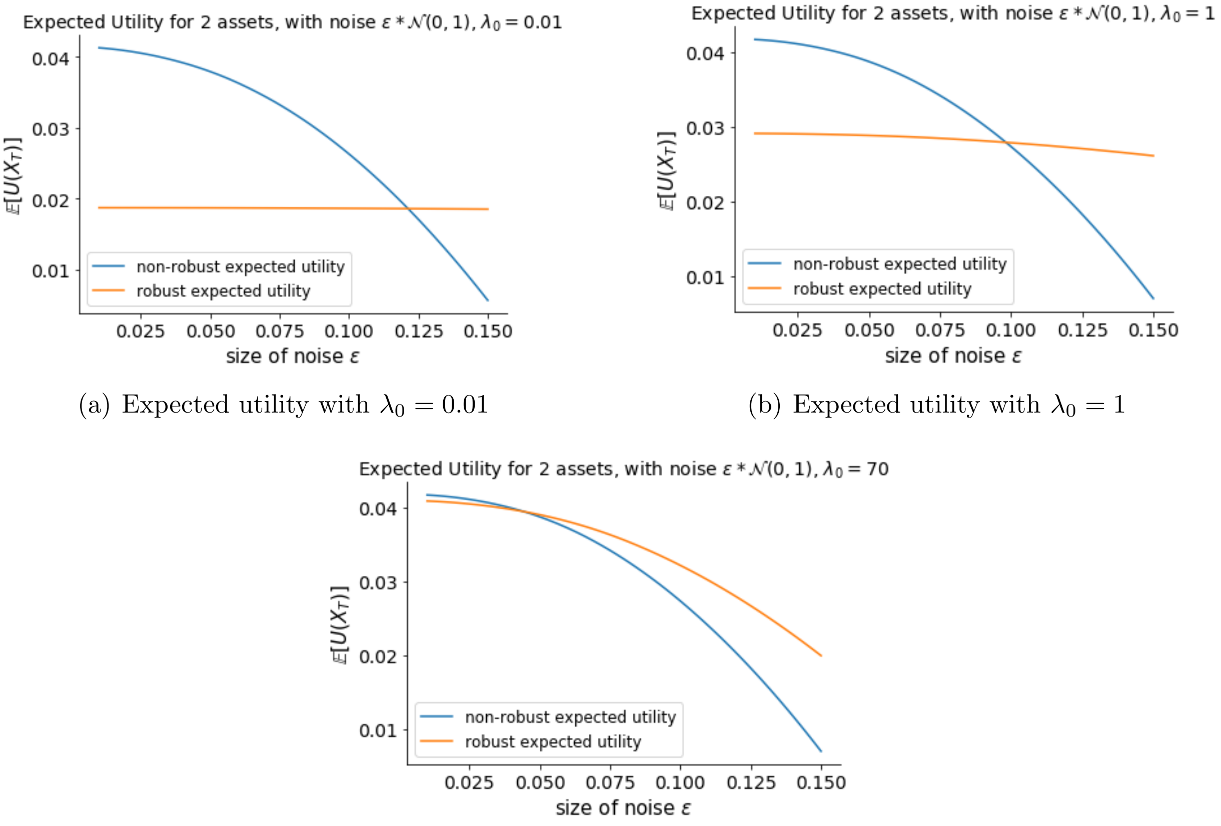
$$\mathbb{E}\Big[\ln(X_T)\Big] = \mathbb{E}^{t,x}\Big[\ln(x) + \int_t^T \alpha_s^{\intercal}(\mu - \mathbf{r}) + r - \frac{1}{2}\alpha_s^{\intercal}\Sigma_{\text{real}}\alpha_s ds\Big]. \tag{2.24}$$

We substitute $\alpha_s = (\hat{\alpha}_s^1, \hat{\alpha}_s^2)$ in (2.24) for the robust portfolio, and $\alpha_s = (\alpha_s^1, \alpha_s^2)$ for the non-robust one.

The results with various λ_0 are shown in Figure 2.3(a) to 2.3(c), where we used 2×10^5 paths in the simulation and the initial wealth $X_0 = 1$. We can observe that the robust portfolio may underperform when there is little noise. But, as the noise size ε increases, the robust strategy will outperform the non-robust strategy eventually. Comparing Figures 2.3(a), 2.3(b) and 2.3(c), we can find that when the penalty is relatively weak ($\lambda_0 = 0.01$), it takes a bigger noise size for the robust strategy to outperform. When the penalty is stiff ($\lambda_0 = 70$), the robust strategy will outperform with a very small noise size. The robust expected utility is almost a constant for all sizes of noise in Figure 2.3(a), meaning that our model is very robust to changes in market circumstances. Among the three values of λ_0 illustrated, Figure 2.3(c) is probably the most attractive to investors. When the reference Σ_0 is perfect, the robust portfolio only loses to the non-robust one by a little, but when Σ_0 is wrong, the robust portfolio outperforms the non-robust one by a large amount. It means the price we pay for the robustness is tolerable, but the potential reward is substantial.

Define the *crossing point* ε as the value of ε for which the robust expected utility matches the non-robust expected utility. Figure 2.4(a) depicts how the crossing point ε varies with respect to λ_0. It tells us how much should our reference covariance be wrong for the robust portfolio to outperform the non-robust portfolio. The behaviour of the robust portfolio varies with λ_0. For a certain ε, by looping over a range of λ_0, we can find the one giving us the maximal robust expected utility. This relation is plotted in Figure 2.4(b). With this plot, if we know how confident we are with the reference Σ_0 (i.e., the value of ε), we can choose the best λ_0 for robust portfolio allocation.

[2] This noise is added to each component of Σ_0, i.e., adding noise to the standard deviation of asset 1 (σ_1), asset 2 (σ_2), and their correlation ρ_{12}.

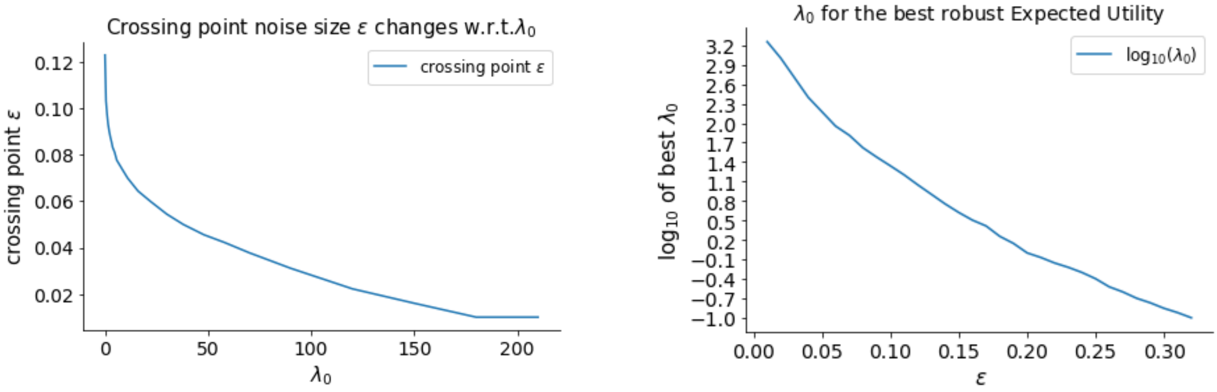


(a) Expected utility with $\lambda_0 = 0.01$

(b) Expected utility with $\lambda_0 = 1$

(c) Expected utility with $\lambda_0 = 70$

Figure 2.3



(a)

(b)

Figure 2.4

2.5.2 Empirical market data

In the second experiment, we implement the robust and non-robust strategies with empirical market data. We have 1007 portfolios, and we construct each portfolio according to robust and non-robust allocations, respectively. Each portfolio consists of 2 risky assets and 1 risk-free asset, with a maturity of $T = 1$ year. The portfolios' starting dates range from 02/04/15 to 03/04/19 (for example, the 1st portfolio starts on 02/04/15 and lasts for one year, the 1007th portfolio starts on 03/04/19 and lasts for one year as well). We choose the S&P500 (^GSPC) and SPDR Gold Shares (GLD)[3] as our risky assets and use a constant interest rate $r = 0.015$. For a specific portfolio, we set Σ_0 to be the sample covariance estimator of the 5 years of daily relative returns before the starting date. The estimated annual expected returns μ_1, μ_2 are the exponentially weighted moving average of the daily relative returns with a 5-year lookback window and 2.75-year half-life. With a decay parameter $\beta = 0.999$, for the nth portfolio, $\mu_{i,i=1,2} = 252 \times \frac{1}{1-\beta^{1260}} \sum_{t=0}^{1260} (1-\beta)\beta^t \frac{S^i_{n-t} - S^i_{n-t-1}}{S^i_{n-t-1}}$.

In this experiment, we use a logarithmic utility function and a penalty function $\lambda_0 F(\Sigma_s) = \lambda_0 \big\| \Sigma_s - \Sigma_0 \big\|_2^2$. At the beginning of the investment process for each portfolio, we estimate parameters μ_1, μ_2, Σ_0 and then compute the robust and non-robust portfolio allocations accordingly. Starting from an initial wealth $X_0 = 1$, the wealth of the non-robust portfolio evolves as

$$\begin{aligned} X_{n+1} = X_n \exp\Big\{ &\alpha^1_n \frac{S^1_{n+1} - S^1_n}{S^1_n} + \alpha^2_n \frac{S^2_{n+1} - S^2_n}{S^2_n} + (1 - \alpha^1_n - \alpha^2_n) r \Delta t \\ &- \frac{1}{2} \left[\alpha^1_n \left(\frac{S^1_{n+1} - S^1_n}{S^1_n} - \mu_1 \Delta t \right) + \alpha^2_n \left(\frac{S^2_{n+1} - S^2_n}{S^2_n} - \mu_2 \Delta t \right) \right]^2 \Big\}, \; n \in [0, 251], \end{aligned} \tag{2.25}$$

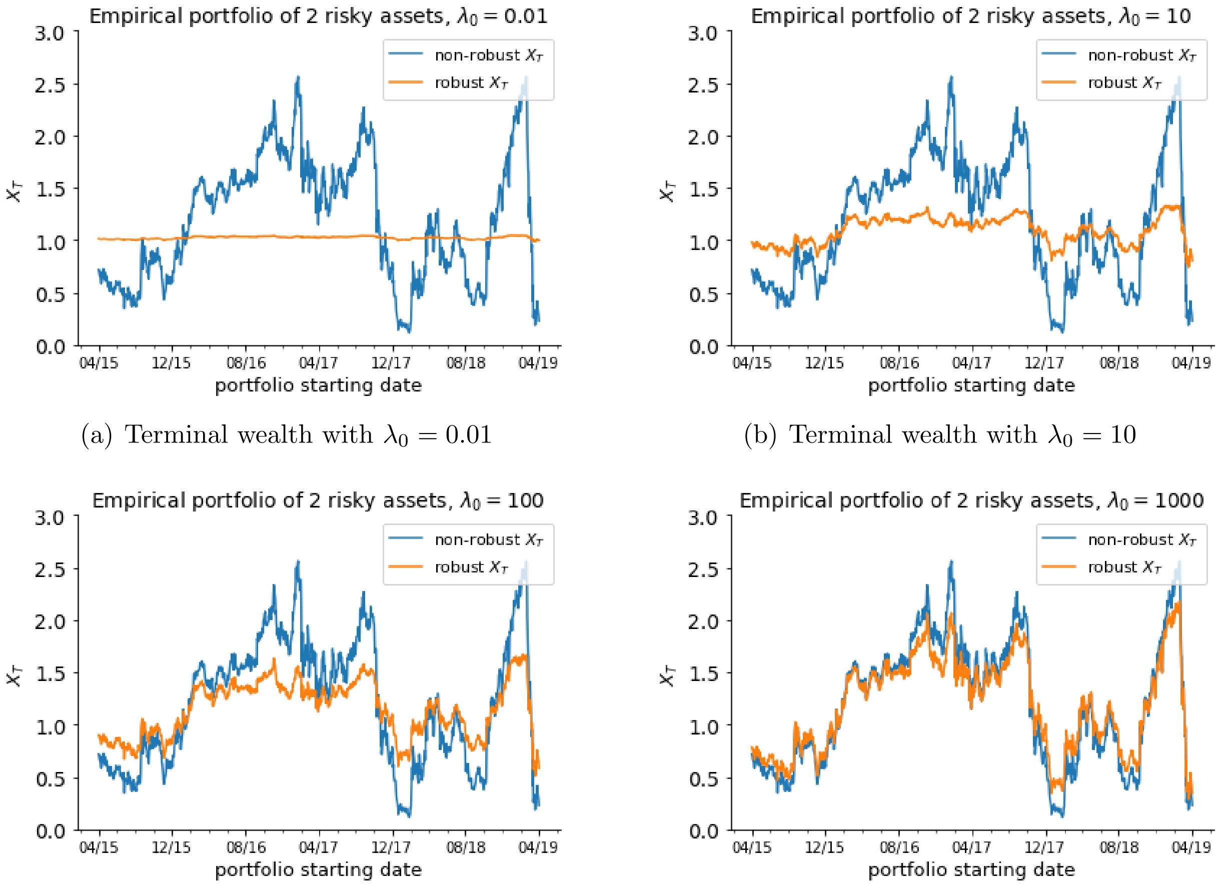
where (α^1_n, α^2_n) are the non-robust allocations on day n. For the wealth of the robust portfolio, just replace (α^1_n, α^2_n) with the robust allocations $(\hat{\alpha}^1_n, \hat{\alpha}^2_n)$ in (2.25). Finally, by averaging the $\ln(X_T)$ of all the portfolios, we get the expected utility function.

Figures 2.5(a)–2.5(d) present the terminal wealth X_T of the 1007 robust and non-robust portfolios. For a small λ_0, the robust portfolios are very stable. No matter how the market changes, the robust terminal wealth stays around 1. As λ_0 increases, the robust portfolios start to show fluctuations. Eventually, their behaviour converges to that of the non-robust portfolios as λ_0 approaches to infinity, which corresponds to the non robust case. This behaviour is consistent with our expectations. The penalty function is not playing its role when λ_0 is close to zero. Hence the robust allocations are optimal for the most chaotic market situations, and the investment strategies are very conservative. As λ_0 becomes larger, the penalty function comes into play and prevents extreme volatilities. As a consequence, the robust strategies are less conservative, and portfolios will show more fluctuations under regime changes.

We show the robust and non-robust expected utilities in Figure 2.6. It depicts how $\mathbb{E}[\ln(X_T^{\alpha^1,\alpha^2})]$ and $\mathbb{E}[\ln(X_T^{\hat{\alpha}^1,\hat{\alpha}^2})]$ change w.r.t. λ_0. We can compare this plot with figures 2.3(a), 2.3(b), 2.3(c) and 2.4(b) in section 2.5.1. For a given amount of noise, the robust portfolio may underperform for small λ_0, but the value will increase gradually and reach a highest point. Finally, the robust expected utility will converge to the non-robust one.

To illustrate the time evolution of the portfolio wealth, we show the stock prices and wealth of two portfolios, starting on 2017-01-03 (Figure 2.7) and 2018-01-26 (Figure 2.8), respectively. For the portfolio in Figure 2.7, the optimal non-robust allocations are $\alpha^1 = 5.778, \alpha^2 = -2.174$, and the robust allocations with $\lambda_0 = 200$ are $\hat{\alpha}^1 = 3.083, \hat{\alpha}^2 = -1.452$.

[3] Stock price data are available on https://au.finance.yahoo.com.



(a) Terminal wealth with $\lambda_0 = 0.01$

(b) Terminal wealth with $\lambda_0 = 10$

(c) Terminal wealth with $\lambda_0 = 100$

(d) Terminal wealth with $\lambda_0 = 1000$

Figure 2.5

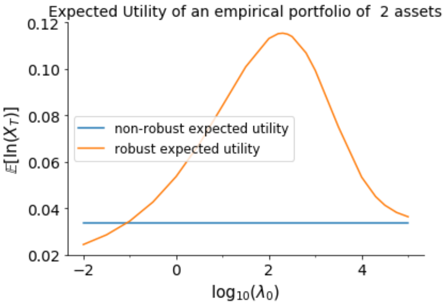
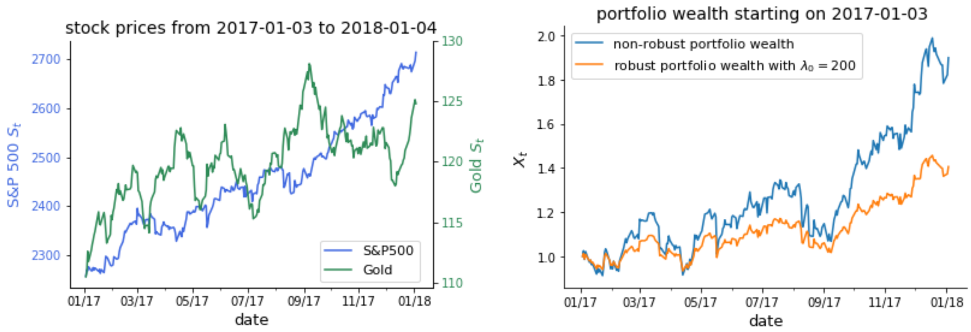


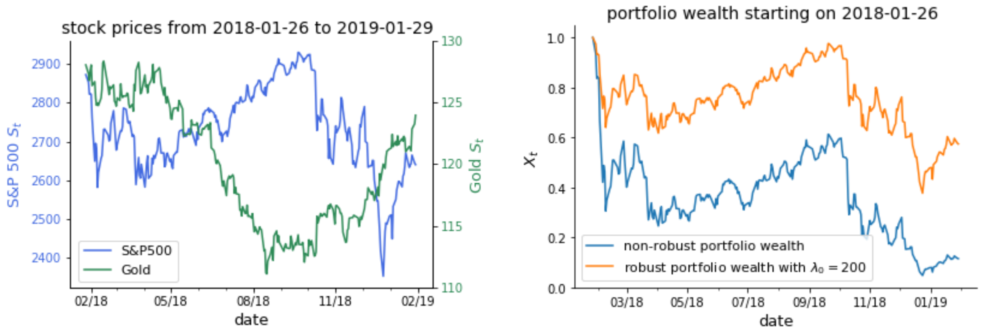
Figure 2.6: Empirical expected utility w.r.t. λ_0

The allocations are both constant, independent of time. The S&P500 keeps rising in Figure 2.7(a), while there are some fluctuations in the Gold price. Over the same period, the absolute performance of the non-robust portfolio is better all the way (Figure 2.7(b)). For the portfolio in Figure 2.8, we have $\alpha^1 = 9.418, \alpha^2 = 0.301$, and $\hat{\alpha}^1 = 3.940, \hat{\alpha}^2 = -0.054$. Since the proportions invested in Gold are small for both robust and non-robust portfolios, the trend of wealth is dominated by the price of S&P500. There are two big drops happening in Feb. 2018 and Dec. 2018, respectively. These are also reflected in the portfolio wealth in Figure 2.8(b). However, compared with the non-robust strategy, the robust strategy is more conservative. Hence, the robust portfolio loses less during the market shocks and outperforms the non-robust one.



(a) (b)

Figure 2.7: The portfolio starting on 2017-01-03



(a) (b)

Figure 2.8: The portfolio starting on 2018-01-26

From the above empirical experiments and the Monte Carlo simulations from section 2.5.1 , we can see that, by adding this robust mechanism with a properly chosen λ_0, the portfolio value can overcome a wrong covariance matrix estimate and is less vulnerable to sudden market shocks. Furthermore, this parameter λ_0 protect us against the objection that the max-min expected utility theory is too cautious because, by acting as if the agent believed in the worst-case model, he puts too much weight on "very unlikely" scenarios (Hansen and Sargent, 2008).

2.6 Numerical results

In this section, we provide a few numerical algorithms to approximate the value functions. In particular, we implemented an implicit finite difference method, a Generative Adversarial Network method, and a control randomization method.

2.6.1 Implicit finite difference method

In this section, we are computing the value function via an implicit finite difference method. We use the penalty function $\lambda_0 F(\sigma_t^2) = \lambda_0(\sigma_t^2)^2$ for simplicity. Then the HJBI equation is

$$\overline{v}_t + H(t, x, \overline{v}_x, \overline{v}_{xx}) = 0, \tag{2.26}$$

where the Hamiltonian is defined by

$$H(t, x, \overline{v}_x, \overline{v}_{xx}) = \inf_{\sigma^2} \sup_{\mathbf{a}} \Big\{ \mathbf{a}(\mu - r)x\overline{v}_x + rx\overline{v}_x + \frac{1}{2}\mathbf{a}^2\sigma^2 x^2 \overline{v}_{xx} + \lambda_0(\sigma^2)^2 \Big\}. \tag{2.27}$$

Solving for the optimal controls in (2.27) using the first order condition, we obtain $\hat{\mathbf{a}} = -\frac{(\mu-r)x\overline{v}_x}{\sigma^2 x^2 \overline{v}_{xx}}$ and $\hat{\sigma}^2 = \left(-\frac{(\mu-r)^2\overline{v}_x^2}{4\lambda_0\overline{v}_{xx}}\right)^{1/3}$. Substituting $\hat{\mathbf{a}}$ and $\hat{\sigma}^2$ into the PDE (2.26), we obtain

$$\overline{v}_t + C\overline{v}_x^{\frac{4}{3}}(-\overline{v}_{xx})^{-\frac{2}{3}} + rx\overline{v}_x = 0,$$

where $C = (3 \times 2^{-\frac{4}{3}})\lambda_0^{\frac{1}{3}}(\mu - r)^{\frac{4}{3}}$. Note we have shown in section 2.3.1 that $\overline{v}_{xx} < 0$.

Since the PDE (2.26) is non-linear, in order to use the implicit finite difference method, we first linearize the function H with respect to the second order term via the Legendre transform. This method was also used by Jonsson and Sircar (2002a,b) to solve nonlinear HJB equations. We combine the linearization step with a fixed-point iteration scheme.

Define H^* as the Legendre transform of H with respect to the second order term; it is given by

$$H^*(a) = -C_2 a^{\frac{2}{5}}\overline{v}_x^{\frac{4}{5}} - rx\overline{v}_x,$$

where $C_2 = \frac{5}{3}(\frac{2}{3})^{-\frac{2}{5}}C^{\frac{3}{5}}$. Hence, we can represent $H(\overline{v}_{xx})$ as the supremum of a linear function in $\overline{v}_{xx}$,

$$H(\overline{v}_{xx}) = \sup_a \Big\{ a \cdot \overline{v}_{xx} - H^*(a) \Big\}. \tag{2.28}$$

It is difficult to check the condition for stability in our PDE as the optimal a is unknown. Fortunately, implicit finite difference methods have a weaker requirement for stability than explicit finite difference methods.

Algorithm 1 Implicit Finite Difference Scheme

```
for step $n = N : 1$ do
    (1) Solve $\mathbf{B}K^{n-1} + G^{n-1} = F^n$ using $\hat{a}_0(i) = g(\overline{v}^n_{i+1}, \overline{v}^n_i, \overline{v}^n_{i-1})$, and get the value vector $K^{n-1}_0$
    (2) Solve $\mathbf{B}K^{n-1} + G^{n-1} = F^n$ using $\hat{a}_1(i) = g(\overline{v}^{n-1}_{i+1}, \overline{v}^{n-1}_i, \overline{v}^{n-1}_{i-1})$, where the values $\overline{v}^{n-1}$ are from $K^{n-1}_0$. Then get the value vector $K^{n-1}_1$.
    (3) Repeat step 2 until $\left\|\hat{a}_j - \hat{a}_{j-1}\right\|_2 \leq$ tolerance
    (4) Let $K^{n-1} = K^{n-1}_j$
end
```

We set the time grid as $0, 1, ..., n, n+1, ..., N$, and the spatial grid as $1, 2, ...i, i+1, ...M$. With the maturity $T = 1$, we use a constant time step $\Delta t = \frac{T}{N}$ and a constant spatial step Δx. We apply a forward approximation for $\overline{v}_t$, a central approximation for $\overline{v}_x$, and a standard approximation for $\overline{v}_{xx}$. Working backward in the implicit scheme, at each time step n, the optimal $\hat{a}$ in (2.28) is the solution of the first order condition $\overline{v}^n_{xx} + C_2(\overline{v}^n_x)^{\frac{4}{5}}\frac{2}{5}\hat{a}^{-\frac{3}{5}} = 0$, or equivalently,

$$\hat{a} = \frac{2}{3}C(\overline{v}^n_x)^{\frac{4}{3}}(-\overline{v}^n_{xx})^{-\frac{5}{3}} =: f(\hat{a}). \tag{2.29}$$

Although we do not have the true values for $\overline{v}^n$ as the values of $\overline{v}^n$ depend on $\hat{a}$, we can use a fixed-point iteration scheme to find the solution of equation (2.29). First we make an initial guess $\hat{a}_0$ using the known values $\overline{v}^{n+1}$, which will be represented by $g(\overline{v}^n_{i+1}, \overline{v}^n_i, \overline{v}^n_{i-1})$, then iteratively generate a sequence $\hat{a}_{k,k=1,2,...}$ with $\hat{a}_k = f(\hat{a}_{k-1})$ until $\hat{a}_k$ converges.

Finally we can substitute the discrete approximations of the derivatives into the HJBI equation (2.26), and we obtain the implicit form:

$$\begin{aligned}\left(\frac{\hat{a}(i)\Delta t}{\Delta x^2} - \frac{r\,(i\Delta x + x_0)\,\Delta t}{2\Delta x}\right)\overline{v}^n_{i-1} + \left(-1 - \frac{2\hat{a}(i)\Delta t}{\Delta x^2}\right)\overline{v}^n_i + \left(\frac{\hat{a}(i)\Delta t}{\Delta x^2} + \frac{r\,(i\Delta x + x_0)\,\Delta t}{2\Delta x}\right)\overline{v}^n_{i+1} \\ = -\overline{v}^{n+1}_i - C_2\hat{a}(i)^{\frac{2}{5}}\left(\frac{\overline{v}^{n+1}_{i+1} - \overline{v}^{n+1}_{i-1}}{2\Delta x}\right)^{\frac{4}{5}}\Delta t. \end{aligned} \tag{2.30}$$

Let $\mathbf{B}$ be the coefficient matrix, K^n the value vector at time n, G^n is a vector with the make-up items, and F^{n+1} the right hand side of (2.30). Then equation (2.30) can be written in a matrix notation:

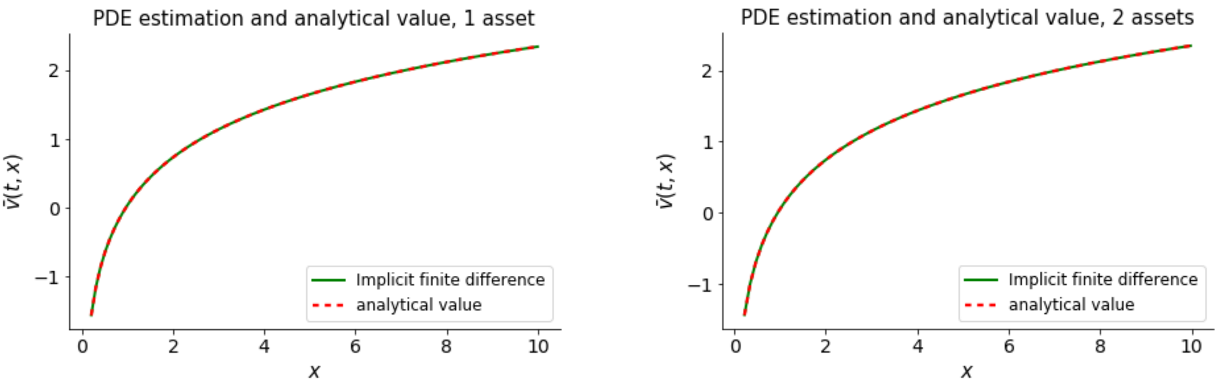
$$\mathbf{B}K^n + G^n = F^{n+1}, \qquad n = N-1, ..., 1, 0.$$

The algorithm for this method is summarized in Algorithm 1.

Logarithmic utility function

In the 1-asset example, we use the logarithmic utility function and the penalty function $\lambda_0 F(\sigma^2_t) = \lambda_0(\sigma^2_t)^2$. We let $N = 50, M = 100$. The terminal condition is given by the utility function,

$$\overline{v}(t_N, x_i) = U(x_i) \quad \forall i \in [1, M].$$



(a) $U(X_T) = \ln(X_T)$, $S_t \in \mathbb{R}^1$

(b) $U(X_T) = \ln(X_T)$, $S_t \in \mathbb{R}^2$

Figure 2.9

The boundary conditions $\overline{v}(t_n, x_1)$ and $\overline{v}(t_n, x_M)$ for $n \in [0, N-1]$ are given explicitly by the equation

$$\overline{v}(t_n, x) = \ln(x) + \sup_{\alpha} \inf_{\sigma^2} \left\{ \sum_{s=n}^{N-1} \left(\alpha_s(\mu - r) + r - \frac{1}{2}\alpha_s^2 \sigma_s^2 + \lambda_0 (\sigma_s^2)^2 \right) \Delta t \right\}.$$

Similarly, we can also implement the above method on a 2-asset example where $S_t \in \mathbb{R}^2$ and $\lambda_0 F(\Sigma_t) = \lambda_0 \left\| \Sigma_t \right\|_2^2$. The HJBI equation becomes

$$\overline{v}_t + \inf_{\sigma_1, \sigma_2, \rho} \sup_{\alpha_1, \alpha_2} \Big\{ (\alpha_1(\mu_1 - r) + \alpha_2(\mu_2 - r) + r)\, x \overline{v}_x + \frac{1}{2}(\alpha_1^2 \sigma_1^2 + 2\alpha_1 \alpha_2 \sigma_1 \sigma_2 \rho + \alpha_2^2 \sigma_2^2) x^2 \overline{v}_{xx} + \lambda_0 (\sigma_1^4 + 2\sigma_1^2 \sigma_2^2 \rho^2 + \sigma_2^4) \Big\} = 0. \quad (2.31)$$

We can solve for the optimal controls $\hat{\alpha}_1, \hat{\alpha}_2, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\rho}$ in (2.31) using the first order condition. In this example, we always have the optimal $\hat{\sigma}_1, \hat{\sigma}_2 > 0$ and $\hat{\rho} \in [-1, 1]$. Then, by applying Algorithm 1, we can get the value function of a portfolio with 2 risky assets.

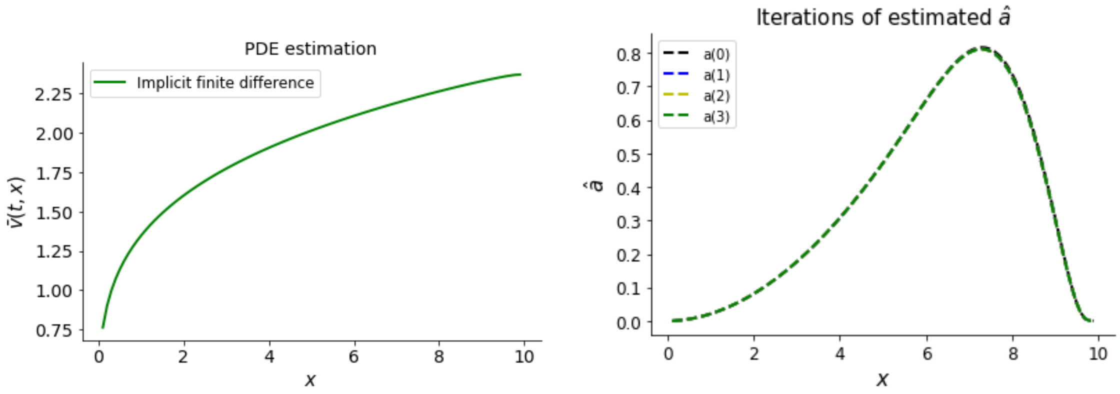
Figure 2.9(a) shows the PDE estimated $\overline{v}(t, x)$ for the 1-asset example with parameters $r = 0.015, \mu = 0.035, \lambda_0 = 10$; Figure 2.9(b) shows result for the 2-asset case with parameters $r = 0.015, \mu_1 = 0.035, \mu_2 = 0.045, \lambda_0 = 10$. Comparing with the analytical solution, we can see that the two curves completely overlap for both 1-asset and 2-asset cases, which validates the accuracy of the PDE approach.

Power utility function

In the second example, we use a power utility function. This time, we only have the terminal condition and the boundary condition for $x_1 = 0$, but not the boundary condition for a large x_M. For functions x^γ where $\gamma \in (0, 1)$, the limit of the first order derivative approaches 0 as x goes to infinity. Therefore we can use a zero Neumann boundary condition when x_M is large. Then we have the following terminal and boundary conditions:

$$\overline{v}(t_N, x_i) = U(x_i)\, \forall i \in [1, M], \quad \overline{v}(t_n, x_1) = 0\, \forall x_1 = 0, n \in [0, N-1], \quad \frac{\partial \overline{v}}{\partial x}(t_n, x_M) = 0\, \forall n \in [0, N-1].$$

Figure 2.10(a) shows the simulated value $\overline{v}(t, x)$ for a range of x, with $U(X_T) = \frac{4}{3} X_T^{\frac{1}{4}}$ and parameters $\mu = 0.035, r = 0.015, \lambda_0 = 10$. We only display the estimated curve computed



(a) estimated value function

(b) estimated $\hat{a}$ in each iteration

Figure 2.10: Finite Difference Method for $U(X_T) = \frac{4}{3}X_T^{\frac{1}{4}}$

by our PDE method, as there is no analytical solution available for comparison in this example. Figure 2.10(b) shows the first four iterations of the estimated $\hat{a}$ from an initial guess. There is almost no difference between the four curves, indicating that the fixed point iteration scheme has converged within the first four iterations.

This subsection has shown that the PDE method converges to the true value efficiently. Nevertheless, there are a few shortcomings to this approach:

- The PDE approach requires tedious algebraic manipulation before implementation. In particular, even when using the same utility function, the preliminary computations have to be redone if we switch to a different penalty function.

- In general, PDE approaches suffer from the curse of dimensionality. As the dimension of the problem becomes higher, the computational complexity increases exponentially and the approach becomes infeasible. Although the PDE approach suffices for our current problem as the wealth process is only one-dimensional, it may not be feasible for other problems arising from multidimensional stochastic differential games.

For these two reasons, in the next subsection we develop numerical schemes based on GANs and Monte Carlo simulations, which can be potentially useful for high-dimensional problems or in the case of complex penalty functions.

2.6.2 Generative Adversarial Networks

In this section, we devise a GAN-based algorithm to solve the two-player zero-sum differential game.

Generative Adversarial Networks were introduced in Goodfellow et al. (2014). A GAN is a combination of two competing (deep) neural networks: a generator and a discriminator. The generator network tries to generate data that looks similar to the training data, and the discriminator network tries to tell the real data from the fake data. The idea behind GANs is very similar to the robust optimization problem studied in our paper: GANs can be interpreted as minimax games between the generator and the discriminator, whereas our problem is a minimax game between the agent who controls the portfolio allocation and the market who controls the covariance matrix. Inspired by this connection, we propose the following GAN-based algorithm.

Our GANs are composed of two neural networks; one generates α (α-generator), the other generates σ (σ-generator). The two networks have conflicting goals, the α-generator tries to maximize the expected utility, while the σ-generator wants to minimize the expected utility. They compete against each other during the training. Because we have two networks with different objectives, it cannot be trained as a regular neural network. Each training iteration is divided into two phases: In the first phase, we train the α-generator, with the loss function $L_1 = -\mathbb{E}\left[U(X_T) + \lambda_0 \int_t^T F(\sigma_s^2)ds\right]$. Then the back-propagation only optimizes the weights of the α-generator. In the second phase, given the output α from the α-generator, we train the σ-generator with a loss function $L_2 = \mathbb{E}\left[U(X_T) + \lambda_0 \int_t^T F(\sigma_s^2)ds\right]$. During this phase, the weights of the α-generator are frozen and the back-propagation only updates the weights of the σ-generator. In a zero-sum game, the α-generator and σ-generator constantly try to outsmart each other. As training advances, the game may end up at a Nash Equilibrium.

A demonstration of the simplified network architecture is illustrated in Figure 2.11. The blue part on the left of Figure 2.11 is the α-generator. For each time step n, we construct a network $(\mathcal{A}_n)$, with the input X_n and parameter σ_n, the network generates output α_n. With the dynamics of wealth (2.32), we can continue this process until we get the terminal wealth X_N. Once we get the output $\{\alpha_n\}_{n\in[1,N]}$, we can use them as parameters for the σ-generator (the green part in the figure). In the σ-generator, similarly, we have one network $(\mathcal{S}_n)$ for each time step n. With the input X_n and parameter α_n, we can generate σ_n. At the end of this phrase, the sequence $\{\sigma_n\}_{n\in[1,N]}$ will be fed into the α-generator as parameters as well. We have summarized this training process for 1-asset examples in Algorithm 2.

In the implementation, we choose the parameters $T = 1, r = 0.015, \mu = 0.035$. The training data has a sample size $M = 200,000$. We discretize the investment process into $N = 65$ time steps. The deep neural network for each time step contains 4 hidden layers, using Leaky ReLU as the activation function. For the σ generator, to ensure the positivity of the output, we use Leaky Sigmoid as the activation function of the output layer. It is defined as $\text{LeakySigmoid}_\beta(z) = \frac{1}{1+e^{-x}}\mathbb{1}(x \le \beta) + \left[\frac{e^{-\beta}}{(1+e^{-\beta})^2} \times (x - \beta) + \frac{1}{1+e^{-\beta}}\right]\mathbb{1}(x > \beta)$. Its shape is similar to Sigmoid, but its range is $[0, +\infty]$. We train the first 100 epochs with a learning rate 5×10^{-4}, and then we train another 50 epochs with a decreased learning rate 1×10^{-4}.

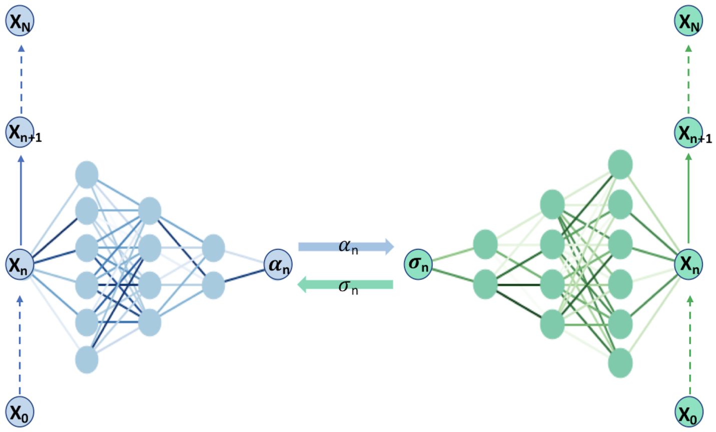


Figure 2.11: A demonstration of the adversarial networks

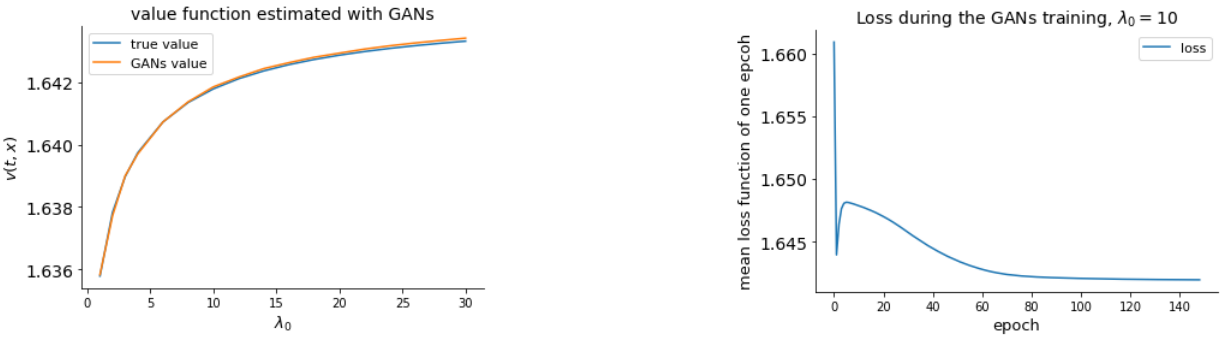
We now assess the quality of Algorithm 2. Firstly, we use a utility function $U(X_T) = \ln(X_T)$ and a cost function $\lambda_0 F(\sigma_t^2) = \lambda_0(\sigma_t - \sigma_0)^2$. Assuming the portfolio has an initial wealth $x_0 = 5$, the analytical solution facilitates numerical comparison. Figure 2.12(a) compares the learned value functions with the true values for a range of λ_0. It shows good

Algorithm 2 Training Generative Adversarial Networks

```
Given the initial condition $X_0^m = x_0, \forall m \in [1, M]$, and the initial starting point of $\sigma = \{\sigma_n\}_{n,m=1}^{N-1,M}$:
for epoch = 1 : number of epochs do
    Phase 1: train the $\alpha$-generator
    for time step $n = 1 : N-1$ do
        With the network $\mathcal{A}_n$, inputs $\{X_n\}_{m=1}^M$, parameters $\{\sigma_n\}_{m=1}^M$, outputs $\alpha_{n,m} = \mathcal{A}_n(X_{n,m}, \sigma_{n,m})$;
        $X_{n+1,m} = X_{n,m} \exp\left\{\left(\alpha_{n,m}(\mu - r) + r - \frac{1}{2}\alpha_{n,m}^2\sigma_{n,m}^2\right)\Delta t + \alpha_{n,m}\sigma_{n,m}\Delta W_{n,m}\right\}$;
    end
    Loss function $L_1 = -\frac{1}{M}\sum_{m=1}^M \left\{ U(X_{N,m}) + \lambda_0 \sum_{n=1}^{N-1} F(\sigma_{n,m}^2)\Delta t \right\}$;
    Train the neurons with an Adam optimizer and update $\mathcal{A}_n, n \in [1, N-1]$.
    Phase 2: train the $\sigma$-generator
    for time step $n = 1 : N-1$ do
        With the network $\mathcal{S}_n$, inputs $\{X_n\}_{m=1}^M$, parameters $\{\alpha_n\}_{m=1}^M$, outputs $\sigma_{n.m} = \mathcal{S}_n(X_{n,m}, \alpha_{n,m})$;
        $X_{n+1,m} = X_{n,m} \exp\left\{\left(\alpha_{n,m}(\mu - r) + r - \frac{1}{2}\alpha_{n,m}^2\sigma_{n,m}^2\right)\Delta t + \alpha_{n,m}\sigma_{n,m}\Delta W_{n,m}\right\}$;
    end
    Loss function $L_2 = \frac{1}{M}\sum_{m=1}^M \left\{ U(X_{N,m}) + \lambda_0 \sum_{n=1}^{N-1} F(\sigma_{n,m}^2)\Delta t \right\}$;
    Train the neurons with an Adam optimizer and update $\mathcal{S}_n, n \in [1, N-1]$.
end
```

accuracy of the learned functions versus the true ones. The errors are of magnitude 10^{-5}. The loss function L_2 during the training is presented in Figure 2.12(b). Unlike the trend in training regular deep neural networks, the loss function is not monotonically decreasing. As we can see, the minimizer was dominating the competition at the beginning, the loss function decreasing rapidly. Then the maximizer caught up, the loss function increased for a while and finally converged to the true value.

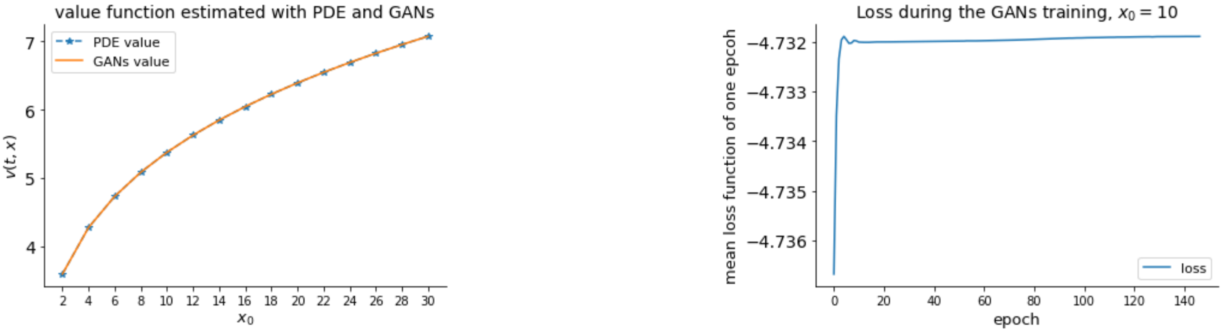
In the second example, we use a utility function $U(X_T) = 3X_T^{\frac{1}{4}}$ and a cost function $F(\sigma_t^2) = (\sigma_t^2)^2$. We set $\lambda_0 = 10$ in this case and estimate the value functions for a range of x_0. Since we do not have access to the true values for power utility, we compare the GANs estimated values with the PDE estimations in Figure 2.13(a). The loss function L_1 for $x_0 = 6$ during the training is presented in Figure 2.13(b).



(a) value functions estimated with GANs, $U(X_T) = \ln(X_T)$

(b) the minimizer's loss function, $\lambda_0 = 10$

Figure 2.12



(a) value functions estimated with GANs, $U(X_T) = 3X_T^{\frac{1}{4}}$

(b) the maximizer's loss function, $x_0 = 6$

Figure 2.13

Despite the promising results, a limitation of GANs, shared with deep neural networks in general, is the sensitivity of training to the chosen parameters. On difficult problems, fine-tuning the hyper-parameters of the GAN to facilitate training might require a lot of effort. One standard strategy for stabilizing training is to carefully design the model, either by adopting a proper architecture (Radford et al., 2015) or by selecting an easy-to-optimize objective function (Salimans et al., 2016). In spite of this caveat, GANs can be considered a viable contender to the more classical Monte Carlo methods of section 2.6.3 for robust portfolio allocation involving multiple risky assets, and deserve further investigation.

2.6.3 Monte Carlo method

In this section, we implement a Regression Monte Carlo scheme to solve the same robust portfolio allocation problems. Carriere (1996) introduced the Regression Monte Carlo approach to solve optimal stopping problems for any Markovian process in discrete time. In particular, he used non-parametric regression techniques. Later, Tsitsiklis and Van Roy (2001) and Longstaff and Schwartz (2001) used a similar scheme with ordinary least squares (a.k.a. Least Squares Monte Carlo) to value American options, respectively by value iteration and by performance iteration (see for example Denault and Simonato 2017). Since then, Regression Monte Carlo has become a popular tool in option pricing and more generally for solving discrete-time stochastic control problems in finite horizon.

First of all, we discretize the time interval $[0, T]$ into N time steps with a constant step size $\Delta t = \frac{T}{N}$. Using the Euler scheme on the logarithm of the state variable, one obtains the following dynamics for the discrete-time wealth X_n:

$$\begin{aligned} X_0 &= x \\ X_{n+1} &= X_n \exp\left(\left[(\alpha_n^\intercal(\mu - \mathbf{r}) + r - \frac{1}{2}\alpha_n^\intercal \Sigma_n \alpha_n\right] \Delta t + \alpha_n^\intercal (\Sigma_n)^{\frac{1}{2}} \Delta W_n\right) \ , n \in [0, N-1] \end{aligned} \tag{2.32}$$

and the discretized form of our value is

$$\overline{v}(0, X_0) = \sup_{\alpha \in \mathcal{A}} \inf_{\Sigma \in \mathcal{B}} \left\{ \mathbb{E}\big[\lambda_0 \sum_{n=0}^{N-1} F(\Sigma_n)\Delta t + U(X_N) \big| X_0 = x\big] \right\}. \tag{2.33}$$

As we have proved in section 2.3.2, this value function satisfies the DPP:

$$\begin{aligned} \overline{v}(N, X_N) &= U(X_N) \\ \overline{v}(n, X_n) &= \sup_{\alpha \in \mathcal{A}} \inf_{\Sigma \in \mathcal{B}} \left\{ \lambda_0 F(\Sigma_n)\Delta t + \mathbb{E}\big[\overline{v}(n+1, X_{n+1}) \big| \mathcal{F}_n\big] \right\} \ , n \in [0, N-1] \,. \end{aligned} \tag{2.34}$$

Control randomization

Inspired by the Dynamic Programming Principle, we can start from the known terminal condition and compute the value functions backward in time recursively. Equation (2.34) involves a conditional expectation, which cannot be computed explicitly. Instead, one can for example use a least squares regression to approximate $\mathbb{E}\big[\overline{v}(n+1, X_{n+1}) \big| \mathcal{F}_n\big]$ with a polynomial basis function. The obstacle in the implementation is that we are not able to simulate the paths X_n forward, since the dynamics of the state variable depends on the uncertain controls. Following Kharroubi et al. (2014), one way to tackle this problem is an initial randomization of the controls, i.e., we choose an arbitrary initial distribution for the controls and simulate the X_n with these dummy α_n and Σ_n , before including these dummy controls in the regressors of the least-squares regressions.

Proofs of the convergence and error bounds for standard Regression Monte Carlo are available in Clément et al. (2002) and Beutner et al. (2013) for example. In the case of controlled dynamics, Kharroubi et al. (2015) analyzed the time-discretization error, and Kharroubi et al. (2014) investigated the projection error generated by approximating the conditional expectation by basis functions for the control randomization scheme. Recently, alternative randomization schemes have been proposed in the literature, such as Ludkovski and Maheshwari (2019), Balata and Palczewski (2018), Bachouch et al. (2018a) or Shen and Weng (2019), which are more amenable to comprehensive convergence proofs, see Balata and Palczewski (2017) and Huré et al. (2018). Nevertheless, the classical control randomization scheme retains some advantages, such as the ease with which it can handle switching costs, as shown in Zhang et al. (2019b).

For the choice of basis function ϕ, we can use a polynomial function in X_n, α_n, Σ_n, and let $\phi = \sum_{k=0}^{K} \beta_k \phi_k$. Once we complete the regression, we can approximate the conditional expected value function $\mathbb{E}\big[\overline{v}(n+1, X_{n+1}) \big| \mathcal{F}_n\big]$ in (2.34) by $\phi(\hat{\beta}; X_n, \alpha_n, \Sigma_n)$. For the mth simulation path, we can find the optimal controls by:

$$\begin{aligned} \hat{\Sigma}_n^m &= \arg\min_{\Sigma_n^m} \left\{ \lambda_0 F(\Sigma_n^m)\Delta t + \phi(\hat{\beta}; X_n^m, \alpha_n^m, \Sigma_n^m) \right\}, \\ \hat{\alpha}_n^m &= \arg\max_{\alpha_n^m} \left\{ \lambda_0 F(\hat{\Sigma}_n^m)\Delta t + \phi(\hat{\beta}; X_n^m, \alpha_n^m, \hat{\Sigma}_n^m(\alpha_n^m)) \right\}. \end{aligned}$$

The complete process is shown in Algorithm 3.

Algorithm 3 Control Randomization

```
Backward Regression:
  (1) Choose an initial distribution and generate initial random controls accordingly.
  (2) Generate $M$ paths of state variable $X_n$. The $m$th path starts from the initial condition $X_0^m = x$, evolves following the dynamics with $\{\alpha_n^m, \Sigma_n^m\}_{n=0}^{N-1}$ and assign $\overline{v}(N, X_N^m) = U(X_N^m)$.
  (3) For $n = N-1 : 0$ do
      (1) Regress $\big\{\overline{v}(n+1, X_{n+1}^m)\big\}_{m=1}^M$ on $\big\{X_n^m, \alpha_n^m, \Sigma_n^m\big\}_{m=1}^M$ , and get the regression coefficients $\big\{\hat{\beta}_{n+1}^k\big\}_k$
      (2) Find the optimal controls $\hat{\alpha}_n^m, \hat{\Sigma}_n^m$ by $\arg\max_\alpha \min_\Sigma \Big\{\lambda_0 F(\Sigma_n^m)\Delta t + \sum_{k=0}^K \hat{\beta}_{n+1}^k \phi_k(X_{n,}^m \alpha_n^m, \Sigma_n^m)\Big\}$
      (3) The value function at time step $n$ is $\overline{v}(n, X_n^m) = \lambda_0 F(\hat{\Sigma}_n^m)\Delta t + \sum_{k=0}^K \hat{\beta}_{n+1}^k \phi_k(X_{n,}^m \hat{\alpha}_n^m, \hat{\Sigma}_n^m)$
  (4) The value function $\overline{v}(0, x) = \frac{1}{M}\sum_{m=1}^M \overline{v}(0, X_0^m)$

Forward Resimulation:
  (1) Set the initial condition $\tilde{X}_0^m = x$
  (2) For $n = 0 : N-1$
      (1) Find the optimal controls $\tilde{\alpha}_n^m, s\tilde{\Sigma}_n^m$ by $\arg\max_{\alpha_n} \min_{\Sigma_n} \Big\{\lambda_0 F(\Sigma_n)\Delta t + \sum_{k=0}^K \hat{\beta}_{n+1}^k \phi_k(\tilde{X}_{n,}^m \alpha_n, \Sigma_n)\Big\}$, using the regression coefficients obtained in the backward part and the new state variable $\tilde{X}_n^m$.
      (2) The state variable at time step $n+1$ is $\tilde{X}_{n+1}^m = \tilde{X}_n^m \exp\Big\{\Big[(\tilde{\alpha}_n^m)^\intercal(\mu - \mathbf{r}) + r - \frac{1}{2}(\tilde{\alpha}_n^m)^\intercal \tilde{\Sigma}_n^m \tilde{\alpha}_n^m\Big]\Delta t + (\tilde{\alpha}_n^m)^\intercal (\tilde{\Sigma}_n^m)^{\frac{1}{2}} \Delta W_n\Big\}$
  (3) The forward simulated value function $\overline{v}_f(0, x) = \frac{1}{M}\sum_{m=1}^M \Big[\lambda_0 \sum_{n=0}^{N-1} F(\tilde{\Sigma}_n^m)\Delta t + U(\tilde{X}_N^m)\Big]$
```

Logarithmic utility function

We first consider an example with 1 risky asset. When the utility function is logarithmic and the penalty function is $\lambda_0 F(\sigma_t^2) = \lambda_0(\sigma_t^2)^2$, we choose the following basis function

$$\sum_{k=0}^K \beta_{n+1}^k \phi_k(X_{n,}\alpha_n, \sigma_n) = \beta_0 + \beta_1 \ln(X_n) + \beta_2 \alpha_n + \beta_3 \alpha_n \sigma_n + \beta_4 \sigma_n^2 \alpha_n^2.$$

To find the optimal controls, we differentiate $\lambda_0 F(\sigma_n^2)\Delta t+\sum_{k=0}^{K}\beta_{n+1}^k\phi_k(X_{n,}\alpha_n,\sigma_n)$ with respect to α_n and σ_n^2, then we can get the optimal controls by solving the following polynomial equation

$$4\lambda_0 dt\hat{\sigma}_n^6+\frac{\beta_2\beta_3}{2\beta_4}\hat{\sigma}_n+\frac{\beta_2^2}{2\beta_4}=0.$$

With $\beta_4<0$, there exists a real positive root. We can see the optimal controls are constants for each step, being independent of the state variable X_n, this is the same as our observation in the analytical solution.

We used $M=5\times 10^6$ paths, $T=1$ and step size $\Delta t=\frac{1}{50}$ in the simulation, with the parameters $x_0=5, r=0.015, \lambda_0=10$. Figure 2.14 shows the backward regression values, forward resimulation values and true values as we change the parameter μ. Figure 2.15 show how these values change as λ_0, the strength of the penalty function. Figure 2.16 compares the forward resimulation values, finite difference results and true values as we change the parameter μ. It shows that both the PDE and Monte Carlo approach the true value in this example.

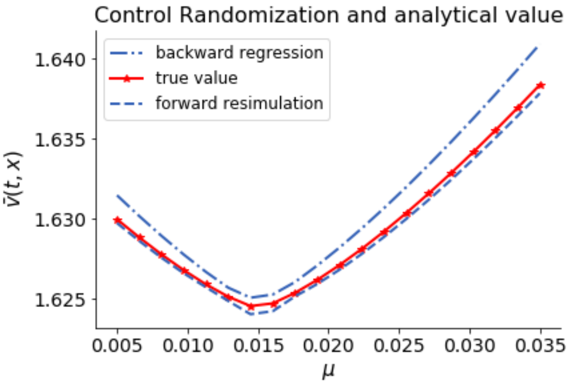


Figure 2.14: $U(X_T)=\ln(X_T), S_t\in\mathbb{R}^1$

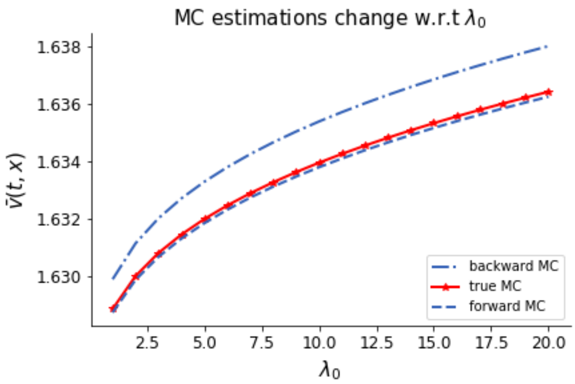


Figure 2.15: $U(X_T)=\ln(X_T), S_t\in\mathbb{R}^1$, with $x_0=5, r=0.015, \mu=0.03$

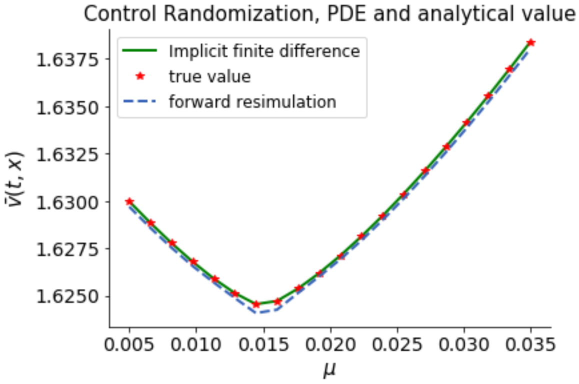


Figure 2.16: $U(X_T)=\ln(X_T), S_t\in\mathbb{R}^1$

For the example with 2 risky assets, we use the logarithmic utility function and the penalty function $F(\Sigma_t)=\lambda_0\left\|\Sigma_t\right\|_2^2$. We choose the following basis function in this case:

$$\sum_{k=0}^{K}\beta_{n+1}^k\phi_k(X_n,\alpha_n^1,\alpha_n^2,\sigma_n^1,\sigma_n^2,\rho_n)=\beta_0+\beta_1\ln(X_n)+\beta_2\alpha_n^1+\beta_3\alpha_n^2+\beta_4(\alpha_n^1)^2(\sigma_n^1)^2+\beta_5(\alpha_n^2)^2(\sigma_n^2)^2$$
$$+\beta_6\alpha_n^1\alpha_n^2\sigma_n^1\sigma_n^2\rho_n+\beta_7(\sigma_n^1)^4+\beta_8(\sigma_n^2)^4+\beta_9(\sigma_n^1)^2(\sigma_n^2)^2\rho_n^2,$$

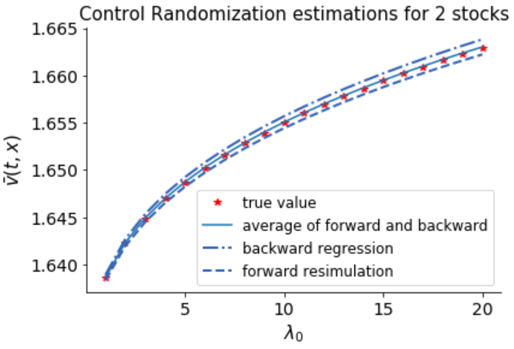


Figure 2.17: $U(X_T) = \ln(X_T)$, $S_t \in \mathbb{R}^2$, $x_0 = 5, r = 0.015, \mu_1 = 0.035, \mu_2 = 0.045$

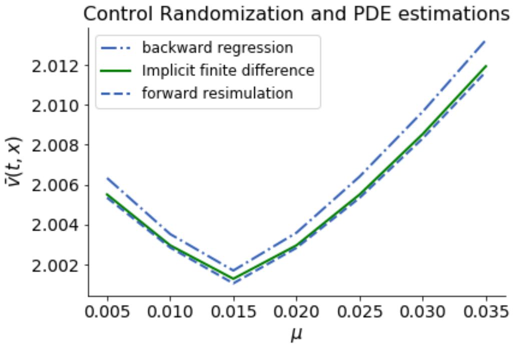


Figure 2.18: $U(X_T) = \frac{4}{3}X_T^{\frac{1}{4}}$

where σ_n^1, σ_n^2 are the volatilities of the two assets and ρ_n is the correlation between the assets. We can differentiate $\lambda_0 \|\Sigma_t\|_2^2 \Delta t + \sum_{k=0}^{K} \beta_{n+1}^k \phi_k(X_n, \alpha_n^1, \alpha_n^2, \sigma_n^1, \sigma_n^2, \rho_n)$ to get the optimal controls. In practice, we always have $\hat{\sigma}_n^1, \hat{\sigma}_n^2 > 0$, but we need to truncate $\hat{\rho}_n$ to $[-1, 1]$. The optimal controls are also constants for each step as in the 1-asset case.

In the implementation, we use $M = 4 \times 10^6$ paths, $T = 1$ and step size $\Delta t = \frac{1}{50}$. The result is provided in Figure 2.17. This plot compares the backward regression values, forward resimulation values and the analytical values, and it shows how the values change w.r.t. the penalty strength λ_0. From our observation, the average of the forward and backward results yields an even better estimate.

We can observe from Figure 2.14 and 2.17 that, as claimed in Kharroubi et al. (2014), the value function estimated at the end of the backward loop serves as an upper bound for the true value, while the one obtained from the forward resimulation serves as a lower bound and has a smaller error than the upper bound.

Power utility function

Here we show a 1-asset example with power utility. When the utility function is $U(X_T) = \frac{4}{3}X_T^{\frac{1}{4}}$ and the penalty function $\lambda_0 F(\sigma_t^2) = \lambda_0(\sigma_t^2)^2$, we choose the basis function

$$\phi = \beta_0 + \beta_1 X_n^{\frac{1}{4}} + \beta_2 X_n^{\frac{1}{4}} \alpha_n + \beta_3 X_n^{\frac{1}{4}} \alpha_n \sigma_n + \beta_4 X_n^{\frac{1}{4}} \alpha_n^2 \sigma_n^2. \tag{2.35}$$

To find the optimal controls, we differentiate $\lambda_0 F(\sigma_n^2)\Delta t + \sum_{k=0}^{K} \beta_{n+1}^k \phi_k(X_n, \alpha_n, \sigma_n)$ and then get the polynomial equation (2.36) for each path. We can see the optimal controls $\hat{\alpha}_n$ and $\hat{\sigma}_n$ depend on X_n in this case.

$$\beta_2^2 X_n^{\frac{1}{4}} + \beta_2 \beta_3 X_n^{\frac{1}{4}} \sigma_n + 8\beta_4 \lambda_0 dt \sigma_n^6 = 0 \tag{2.36}$$

Figure 2.18 shows Monte Carlo and finite difference approximations for a range of drifts μ, with $x_0 = 5, , r = 0.015, \lambda_0 = 10$, $M = 5 \times 10^6$, $N = 65$. We can see that the PDE estimates lie within the Monte Carlo bounds and that the forward simulation values almost overlap the PDE estimations. Although we do not have the analytical solution for this power utility case, these plots suggest that we are able to estimate the true values accurately with both Control Randomization and Finite Difference methods.

In both the logarithmic and power utility cases, the forward resimulation always performs better than the backward loop estimates. That is because the forward resimulation only suffers from one source of error, the optimal control estimation, while the backward regression suffers more directly from regression error (see Kharroubi et al. 2014). So the forward simulation result is a better estimator of the true value and is the one we use for comparison with the analytical and PDE approaches.

From the results above, we can see that for these robust portfolio allocation problems with one single risky asset, both PDE and Monte Carlo methods provide accurate estimates, with the PDE estimates being slightly better overall. Both methods can be considered for solving robust portfolio allocation problems in practice. Some difficulties with the Monte Carlo approach are the choice of the basis and the number of Monte Carlo paths needed for a stable convergence. Still, the Monte Carlo would be the method of choice for more realistic portfolio allocation with multiple risky assets (see Zhang et al. 2019b), as the PDE approach could quickly become computationally intractable in this situation.

2.7 Conclusion

In this chapter, we interpreted a robust portfolio optimization problem as a two-player zero-sum stochastic differential game. We compared the performance of the robust and non-robust portfolios using both Monte Carlo simulation and empirical market data. Under market shocks, our robust mechanism can prevent huge losses. By choose the λ_0 properly, the robust portfolios have a higher expected utility than the non-robust one. In addition to the finite difference method, we provide GANs and control randomization algorithms to estimate the value functions. These two methods can enrich quantitative techniques for solving robust portfolio optimization problems. Both of them have demonstrated high accuracy in the numerical results.

Chapter 3

Portfolio optimization with a prescribed terminal wealth distribution

This chapter studies a portfolio allocation problem, where the goal is to reach a prescribed wealth distribution at final time. We study this problem with the tools of optimal mass transport. We provide a dual formulation which we solve by a gradient descent algorithm. This involves solving an associated Hamilton–Jacobi–Bellman and Fokker–Planck equation by a finite difference method. Numerical examples for various prescribed terminal distributions are given, showing that we can successfully reach attainable targets. We next consider adding consumption during the investment process, to take into account distributions that are either not attainable, or sub-optimal.

3.1 Introduction

The classical objective function in a portfolio optimization problem is to maximize the expected return given variance level. However, the first and second moments of the return of a portfolio is only a simplified description of the wealth. Researchers then introduced objective functions that include more moments, such as skewness, to provide a more accurate statistic description of the distribution of the return (see, for example, Kraus and Litzenberger 1976 and Lee 1977).

The whole distribution of the portfolio wealth would provide investors a complete information. Instead of optimizing the first moments of the distribution, this work introduces an objective function which includes a target distribution of the terminal wealth. We address the problem of controlling the portfolio allocation process to reach the prescribed terminal distribution. Of course, as we will see not all distributions are attainable.

On the one hand, this problem can be categorized as a stochastic control problem. The state variable (the wealth) is driven by a process (the investment strategy) whose value is decided at any time $t \in [0, T]$, and we define such a process as a control. In our investment process, the portfolio allocation process is a control. We aim to design the time path of the portfolio allocation process such that it steers the portfolio wealth from an initial state to a prescribed terminal distribution.

One the other hand, designing a continuous semimartingale having prescribed distributions at given times can be addressed with the optimal mass transport (OMT) theory. The optimal transport problem is an old problem first addressed in the work of Monge (1781), and was later revisited by Kantorovich (1942) leading to the so-called Monge–Kantorovich formulation. Suppose a worker wants to move a pile of sand and the goal is to erect with all that sand a target pile with a prescribed shape. Naturally, the worker wishes to minimize the total effort. In the sense of physical, the work can be described as the product of force and displacement. Mathematically, the optimal transport problem can be formulated as follows. Let X, Y be two separable metric spaces such that probability measure on X (or Y) is a Radon measure. For measure μ on X and ν on Y, let Φ be the set of all maps $\phi : X \to Y$ such that $\phi_{\#}(\mu) = \nu$ and $\phi_{\#}^{-1}(\nu) = \mu$. The notation $\#$ represents the push-forward of μ by ϕ. Let $c : X \times Y \to [0, \infty]$ be a Borel-measurable cost function, the optimal map is defined as

$$\phi^* := \operatorname{arginf}_{\phi \in \Phi} \int_X c(x, \phi(x)) d\mu(x). \tag{3.1}$$

In the Monge formulation, the cost function is chosen to be $c(x, y) = ||x - y||^2$ and (3.1) becomes

$$\phi^* := \operatorname{arginf}_{\phi \in \Phi} \int_X ||x - \phi(x)||^2 d\mu(x), \tag{3.2}$$

and ϕ^* is called the optimal transport map. From Brenier's Theorem (Brenier, 1991), we know that if at least one of μ and ν has a density with respect to the Lebesgue measure, then the optimal transport map exists and is unique. However, when the conditions in Brenier's Theorem are not satisfies, the optimal transport map may not exist. To overcome this issue, Kantorovich (1942) introduced "couplings" π, which are joint distributions of μ and ν and the two marginal distributions of π are μ and ν, respectively. It is defined as

$$M(\mu, \nu) = \{\pi \in \mathcal{P}(X \times Y) \text{ s.t. } \pi(A, X) = \mu(A), \pi(Y, B) = \nu(B)\}. \tag{3.3}$$

Instead of looking for the optimal transport map, Kantorovich formulated the optimal transport problem as finding the optimal transport plan:

$$\pi^* := \operatorname{arginf}_{\pi \in M(\mu, \nu)} \int_{X \times Y} ||x - y||^2 d\pi(x, y). \tag{3.4}$$

Compared to the Monge formulation, the Kantorovitch formulation does not require the solution to be a one-to-one map and is more realistic in practice. Thus, the optimal solution of resource allocation problems is usually an optimal transport plan instead of an optimal transport map. However, when an optimal transport map exists, it is proved to be equivalent to the optimal transport plan.

A comprehensive review of the extensions and applications of the Monge–Kantorovich problem can be found in the book by Rachev and Rüschendorf (1998) and the books of Villani (2003, 2008). Benamou and Brenier (2000) reinterpreted the optimal transport problem in a fluid mechanics framework, where one is not looking only for an optimal transport map, but instead for the whole trajectory of the mass distribution over time. This contribution opened the way to the problem of continuous optimal transport. Stochastic extensions of the discrete and time continuous OMT problem have then flourished, see e.g. Mikami and Thieullen (2006), Tan et al. (2013), Mikami (2015), Henry-Labordère et al. (2016).

Beyond its mathematical interest, the optimal mass transport problem has applications in many fields, such as economy, meteorology, astrophysics (Brenier et al. 2003, Loeper 2006), image processing (Ferradans et al. 2014). In the field of finance, optimal transport has been applied in robust hedging (Dolinsky and Soner 2014, Henry-Labordère 2017) and volatility calibration (Guo et al. 2019b). Optimal transport is also closely connected to the Schrödinger problem, see Léonard (2013) for a comprehensive survey.

The novelty of this work is to provide a new perspective on portfolio optimization inspired by OMT. An investor must decide how to allocate her portfolio between a risky and a risk-free asset. The price of the risky asset is modelled by a semimartingale, with prescribed drift and diffusion coefficients. By controlling the portfolio allocation, she wants the distribution of the terminal wealth to match, or be close to, a given target distribution. Depending on the risky asset diffusion coefficients, not all target distributions are attainable (think for example of too high an expected return versus variance), or optimal (one could reach a "better" distribution than the target). We consider two different approaches: either relaxing the terminal constraint by penalization, or adding a consumption process, whereby the investor can either inject or withdraw cash from the portfolio in order to reach the target.

In other related papers, Zhang et al. (2019c) proposed an investment strategy which maximizes the expected portfolio value bounded within a targeted range. Chalabi and Wuertz (2012) sought the allocation that produces a portfolio that is the closest to a reference model while remaining as compatible as possible to the empirical data. They used a ϕ-divergence measure from information theory as the tool and they did not consider any cost function in the formulation. Chen et al. (2019b) formulated a class of stochastic two-person zero-sum differential games, where a specification on a target terminal state distribution is imposed on the players. However, their paper only applies to linear quadratic games with Gaussian target distributions. Instead, our problem has a general class of dynamics and cost functions. Chen et al. (2018) worked on guiding particles to a specific distribution. But the prescribed terminal distribution in their problem is restricted to Gaussian. By contrast, our method applies to a general choice of target distributions.

The rest of this chapter is organized as follows. In section 3.2, we formulate the problem. Then we introduce the dual formulation in section 3.3. In section 3.4, we provide a gradient descent algorithm to solve the dual problem, and the numerical results are presented in section 3.5. We give examples for general target distributions with various penalty functionals in section 3.5.1. We consider the addition of cash saving/cash input in section 3.5.2 and section 3.5.3.

3.2 Problem Formulation

Without loss of generality, we set the time horizon T to be 1. Let $\Omega := (\omega \in C([0,1];\mathbb{R}^d))$. The process W is a d-dimensional standard Brownian motion on the filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$. In this problem, the risk-free interest r is set to 0 for simplicity. The drift μ takes values in $\mathbb{R}^d$, the invertible matrix σ takes values in $\mathbb{R}^{d\times d}$ and the covariance matrix $\Sigma := \sigma\sigma^\intercal$ takes value in $\mathbb{S}^d$. We assume these are known processes.

The portfolio allocation process $\alpha = (\alpha_t)_{t\in[0,1]}$ is progressively measurable with respect to $\mathbb{F}$, taking values in a compact convex set $K \subset \mathbb{R}^d$. The set of all admissible α is compact and convex, denoted by $\mathcal{K}$.

We denote by $X_t \in \mathbb{R}$ the portfolio wealth at time t. Starting from an initial wealth x_0, the wealth of the self-financing portfolio evolves as follows,

$$dX_t = X_t\alpha_t^\intercal\mu(t, X_t)dt + X_t\alpha_t^\intercal\sigma(t, X_t)dW_t, \tag{3.5}$$

$$X_0 = x_0. \tag{3.6}$$

3.2.1 Portfolio optimization with a prescribed terminal distribution

We denote by $\rho_t := \mathbb{P} \circ X_t^{-1} \in \mathcal{P}(\mathbb{R})$ the distribution of X_t. In this problem, we are given the initial distribution of the state variable ρ_0 (ρ_0 can be a Dirac measure), a prescribed terminal distribution $\overline{\rho}_1 \in \mathcal{P}(\mathbb{R})$ and a convex cost function $f(\alpha_t) : K \to \mathbb{R}$.

With ρ_0 and a process α, the realized terminal distribution of the portfolio wealth is $\rho_1 := \mathbb{P} \circ X_1^{-1}$ (ρ_1 is not necessarily the same as $\overline{\rho}_1$). We want ρ_1 to be close to our target $\overline{\rho}_1$, hence we introduce a functional $C(\rho_1, \overline{\rho}_1)$ to penalize the deviation of ρ_1 from $\overline{\rho}_1$. At the same time, we want to minimize the expectation of the transportation cost from ρ_0 to ρ_1. Combining the expected transportation cost and the penalty functional, our objective function is

$$\inf_{\alpha,\rho} \left\{ \mathbb{E}^{\mathbb{P}} \left[\int_0^T f(\alpha_t) dt \right] + C(\rho_1, \overline{\rho}_1) \right\}, \tag{3.7}$$

where the feasible (α, ρ) in (3.7) should satisfy the initial distribution

$$\rho(0, x) = \rho_0(x) \quad \forall x \in \mathbb{R}. \tag{3.8}$$

Following closely Guo et al. (2019b)[Lemma3.1], we state the result below.

Lemma 3.2.1. For $t \in [0, 1]$, the optimal strategy α_t reduces to a Markovian process $\alpha(t, X_t)$.

Remark. This is actually a consequence of the following facts:

a. the cost function only depends on α_t and the constraint only depends on X_1 via ρ_1 (and not on any path dependent variable);

b. Jensen's inequality and the convexity of f.

The feasible (α, ρ) in (3.7) should satisfy the Fokker–Planck equation

$$\partial_t \rho(t,x) + \partial_x(\alpha(t,x)^\intercal \mu(t,x) x \rho(t,x)) - \frac{1}{2}\partial_{xx}(\alpha(t,x)^\intercal \Sigma(t,x)\alpha(t,x)x^2\rho(t,x)) = 0 \quad \forall (t,x) \in \mathcal{E}. \tag{3.9}$$

However, the feasible set for (α, ρ) defined by equality (3.9) is not convex, which means we may not be able to find the optimal solution. To address this issue, we introduce the following definition.

Notation 3.2.1. We define maps $\tilde{B}, \tilde{A} : \mathcal{E} \to \mathbb{R}$ as $\tilde{B}(t,x) := \alpha(t,x)^\intercal \mu(t,x) x$ and $\tilde{A}(t,x) := \alpha(t,x)^\intercal \Sigma(t,x)\alpha(t,x)x^2$. Then define $B(t,x) := \tilde{B}\rho$, $B \in \mathcal{M}(\mathcal{E}; \mathbb{R})$ and $A(t,x) := \tilde{A}\rho$, $A \in \mathcal{M}^+(\mathcal{E}; \mathbb{R})$. Densities B and A are absolutely continuous with respect to ρ.

Note that ρ, B, A are densities. As a shorthand, we will write $d\rho = \rho dt dx, dA = A dt dx, dB = B dt dx$ in the subsequent sections. We show that B and A are connected in the following way:

Proposition 3.2.1. When $d > 1$ (resp. $d = 1$), the necessary and sufficient condition for the existence of an $\alpha(t,x) \in \mathbb{R}^d$ such that $A = \alpha(t,x)^\intercal \Sigma(t,x)\alpha(t,x)x^2\rho$, $B = \alpha(t,x)^\intercal \mu(t,x) x \rho$ is $A \geq \frac{B^2}{\|\nu(t,x)\|^2 \rho}$ (resp. $A = \frac{B^2}{\|\nu(t,x)\|^2 \rho}$), where $\nu(t,x) := \Sigma(t,x)^{-\frac{1}{2}}\mu(t,x)$.

Proof. See appendix B.1. □

Using notations $\rho, B, \tilde{B}$ and $A, \tilde{A}$, the Fokker–Planck equation (3.9) becomes linear and the SDE of the portfolio wealth reads

$$dX_t = \tilde{B}(t, X_t)dt + \tilde{A}^{\frac{1}{2}}(t, X_t)dW_t, \tag{3.10}$$

$$X_0 = x_0. \tag{3.11}$$

To attain a certain prescribed terminal distribution, as we see in Proposition 3.2.1, it is possible that the optimal drift $\tilde{B}(t,x)$ is not saturated, i.e., $\tilde{B}(t,x)^2 < \|\nu(t,x)\|^2 \tilde{A}(t,x)$. It means that the drift function in SDE (3.10) can be greater and eventually we should be able to reach a better terminal wealth with a higher expectation. Then, instead of using this unsaturated drift $\tilde{B}(t,x)$ to reach the prescribed terminal distribution, we can use the drift $\tilde{B}(t,x) = \|\nu(t,x)\| \sqrt{\tilde{A}(t,x)}$ to attain a more ambitious distribution, and the extra part in the drift can be interpreted as cash saving. In this case, even when we have multiple assets $(d > 1)$ in the portfolio, optimal portfolios should lie on the curve $\tilde{B}(t,x) = \|\nu(t,x)\| \sqrt{\tilde{A}(t,x)}$, as in the $d = 1$ case. Any portfolio lying below the curve represents a less than ideal investment because for the same level of risk (variance), we could achieve a greater return. This is consistent with the efficient frontier in modern portfolio theory (Markowitz 1952).

Now we define the concept of *cash saving* at time t as $c(t,x) \coloneqq \|\nu(t,x)\| \sqrt{\tilde{A}(t,x)} - \tilde{B}(t,x)$. When the prescribed terminal distribution is not ambitious enough, to ensure we have as much cash saving as we can, we define the feasible set as $\Pi(t,x) \coloneqq \{(\tilde{B}, \tilde{A}) : \tilde{A} \geq \frac{(\tilde{B}^+)^2}{\|\nu(t,x)\|^2}\}$ $(\tilde{B}^+(t,x) \coloneqq \max(0, \tilde{B}(t,x)))$ and we can see the set $\Pi(t,x)$ is convex.

In the following parts, we focus on a particular subset of f, which can be represented by $f(\alpha(t,x)^\intercal \Sigma(t,x) \alpha(t,x) x^2)$. To penalize functions out of the set $\Pi(t,x)$, we define a cost function $F : \mathcal{E} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \cup \{+\infty\}$ such that

$$F(p,q) = f\,(q) + \delta(p,q), \tag{3.12}$$

where $f : \mathcal{E} \times \mathbb{R} \to \mathbb{R}^+$ is a convex function and $\delta(p,q)$ is a delta function defined as

$$\delta(p,q) = \begin{cases} 0 & \text{if } (p,q) \in \Pi(t,x), \\ +\infty & \text{otherwise.} \end{cases} \tag{3.13}$$

For simplicity, we write $F(p,q) \coloneqq F(t,x,p,q)$ if there is no ambiguity.

Now we are ready to introduce formally the problem:

Problem 3.2.1. Starting from an initial distribution ρ_0, with a prescribed terminal distribution $\overline{\rho}_1$ and a cost function (3.12), we want to solve the infimum of the functional

$$V(\rho_0, \overline{\rho}_1) = \inf_{\rho, B, A} \int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho + C(\rho_1, \overline{\rho}_1) \tag{3.14}$$

over all $(\rho, B, A) \in \mathcal{M}(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R})$ satisfying the constraints

$$\partial_t \rho(t,x) + \partial_x B(t,x) - \frac{1}{2}\partial_{xx} A(t,x) = 0 \quad \forall (t,x) \in \mathcal{E}, \tag{3.15}$$

$$\rho(0,x) = \rho_0(x) \quad \forall x \in \mathbb{R}. \tag{3.16}$$

3.2.2 Assumptions

Assumption 3.2.1. The penalty functional $C(\cdot, \overline{\rho}_1) : \mathcal{P}(\mathbb{R}) \to \mathbb{R}^+$ is convex and lower semi-continuous with respect to the weak$-*$ convergence of ρ. We have $C(\rho_1, \overline{\rho}_1) = 0$ if and only if $\rho_1(x) = \overline{\rho}_1(x)$ almost everywhere.

Assumption 3.2.2.

(i) The cost function $F(p, q)$ is non-negative, lower semi-continuous and strictly convex in (p, q).

(ii) The cost function $F(p, q)$ is coercive in the sense that there exist constants $m > 1$ and $C > 0$ such that

$$|p|^m + |q|^m \leq C\left(1 + F\left(p, q\right)\right) \quad \forall (p, q) \in \mathit{\Pi}.$$

3.3 Duality

In this section, we introduce the dual problem to Problem 3.2.1, this allows us to give optimality condition for the primal problem. First of all, we find out the convex conjugate of the cost functional, which will be used in the later proof.

3.3.1 Convex Conjugate

Define a function $G : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ as

$$\begin{aligned} G(u, b, a) &= \sup_{(\rho, \tilde{B}, \tilde{A}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}} \Big\{ u\rho + b\tilde{B}\rho + a\tilde{A}\rho - F(\tilde{B}, \tilde{A})\rho \Big\} \\ &= \sup_{\rho \in \mathbb{R}} \Big\{ \rho \Big[u + \sup_{\tilde{A} \geq \frac{(\tilde{B}^+)^2}{\|\nu(t,x)\|^2}} \Big(b\tilde{B} + a\tilde{A} - F(\tilde{B}, \tilde{A}) \Big) \Big] \Big\} \\ &= \sup_{\rho \in \mathbb{R}} \Big\{ \rho \Big[u + F^*(b, a) \Big] \Big\}, \end{aligned}$$

where F^* is the convex conjugate of F. Since $\rho(t, x)$ is non-negative, it is obvious that

$$G(u, b, a) = \begin{cases} 0 & \text{if } u + F^*(b, a) \leq 0 \quad \forall (t, x) \in \mathcal{E}, \\ +\infty & \text{otherwise.} \end{cases}$$

If we restrict the domain of its convex conjugate to $\mathcal{M}(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R})$, then

$$G^*(\rho, B, A) = \sup_{(u,b,a) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}} \left\{ u\rho + bB + aA : u + F^*(b, a) \leq 0 \right\}. \tag{3.17}$$

Because the function to be optimized is linear and $\rho(t, x) \geq 0$, we can see the optimal $u^* = -F^*(b, a)$ in (3.17). With F being convex and lower-semicontinuous, we have

$$\begin{aligned} G^*(\rho, B, A) &= \sup_{(b,a) \in \mathbb{R} \times \mathbb{R}} \Big\{ -F^*(b, a) + b\tilde{B} + a\tilde{A} \Big\} \rho \\ &= F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) \rho. \end{aligned}$$

We interchange the supremum and integral following Guo et al. (2019b, Lemma A.1), and get

$$\int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho = \sup_{(u,b,a)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{\int_{\mathcal{E}} ud\rho + bdB + adA : u + F^*(b,a) \leq 0\right\}. \quad (3.18)$$

Note that (u, b, a) are real numbers only in Section 3.3.1, and they are functions after the interchange, this will apply to the following sections.

3.3.2 Dual Problem

Now we can state our main result. A key element in the dual problem is the Hamilton–Jacobi–Bellman (HJB) equation:

$$\partial_t \phi + \sup_{\tilde{A} \geq \frac{(\tilde{B}+)^2}{\|\nu(t,x)\|^2}} \left\{\partial_x \phi \tilde{B} + \frac{1}{2}\partial_{xx}\phi \tilde{A} - F(\tilde{B}, \tilde{A})\right\} = 0. \quad (3.19)$$

If $\phi(t, x) \in C_b^{1,2}(\mathcal{E})$ is a solution of the HJB equation (3.19), then Itô's formula yields,

$$\begin{aligned}\int_{\mathbb{R}} \phi_1 d\rho_1 - \phi_0 d\rho_0 &= \int_{\mathcal{E}} \left(\partial_t \phi + \partial_x \phi \tilde{B} + \frac{1}{2}\partial_{xx}\phi \tilde{A}\right) d\rho \\ &= \int_{\mathcal{E}} \left(-F^*(\partial_x \phi, \frac{1}{2}\partial_{xx}\phi) + \partial_x \phi \tilde{B} + \frac{1}{2}\partial_{xx}\phi \tilde{A}\right) d\rho \\ &\leq \int_{\mathcal{E}} F(\tilde{B}, \tilde{A}) d\rho.\end{aligned}$$

Adding the penalty functional to both sides yields

$$\int_{\mathbb{R}} \phi_1 d\rho_1 - \int_{\mathbb{R}} \phi_0 d\rho_0 + C(\rho_1, \overline{\rho}_1) \leq \int_{\mathcal{E}} F(\tilde{B}, \tilde{A}) d\rho + C(\rho_1, \overline{\rho}_1). \quad (3.20)$$

Taking the infimum of the left hand side of (3.20) over ρ_1 and taking the infimum of the right hand side of (3.20) over (ρ, B, A), we get

$$\begin{aligned}-C^*(-\phi_1) - \int_{\mathbb{R}} \phi_0 d\rho_0 &\leq \inf_{(\rho,B,A)\in\mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho + C(\rho_1, \overline{\rho}_1) \\ &\leq V(\rho_0, \overline{\rho}_1),\end{aligned}$$

where C^* is the convex conjugate of C.

The following result shows that optimizing the left hand side yields an equality.

Theorem 3.3.1 (Duality)**.** When $C(\cdot, \overline{\rho}_1)$ is continuous, there holds

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi} \left\{-C^*(-\phi_1) - \int_{\mathbb{R}} \phi_0 d\rho_0\right\}, \quad (3.21)$$

where the supremum is taken over all $\phi(t, x) \in C_b^{1,2}(\mathcal{E})$ satisfying

$$\partial_t \phi(t, x) + \sup_{\tilde{A} \geq \frac{(\tilde{B}+)^2}{\|\nu(t,x)\|^2}} \left\{\partial_x \phi \tilde{B} + \frac{1}{2}\partial_{xx}\phi \tilde{A} - F(\tilde{B}, \tilde{A})\right\} \leq 0, \qquad \forall (t, x) \in \mathcal{E}. \quad (3.22)$$

Proof. This proof is an application of the Fenchel–Rockafellar duality theorem, e.g., Brezis (2010, Theorem 1.12). The constraint (3.15) can be formulated in the following weak form:

$$\int_{\mathbb{R}} \phi_1 d\rho_1 - \phi_0 d\rho_0 - \int_{\mathcal{E}} \partial_t \phi d\rho + \partial_x \phi dB + \frac{1}{2}\partial_{xx}\phi dA = 0, \quad \forall \phi \in C_b^{1,2}(\mathcal{E}). \tag{3.23}$$

Because of equation (3.18), we can reformulate the primal problem (3.14) as a saddle point problem:

$$V(\rho_0, \overline{\rho}_1) = \inf_{\rho,B,A} \sup_{u+F^*(b,a)\leq 0} \int_{\mathcal{E}} ud\rho + bdB + adA + C(\rho_1, \overline{\rho}_1). \tag{3.24}$$

Adding the Lagrangian penalty (3.23) to the functional (3.24), then Problem 3.2.1 can be written as

$$\begin{aligned} V(\rho_0, \overline{\rho}_1) = \inf_{\rho,B,A} \sup_{u+F^*(b,a)\leq 0,\phi} & \int_{\mathcal{E}} ud\rho + bdB + adA + C(\rho_1, \overline{\rho}_1) \\ & + \int_{\mathbb{R}} \phi_1 d\rho_1 - \phi_0 d\rho_0 - \int_{\mathcal{E}} \partial_t \phi d\rho + \partial_x \phi dB + \frac{1}{2}\partial_{xx}\phi dA. \end{aligned} \tag{3.25}$$

The strategy of the proof is to first construct a function α whose convex conjugate α^* is equal to the objective function of Problem 3.2.1, and construct another function β whose convex conjugate β^* is equal to the rest part inside the infimum of (3.25) so that $V = \inf_{\rho,B,A}(\alpha^* + \beta^*)(\rho, A, B)$. Then, the duality is established by applying the FenchelRockafellar theorem.

We write $C^*(r) : C_b(\mathbb{R};\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ for the convex conjugate of functional $C(\rho_1, \overline{\rho}_1)$:

$$C^*(r) = \sup_{\rho_1 \geq 0}\Big\{\int_{\mathbb{R}} rd\rho_1 - C(\rho_1, \overline{\rho}_1)\Big\}.$$

Here we define the functional $\alpha : C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ by

$$\alpha(u,b,a,r) = \begin{cases} C^*(r) & \text{if } u + F^*(b,a) \leq 0, \\ +\infty & \text{otherwise .} \end{cases} \tag{3.26}$$

Its convex conjugate $\alpha^* : C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ is defined as

$$\alpha^*(\rho,B,A,\rho_1) = \sup_{(u,b,a,r)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \Big(\int_{\mathcal{E}} ud\rho + bdB + adA + \Big[\int_{\mathbb{R}} rd\rho_1 - C^*(r)\Big]; u + F^*(b,a) \leq 0\Big). \tag{3.27}$$

If we restrict the domain to $\mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})$, with (3.18) and Assumption 3.2.1, we have

$$\alpha^*(\rho,B,A,\rho_1) = \begin{cases} \int_{\mathcal{E}} F(\frac{B}{\rho}, \frac{A}{\rho})d\rho + C(\rho_1, \overline{\rho}_1) & \text{if } \rho \in \mathcal{M}_+ \text{ and } B = \tilde{B}\rho, A = \tilde{A}\rho, \\ +\infty & \text{otherwise.} \end{cases}$$

Indeed, if ρ is not positive in (3.27), we would let $b = a = 0$ and $u = -\lambda \mathbb{1}_O$ for some O such that $\rho(O) < 0$ and let $\lambda \to +\infty$. If B or A are not absolutely continuous with respect to ρ, we can find some O such that $\rho(O) = 0$ but $B(O) \neq 0$ or $A(O) \neq 0$. Then we let $u = -F^*(b,a)$ and $b = a = \lambda \mathbb{1}_O$, and $\alpha^*(\rho,B,A) \geq \lambda B(O) + \lambda A(O) \to +\infty$ by letting $\lambda \to \pm\infty$ depending on the sign of $B(O)$ and $A(O)$.

Next, we say that the set $(u,b,a,r) \in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})$ is *represented* by $\phi \in C_b^{1,2}(\mathcal{E})$ if

$$u = -\partial_t\phi, \quad b = -\partial_x\phi, \quad a = -\frac{1}{2}\partial_{xx}\phi, \quad r = \phi_1.$$

Then define $\beta : C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ as follows,

$$\beta(u,b,a,r) = \begin{cases} \int_{\mathbb{R}} \phi_0 d\rho_0 & \text{if } (u,b,a,r) \text{ is represented by } \phi \in C_b^{1,2}(\mathcal{E}), \\ +\infty & \text{otherwise.} \end{cases} \tag{3.28}$$

Notice that β is well-defined, indeed, it does not depend on the choice of ϕ. If both ϕ,ψ represent u,b,a,r, then $\phi_1 = \psi_1 \forall x \in \mathbb{R}$, $\partial_t\phi(t,x) = \partial_t\psi(t,x)$, $\partial_x\phi(t,x) = \partial_x\psi(t,x)$, $\partial_{xx}\phi(t,x) = \partial_{xx}\psi(t,x)$ $\forall (t,x) \in \mathcal{E}$. It follows that $\phi_0(x) = \psi_0(x)$ $\forall x \in \mathbb{R}$. The set of represented functions (u,b,a,r) is a linear subspace. Hence β is convex and its convex conjugate $\beta^* : C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ is

$$\beta^*(\rho,B,A,\rho_1) = \sup_{u,b,a,r} \int_{\mathcal{E}} u d\rho + b dB + a dA + \int_{\mathbb{R}} r d\rho_1 - \phi_0 d\rho_0,$$

$$\text{over all } (u,b,a,r) \in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \text{ represented by } \phi \in C_b^{1,2}(\mathcal{E}).$$

Or equivalently,

$$\beta^*(\rho,B,A,\rho_1) = \sup_{\phi} \int_{\mathcal{E}} -\partial_t\phi d\rho - \partial_x\phi dB - \frac{1}{2}\partial_{xx}\phi dA + \int_{\mathbb{R}} \phi_1 d\rho_1 - \phi_0 d\rho_0.$$

We find that $\beta^*(\rho,B,A,\rho_1) = 0$ if (ρ,B,A,ρ_1) satisfies (3.23), and $\beta^*(\rho,B,A,\rho_1) = +\infty$ otherwise.

Now we can express our objective functional $V(\rho_0,\overline{\rho}_1)$ as

$$\begin{aligned} V(\rho_0,\overline{\rho}_1) &= \inf_{(\rho,B,A)\in\mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\} \\ &= \inf_{(\rho,B,A,\rho_1)\in\mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\} \\ &= \inf_{(\rho,B,A,\rho_1)\in C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\}. \end{aligned}$$

The second equality is because $\beta^*(\rho,B,A,\rho_1) = +\infty$ if ρ_1 does not equal to $\rho(t,x)$ at time $t = 1$. We prove the third equality in appendix B.2.

We can let $\phi(t,x) = t$, then $u = -1, b = 0, a = 0, r = 1$. We can see $\alpha(-1,0,0,1) = 1$ and it is continuous in (u,b,a,r) at this point, and $\beta(-1,0,0,1) = 0$ being finite at this point. Finally, the conditions of Fenchel duality theorem in Brezis (2010, Theorem 1.12) are fulfilled, and it implies

$$\begin{aligned} V(\rho_0,\overline{\rho}_1) &= \inf_{(\rho,B,A,\rho_1)\in C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\} \\ &= \sup_{(u,b,a,r)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{-\alpha(-u,-b,-a,-r) - \beta(u,b,a,r)\}, \end{aligned}$$

over the set (u,b,a,r) being represented by $\phi \in C_b^{1,2}(\mathcal{E})$, and satisfying $-u + F^*(-b,-a) \leq 0$.

Therefore we express $V(\rho_0,\overline{\rho}_1)$ in terms of ϕ:

$$\begin{aligned} V(\rho_0,\overline{\rho}_1) &= \sup_{(u,b,a,r)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{-C^*(-r) - \int_{\mathbb{R}} \phi_0 d\rho_0\right\} \\ &= \sup_{\phi\in C_b^{1,2}(\mathcal{E})} \left\{-C^*(-\phi_1) - \int_{\mathbb{R}} \phi_0 d\rho_0\right\}, \end{aligned}$$

under the constraint $\partial_t\phi + F^*(\partial_x\phi, \frac{1}{2}\partial_{xx}\phi) \leq 0$. As a consequence of Fenchel duality theorem, the infimum in the primal problem is attained if finite. This completes the proof. □

Actually, using the same proof as in Guo et al. (2019b), we can write the dual formulation in the following way:

Corollary 3.3.1. When $C(\cdot, \overline{\rho}_1)$ is continuous, there holds

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ -C^*(-\phi_1) - \int_{\mathbb{R}} \phi_0 d\rho_0 \right\}, \tag{3.29}$$

where the supremum is running over all functions $\phi_1 \in C_b^2(\mathbb{R})$, and ϕ_0 is a viscosity solution of the Hamilton–Jacobi–Bellman equation

$$\begin{cases} -\phi_t - \sup_{\tilde{A} \geq \frac{(\tilde{B}+)^2}{\|\nu(t,x)\|^2}} \left[\phi_x \tilde{B} + \frac{1}{2}\phi_{xx}\tilde{A} - F(\tilde{B}, \tilde{A}) \right] = 0, & \text{in } [0,1) \times \mathbb{R}, \\ \phi(1,x) = \phi_1(x), & \text{on } [1] \times \mathbb{R}. \end{cases} \tag{3.30}$$

Because the minimal objective function (3.14) is a trade-off between the cost function and the penalty functional, the optimal ϕ_1 in the dual problem (3.29) will not in general ensure that ρ_1 reaches $\overline{\rho}_1$, unless the penalty functional goes to infinity for $\rho_1 \neq \overline{\rho}_1$. When $\overline{\rho}_1$ is attainable, it can be realized by choosing the penalty functional as an indicator function

$$C(\rho_1, \overline{\rho}_1) = \begin{cases} 0 & \text{if } \rho_1 = \overline{\rho}_1, \\ +\infty & \text{if } \rho_1 \neq \overline{\rho}_1. \end{cases} \tag{3.31}$$

Using the penalty functional (3.31) is equivalent to adding the terminal constraint $\rho_1 = \overline{\rho}_1$, $\forall x \in \mathbb{R}$. This also recovers our problem to the classical optimal transport problem.

Corollary 3.3.2. When $C(\rho_1, \overline{\rho}_1)$ is defined as (3.31), there holds

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ \int_{\mathbb{R}} \phi_1 d\overline{\rho}_1 - \phi_0 d\rho_0 \right\}, \tag{3.32}$$

where the supremum is running over all $\phi_1 \in C_b^2(\mathbb{R})$ and ϕ_0 is a viscosity solution of the Hamilton–Jacobi–Bellman equation (3.30).

Proof. This proof is very similar to the one of Theorem 3.3.1, hence the repetitive steps are omitted here. Being different from the proof of Theorem 3.3.1, in this case, we define the functional $\alpha : C_b(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}) \to \mathbb{R} \cup \{+\infty\}$ by

$$\alpha(u, b, a, r) = \begin{cases} \int_{\mathbb{R}} r d\overline{\rho}_1 & \text{if } u + F^*(b,a) \leq 0, \\ +\infty & \text{otherwise}. \end{cases}$$

Then its convex conjugate of $\alpha^* : C_b^*(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}) \to \mathbb{R} \cup \{+\infty\}$ is

$$\alpha^*(\rho, B, A, \rho_1) = \sup_{u + F^*(b,a) \leq 0, r} \int_{\mathcal{E}} u d\rho + b dB + a dA + \int_{\mathbb{R}} r d\rho_1 - r d\overline{\rho}_1.$$

We restrict the domain to $\mathcal{M}(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})$, we have

$$\begin{aligned} \alpha^*(\rho, B, A, \rho_1) &= \int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho + \sup_r \left\{ \int_{\mathbb{R}} r(d\rho_1 - d\overline{\rho}_1) \right\} \\ &= \int_{\mathcal{E}} F\left(\frac{B}{\rho}, \frac{A}{\rho}\right) d\rho + C(\rho_1, \overline{\rho}_1). \end{aligned}$$

Note that $\sup_r \int_{\mathbb{R}} r(d\rho_1 - d\overline{\rho}_1)$ is equal to 0 if $\rho_1 = \overline{\rho}_1 \forall x \in \mathbb{R}$ and is equal to $+\infty$ otherwise, which is equivalent to $C(\rho_1, \overline{\rho}_1)$ in (3.31). Define $\beta : C_b(\mathcal{E}; \mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}) \to \mathbb{R}\cup\{+\infty\}$ by (3.28), let $\phi(t,x) = t$, the conditions of Fenchel duality theorem in Brezis (2010, Theorem 1.12) are fulfilled. Therefore, we get

$$\begin{aligned} V(\rho_0, \overline{\rho}_1) &= \sup_{(u,b,a,r)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{-\alpha(-u,-b,-a,-r) - \beta(u,b,a,r)\} \\ &= \sup_{(u,b,a,r)\in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{ \int_{\mathbb{R}} r d\overline{\rho}_1 - \int_{\mathbb{R}} \phi_0 d\rho_0 \right\}, \end{aligned}$$

over the set (u,b,a,r) being represented by $\phi \in C_b^{1,2}(\mathcal{E})$, and satisfying $-u + F^*(-b,-a) \leq 0$. For the same reasons as in Corollary 3.3.1, we can express $V(\rho_0, \overline{\rho}_1)$ in terms of ϕ:

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ \int_{\mathbb{R}} \phi_1 d\overline{\rho}_1 - \int_{\mathbb{R}} \phi_0 d\rho_0 \right\},$$

where $\phi_0(x)$ is a viscosity solution of the Hamilton–Jacobi–Bellman equation (3.30). □

3.4 Numerical Methods for the Dual Problem

There has been a vast amount of numerical algorithms for the optimal mass transport problem. Gradient descent based methods are widely used to solve the reformulated dual problem of the Monge–Kantorovich problem, for example, by Chartrand et al. (2009) and Tan et al. (2013). Cuturi (2013) looked at transport problems from a maximum entropy perspective and computed the optimal transport distance through Sinkhorn's matrix scaling algorithm. This algorithm is also used for the entropic regularization of optimal transport by Benamou et al. (2019).

In this work, we also use a gradient descent based method to solve the dual problem in section 3.3. We know $\phi(t,x)$ is the solution of the HJB equation (3.30). For a given terminal function ϕ_1, we can calculate ϕ_0 by solving the HJB equation backward.

3.4.1 Finite Difference Scheme

First of all, to get $\phi(0,x)$, we solve the following PDE

$$\partial_t \phi + \sup_{\tilde{A} \geq \frac{(\tilde{B}+)^2}{\|\nu(t,x)\|^2}} \left\{ \partial_x \phi \tilde{B} + \frac{1}{2} \partial_{xx} \phi \tilde{A} - F(\tilde{A}, \tilde{B}) \right\} = 0, \tag{3.33}$$

with a given terminal boundary condition $\phi_1(x)$ backwardly, using an implicit finite difference scheme. We let

$$(\tilde{A}^*, \tilde{B}^*) = \operatorname*{argmax}_{\tilde{A} \geq \frac{(\tilde{B}+)^2}{\|\nu(t,x)\|^2}} \left\{ \partial_x \phi \tilde{B} + \frac{1}{2} \partial_{xx} \phi \tilde{A} - F(\tilde{A}, \tilde{B}) \right\} \in \mathbb{R}\times\mathbb{R}, \tag{3.34}$$

where $(\tilde{A}^*, \tilde{B}^*)$ are solved with a constrained optimization package in Matlab, using the "interior-point" algorithm, see MathWorks (2020) for the details.

In the numerical setting, we use N time steps and M space grid points. We use a constant time step Δt and a constant spatial step Δx. We discretize the PDE (3.33) using a forward approximation for $\partial_t\phi$, a central approximation for $\partial_x\phi$, and a standard approximation for $\partial_{xx}\phi$, the discretized form of PDE (3.33) at time step n is:

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t}+\tilde{B}^*\frac{\phi_{i+1}^n-\phi_{i-1}^n}{2\Delta x}+\frac{1}{2}\tilde{A}^*\frac{\phi_{i+1}^n+\phi_{i-1}^n-2\phi_i^n}{(\Delta x)^2}-F(\tilde{A}^*,\tilde{B}^*)=0.$$

With some manipulation, we get the

$$\left(\frac{\Delta t\tilde{A}_i^{*^n}}{2(\Delta x)^2}-\frac{\Delta t\tilde{B}_i^{*^n}}{2\Delta x}\right)\phi_{i-1}^n+\left(-1-\frac{\Delta t\tilde{A}_i^{*^n}}{(\Delta x)^2}\right)\phi_i^n+\left(\frac{\Delta t\tilde{A}_i^{*^n}}{2(\Delta x)^2}+\frac{\Delta t\tilde{B}_i^{*^n}}{2\Delta x}\right)\phi_{i+1}^n=-\phi_i^{n+1}+\Delta tF(\tilde{A}_i^{*^n},\tilde{B}_i^{*^n}), \tag{3.35}$$

where the optimal controls $\tilde{A}^{*^n}$ and $\tilde{B}^{*^n}$ depend on ϕ^n. It is difficult to check the stability condition in our PDE because the optimal $\tilde{A}^{*^n},\tilde{B}^{*^n}$ are unknown, but fortunately implicit finite difference methods have a weaker requirement than explicit finite difference methods. At the n-th time step of the implicit finite difference method, although we do not have the true values for ϕ^n, we can make an initial guess of $(\tilde{A}^{*^n},\tilde{B}^{*^n})_0$ using the known values ϕ^{n+1}, then use a fixed-point iteration scheme to generate a sequence $(\tilde{A}^{*^n},\tilde{B}^{*^n})_{k,k=1,2,\ldots}$ with $(\tilde{A}^{*^n},\tilde{B}^{*^n})_k$ being a function of $(\tilde{A}_{k-1}^{*^n},\tilde{B}_{k-1}^{*^n})$, until $(\tilde{A}^{*^n},\tilde{B}^{*^n})_k$ converges. The stopping criteria is when $||\phi_k^n-\phi_{k-1}^n||_2\leq 10^{-7}$. This method is also implemented in Guo et al. (2019a).

With the optimal drift $\tilde{B}^*$ and diffusion $\tilde{A}^*$ known, we can now propagate forward with the Fokker–Planck equation (3.15) to find the empirical terminal density ρ_1. With an initial wealth x_0, the initial distribution ρ_0 is a Dirac Delta distribution $\delta(x-x_0)$. We have

$$\partial_t\rho+\partial_x(\tilde{B}^*\rho)-\frac{1}{2}\partial_{xx}(\tilde{A}^*\rho)=0, \tag{3.36}$$

$$\rho(0,x)=\delta(x-x_0). \tag{3.37}$$

Since we used implicit finite difference to solve the HJB equation (3.33) backward, we use an explicit scheme for the forward Fokker–Planck equation (3.15). Then the discretized form at time step n is

$$\frac{\rho_i^{n+1}-\rho_i^n}{\Delta t}+\frac{\tilde{B}_{i+1}^{*^n}\rho_{i+1}^n-\tilde{B}_{i-1}^{*^n}\rho_{i-1}^n}{2\Delta x}-\frac{1}{2}\frac{\tilde{A}_{i+1}^{*^n}\rho_{i+1}^n+\tilde{A}_{i-1}^{*^n}\rho_{i-1}^n-2\tilde{A}_i^{*^n}\rho_i^n}{\Delta x^2}=0. \tag{3.38}$$

3.4.2 Optimization algorithm

Now, we are using a gradient descent method to seek for the optimal ϕ_1. A key role in the gradient descent method is the optimality condition. By providing a gradient, the computation is faster and more accurate. For convenience, we define another function

$$\tilde{V}(\phi_1):=C^*(-\phi_1)+\int_{\mathbb{R}}\phi_0\rho_0dx, \tag{3.39}$$

and $V(\rho_0,\overline{\rho}_1)=-\inf_{\phi_1}\tilde{V}(\phi_1)$. Then we need to find an optimal ϕ_1 to minimize $\tilde{V}(\phi_1)$. The change of $\tilde{V}(\phi_1)$ caused by the change of ϕ_1 is

$$\delta\tilde{V}(\phi_1)=\delta C^*(-\phi_1)+\int_{\mathbb{R}}\rho_0\frac{\partial\phi_0}{\partial\phi_1}\delta\phi_1dx, \tag{3.40}$$

$$=\delta C^*(-\phi_1)+\int_{\mathbb{R}}\rho_0\delta\phi_0dx. \tag{3.41}$$

We know that $\phi(t,x)$ in (3.39) satisfies the HJB equation $F^*\left(\partial_x\phi, \frac{1}{2}\partial_{xx}\phi\right) = -\partial_t\phi$. If we add an arbitrarily small variation $\delta\phi$ to ϕ and denote $\partial_x\phi$ as p and $\frac{1}{2}\partial_{xx}\phi$ as q for short, then we get $\partial_p F^*(p,q)\partial_x\delta\phi + \frac{1}{2}\partial_q F^*(p,q)\partial_{xx}\delta\phi = -\partial_t\delta\phi$, which is equivalent to

$$\partial_t\delta\phi + \partial x\delta\phi\tilde{B}^* + \frac{1}{2}\partial_{xx}\delta\phi\tilde{A}^* = 0, \tag{3.42}$$

where $\tilde{A}^*, \tilde{B}^*$ are the optimal controls, as defined in (3.34). Multiplying PDE (3.42) by density function $\rho(t,x)$ and with integration by parts, we have

$$\int_{\mathbb{R}} \rho_1\delta\phi_1 - \rho_0\delta\phi_0 dx - \int_{\mathbb{R}}\int_0^1 \delta\phi\partial_t\rho + \delta\phi\partial_x(\rho\tilde{B}^*) - \frac{1}{2}\delta\phi\partial_{xx}(\rho\tilde{A}^*)dxdt = 0.$$

Since the equation $\partial_t\rho + \partial_x(\rho\tilde{B}^*) - \frac{1}{2}\partial_{xx}(\rho\tilde{A}^*) = 0$ holds for admissible $(\rho, \tilde{A}^*, \tilde{B}^*)$, we get

$$\int_{\mathbb{R}} \rho_0\delta\phi_0 dx = \int_{\mathbb{R}} \rho_1\delta\phi_1 dx. \tag{3.43}$$

Substituting (3.43) into (3.41), we can see an optimal terminal function ϕ_1 should satisfy the optimality condition

$$\nabla_{\phi_1}\tilde{V} := \lim_{\delta\phi_1\to 0} \frac{\delta\tilde{V}(\phi_1)}{\delta\phi_1} \tag{3.44}$$

$$= \lim_{\delta\phi_1\to 0} \frac{\delta C^*(-\phi_1)}{\delta\phi_1} + \rho_1 = 0, \quad \forall x \in \mathbb{R}. \tag{3.45}$$

Remark 3.4.1. When $C(\overline{\rho}_1, \rho_1)$ is defined as (3.31), the corresponding optimality condition is

$$\nabla_{\phi_1}\tilde{V} = -\overline{\rho}_1 + \rho_1 = 0, \quad \forall x \in \mathbb{R}. \tag{3.46}$$

Now we are ready to solve the dual problem numerically. In Algorithm 4, we state the gradient descent based algorithm to look for the optimal ϕ_1 in (3.29). It includes solving the HJB equation and the Fokker–Planck equation with a finite difference method combined with a fixed-point iteration, as described in section 3.4.1. A similar numerical scheme can be found in Guo et al. (2019b) for calibrating volatilities by optimal transport.

3.5 Numerical Results

In this section, we will apply Algorithm 4 and demonstrate various numerical examples. Later, we will also consider the situations with cash saving and cash input during the investment process.

3.5.1 Penalty functional with an intensity parameter

Before we demonstrate the numerical results, we need to choose an appropriate penalty functional $C(\rho_1, \overline{\rho}_1)$. There are a range of methods to measure distribution discrepancy. A comprehensive survey on the distance or similarity measures between probability density functions (PDFs) is provided by Cha (2007). Note that our choice of penalty functional is not restricted to metrics, as long as $C(\rho_1, \overline{\rho}_1)$ satisfies Assumption 3.2.1 and describes similarity of the two PDFs.

The most intuitive choice is the L^2 norm of the difference. This quadratic function is convex and easy to implement. In the first example, we use the squared Euclidean distance

Algorithm 4 A gradient descent based optimization scheme

```
Initial guess $\phi_N^1 := 0$
while $1 \leq k \leq$ max iteration and $\left\|\nabla_{\phi_N} \tilde{V}(\phi_N^k)\right\|_\infty >$ tolerance do
    Let $\phi_N = \phi_N^k$;
    for time step $n = N - 1 : 0$: do
        Let $\phi_n = \phi_{n+1}$, solve the PDE (3.35) with $(\tilde{A}^{*^n}, \tilde{B}^{*^n})^0$ obtained from (3.34).
        Get the value vector $\phi_n^0$;
        while $1 \leq j \leq$ max iteration and $\left\|\phi_n^j - \phi_n^{j-1}\right\|_2 >$ tolerance do
            Let $\phi_n = \phi_n^{j-1}$, solve the PDE (3.35) with $(\tilde{A}^{*^n}, \tilde{B}^{*^n})^j$ obtained from (3.34).
            Get the value vector $\phi_n^j$;
            $j = j + 1$;
        end
        Let $\phi_n = \phi_n^j$, store the optimal controls $(\tilde{A}^{*^n}, \tilde{B}^{*^n}) = (\tilde{A}^{*^n}, \tilde{B}^{*^n})^j$;
    end
    Compute the empirical distribution $\rho_1^k$ from $\rho_0$ with Fokker–Planck equation;
    Compute the gradient vector $\nabla_{\phi_N} \tilde{V}(\phi_N^k) = \left(\frac{\delta C^*(-\phi_N^k)}{\delta \phi_N^k} + \rho_1^k\right) \Delta x$;
    Update $\phi_N^{k+1}$ with Quasi-Newton Method using the gradient information $\nabla_{\phi_N} \tilde{V}(\phi_N^k)$;
    $k = k + 1$;
end
The optimal $\phi_N = \phi_N^k$.
```

as the penalty functional and $F(\tilde{A}, \tilde{B}) = (\tilde{A} - 0.2)^2 + (\tilde{B} - 0.2)^2$ as the cost function. We define the penalty functional as

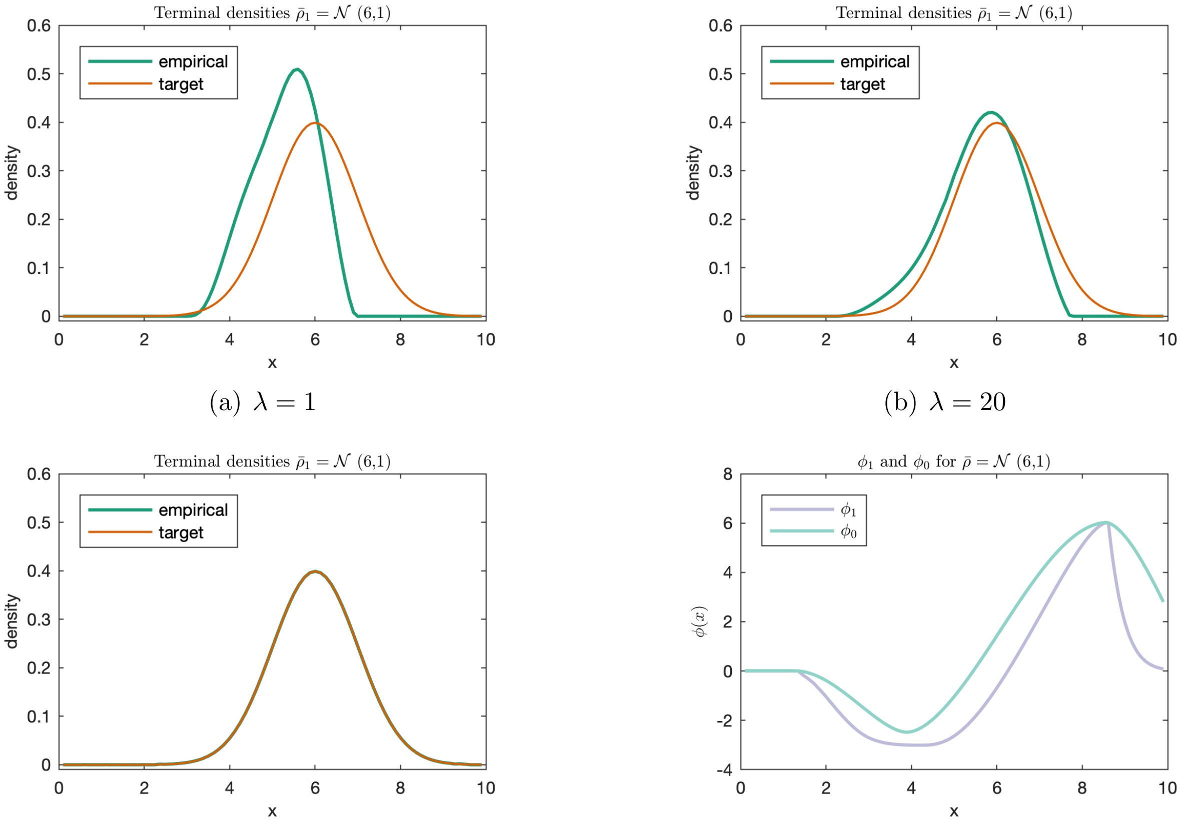
$$C(\rho_1, \overline{\rho}_1) = \frac{\lambda}{2} \int_{\mathbb{R}} (\rho_1 - \overline{\rho}_1)^2 dx,$$

where the parameter λ can be regarded as the intensity of the penalty for the inconsistency. Then the dual problem (3.29) can be expressed explicitly as

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ \int_{\mathbb{R}} -\frac{1}{2\lambda}\phi_1^2 + \phi_1 \overline{\rho}_1 - \rho_0 \phi_0 dx \right\}. \tag{3.47}$$

In this and the following numerical examples, we set the initial wealth $x_0 = 5, \mu = 0.1, \sigma = 0.1$. Figures 3.1(a) and 3.1(b) compare the empirical distribution of the terminal wealth (ρ_1) and the prescribed terminal distribution ($\overline{\rho}_1$) for different intensities λ, where $\overline{\rho}_1 = \mathcal{N}(6, 1)$[1]. We can see that ρ_1 gets closer to $\overline{\rho}_1$ as we increase the intensity of the penalty. In Figure 3.1(c) and 3.1(d), we use the penalty functional (3.31), which is equivalent to letting $\lambda = +\infty$. As shown in Figure 3.1(c), this penalty functional makes ρ_1 attain the target $\overline{\rho}_1$, and the plot 3.1(d) illustrates the optimal function ϕ_1 and the corresponding ϕ_0 we got from Algorithm 4.

[1] We denote $\mathcal{N}(\mu, \sigma)$ a Normal distribution with mean μ and standard deviation σ.



(a) $\lambda = 1$

(b) $\lambda = 20$

(c) infinite penalty

(d) optimal ϕ_1 and ϕ_0

Figure 3.1: Attainable example: $\overline{\rho}_1 = \mathcal{N}(6, 1)$

In Figure 3.2, we plot how the Euclidean distance $\left(\int_{\mathbb{R}}(\rho_1 - \overline{\rho}_1)^2 dx\right)^{\frac{1}{2}}$ changes with respect to λ . As we increase the intensity parameter λ, the Euclidean distance between ρ_1 and $\overline{\rho}_1$ decreases. As λ goes to infinity, the distance asymptotically goes to zero.

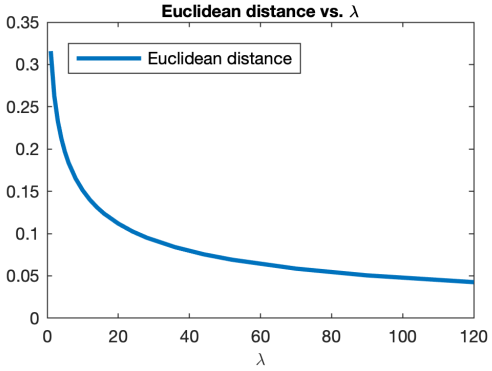


Figure 3.2: Distance metric vs. λ

Compared to other research where the prescribed distributions are restricted to Gaussian, our method applies to a wide choice of $\overline{\rho}_1$, such as heavy-tailed and asymmetric distributions. In Figure 3.3, we illustrate that we can attain the target when $\overline{\rho}_1$ is a mixture of two Normal distributions, in particular

$$\overline{\rho}_1(x) = 0.5\mathcal{N}(4, 1) + 0.5\mathcal{N}(7, 1).$$

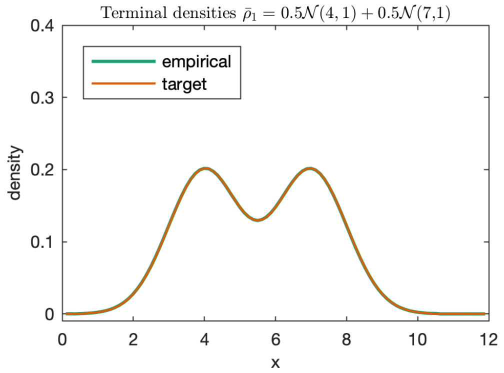


Figure 3.3: mixture of Normal distributions

The Kullback–Leibler (K–L) divergence introduced in Kullback and Leibler (1951), is also known as relative entropy or information deviation. It measures the divergence of the distribution ρ_1 from the target $\bar{\rho}_1$, the more similar the two distributions are, the smaller the relative entropy will be. This measurement is widely used in Machine Learning to compare two densities because it has the following advantages: 1) this function is non-negative; 2) for a fixed distribution $\bar{\rho}_1$, $C(\rho_1, \bar{\rho}_1)$ is convex in ρ_1; 3) $C(\rho_1, \bar{\rho}_1) = 0$ if and only if $\rho_1 = \bar{\rho}_1$ everywhere. There are also caveats to the implementation of this penalty function. We may face $0 \log 0$ or *division by zero* cases in practice; to address this, we can replace zero with an infinitesimal positive value.

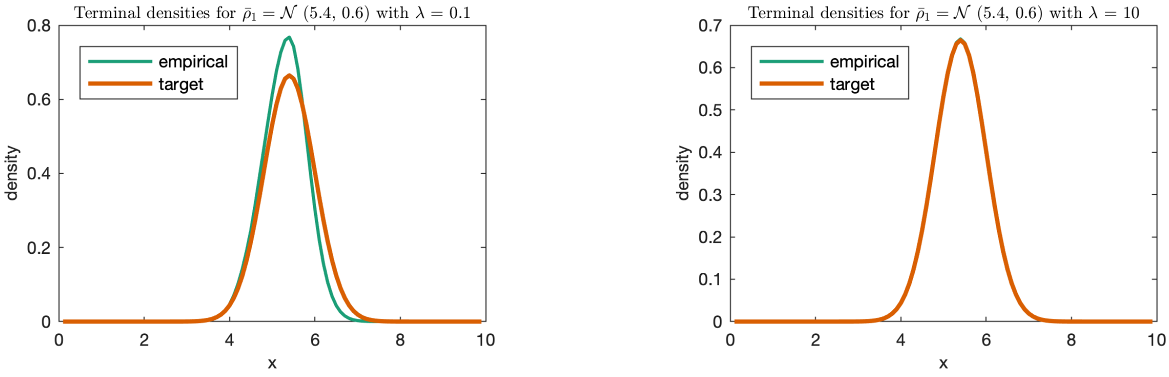
In this case, the penalty functional is defined as

$$C(\rho_1, \bar{\rho}_1) = \int_{\mathbb{R}} \lambda \rho_1(x) \ln \left(\frac{\rho_1(x)}{\bar{\rho}_1(x)} \right) dx, \tag{3.48}$$

and the dual problem (3.29) can be expressed explicitly as

$$V(\rho_0, \bar{\rho}_1) = \sup_{\phi_1} \left\{ - \int_{\mathbb{R}} \lambda \exp\left(-\frac{\phi_1}{\lambda} - 1 \right) \bar{\rho}_1 - \phi_0 \rho_0 dx \right\}.$$

In Figure 3.4, we compare the empirical terminal density ρ_1 and the target $\bar{\rho}_1$ when $C(\rho_1, \bar{\rho}_1)$ is defined by (3.48) and $F(\tilde{A}, \tilde{B}) = (\tilde{A} - 0.2)^2 + (\tilde{B} - 0.2)^2$. The initial wealth $x_0 = 5$ and we set $\lambda = 0.1$ in Figure 3.4(a) and $\lambda = 10$ in Figure 3.4(b).



(a) $\bar{\rho}_1 = \mathcal{N}(5.4, 0.6), \lambda = 0.1$

(b) $\bar{\rho}_1 = \mathcal{N}(5.4, 0.6), \lambda = 10$

Figure 3.4: K–L divergence as the penalty functional

3.5.2 Distribution of the wealth with cash saving

In this section, we consider the cash saving during the investment process. From previous parts, we have the constraint $(\tilde{B}^+)^2 \leq \|\nu(t,x)\|^2 \tilde{A}$. However, when the prescribed target $\overline{\rho}_1$ is not ambitious enough, we will find the optimal drift $\tilde{B}^*$ is not saturated, i.e., $(\tilde{B}^{*+})^2 < \|\nu(t,x)\|^2 \tilde{A}^*$ in (3.34). Our goal in this section is to show that we can reach a better terminal distribution, in the sense that the terminal wealth has a higher expected value, when we take cash saving into account.

Keep using the concept *cash saving* defined in section 3.2.1, we denote $(C_t)_{t\in[0,1]}$ the accumulated cash saving up to time t. Then the evolution of C_t is

$$\begin{aligned} dC_t &= \left(\|\nu(t,x)\| \sqrt{\tilde{A}^*} - \tilde{B}^*\right) dt, \\ C_0 &= 0. \end{aligned}$$

If we add up the cash saving C_t and the portfolio wealth X_t, we can get a new process *wealth with cash saving.* Define $X_t^c := X_t + C_t$, it is obvious to see that X_t^c follows the dynamics

$$\begin{aligned} dX_t^c &= \|\nu(t,x)\| \sqrt{\tilde{A}^*} dt + \tilde{A}^{*\frac{1}{2}} dW_t, \\ X_0^c &= x_0. \end{aligned}$$

Denote by $p(t,x) \in \mathcal{P}(\mathbb{R})$ the distribution of X_t^c at time t, then $p(t,x)$ satisfies the following Fokker–Planck equation

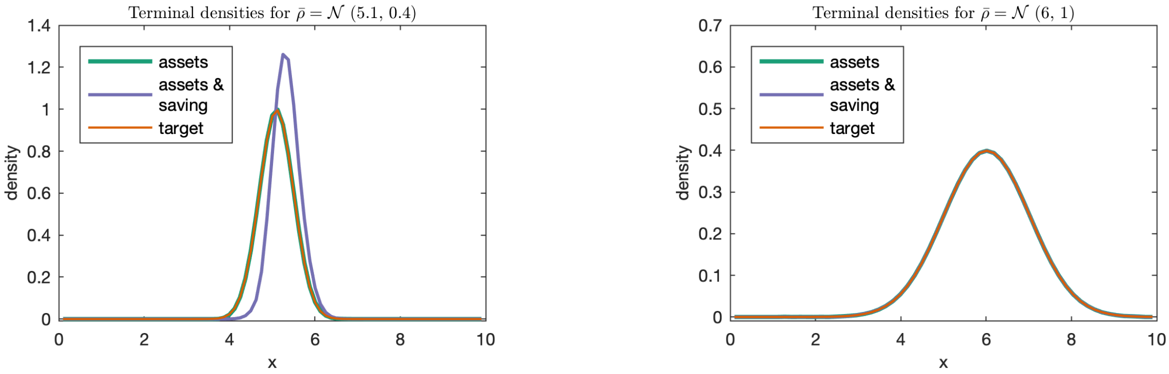
$$\partial_t p + \partial_x \left(\|\nu(t,x)\| \sqrt{\tilde{A}^*} p\right) - \frac{1}{2}\partial_{xx}\left(\tilde{A}^* p\right) = 0,$$
$$p_0(x) = \delta(x - x_0).$$

Therefore, after solving for the optimal $\tilde{A}^*$, $\tilde{B}^*$ according to (3.34) over time, we can find the densities of X_t as well as X_t^c. We keep using the squared Euclidean distance as the penalty functional and $F(\tilde{A}, \tilde{B}) = (\tilde{A} - 0.2)^2 + (\tilde{B} - 0.2)^2$ as the cost function. Figure 3.5 compares the densities for X_1 (terminal wealth), X_1^c (terminal *wealth with cash saving*) and the prescribed target density. In Figure 3.5(a), with a rather conservative target $\overline{\rho}_1 = \mathcal{N}(5.1, 0.4)$, although ρ_1 has attained the target, the distribution for the *wealth with cash saving* gathers at a higher value. When we set a higher target $\overline{\rho}_1 = \mathcal{N}(6, 1)$, as in Figure 3.5(b), we see there is no cash saved in the process since the paths for ρ_1 and p_1 overlapped.

3.5.3 Distribution of the wealth with cash input

As stated in Proposition 3.2.1, we always have $(\tilde{B}^+)^2 \leq \|\nu(t,x)\|^2 \tilde{A}$ for a self-financing portfolio. However, in this section, we remove the constraint $(\tilde{B}^+)^2 \leq \|\nu(t,x)\|^2 \tilde{A}$, and we allow $\tilde{B} \in \mathbb{R}$ instead. Then the part $\left(\tilde{B} - \|\nu(t,x)\| \sqrt{\tilde{A}}\right)^+$ can be interpreted as the extra cash we invest during the process. In this case, theoretically, we can attain any prescribed target distribution as we want (see Tan et al. 2013, Remark 2.3). For the $\overline{\rho}_1$ which is unattainable by the self-financing portfolio, we can now attain it with the help of cash input. However, to limit the use of cash, we design a cost function as follows,

$$F(\tilde{A}, \tilde{B}) = \begin{cases} K(\tilde{B}^2 - \|\nu(t,x)\|^2 \tilde{A}) + w\tilde{A}^2, & \forall \tilde{B} > \|\nu(t,x)\| \sqrt{\tilde{A}}, \\ w\tilde{A}^2, & \forall 0 \leq \tilde{B} \leq \|\nu(t,x)\| \sqrt{\tilde{A}}, \\ l\tilde{B}^2 + w\tilde{A}^2, & \forall \tilde{B} < 0, \end{cases} \tag{3.49}$$



(a) $\overline{\rho}_1 = \mathcal{N}(5.1, 0.4)$

(b) $\overline{\rho}_1 = \mathcal{N}(6, 1)$

Figure 3.5: Compare terminal distributions with or without cash saving

where K, w, l are positive constants. In the cost function (3.49), we use the term $K(\tilde{B}^2 - \|\nu(t,x)\|^2 \tilde{A})$ to penalize the part $\left(\tilde{B} - \|\nu(t,x)\| \sqrt{\tilde{A}}\right)^+$. By varying K, we can control the strength of penalty and hence control the cash input flow. When K is small, we are allowed to put in cash without being penalized excessively. When K is large, we have to pay a high price for the cash input; consequently, the usage is limited. The terms $w\tilde{A}^2$ and $l\tilde{B}^2$ add coercivity to the function to ensure the existence of the solution, we set w, l to be small positive real values.

With the optimal drift $\tilde{B}^* \in \mathbb{R}$ and diffusion $\tilde{A}^* \in \mathbb{R}^+$, the dynamics of the wealth X_t is

$$\begin{aligned} dX_t &= \tilde{B}^* dt + \sqrt{\tilde{A}^*} dW_t, \\ X_0 &= x_0. \end{aligned}$$

If there is no cash input, the maximum drift is $\|\nu(t,x)\| \sqrt{\tilde{A}}$. Denote $(I_t)_{t\in[0,1]}$ the accumulated cash input up to time t, and I_t follows the dynamics

$$\begin{aligned} dI_t &= \left(\tilde{B}^* - \|\nu(t,x)\| \sqrt{\tilde{A}^*}\right)^+ dt, \\ I_0 &= 0. \end{aligned}$$

Define $X_t^I := X_t - I_t$ as the *path without the cash input.* Then the dynamics of X_t^I is

$$\begin{aligned} dX_t^I &= \min\left(\tilde{B}^*, \|\nu(t,x)\| \sqrt{\tilde{A}^*}\right) dt + \sqrt{\tilde{A}^*} dW_t, \\ X_0^I &= x_0. \end{aligned}$$

Let the density of X_t^I be $q(t,x) \in \mathcal{P}(\mathbb{R})$, then $q(t,x)$ follows the following Fokker-Planck equation

$$\partial_t q + \partial_x \left[\min\left(\tilde{B}^*, \|\nu(t,x)\| \sqrt{\tilde{A}^*}\right) q\right] - \frac{1}{2}\partial_{xx}\left(\tilde{A}^* q\right) = 0,$$
$$q_0(x) = \delta(x - x_0).$$

Finally, we can see the effect of cash input by comparing $\rho_1(x)$ and $q_1(x)$.

Attainable target

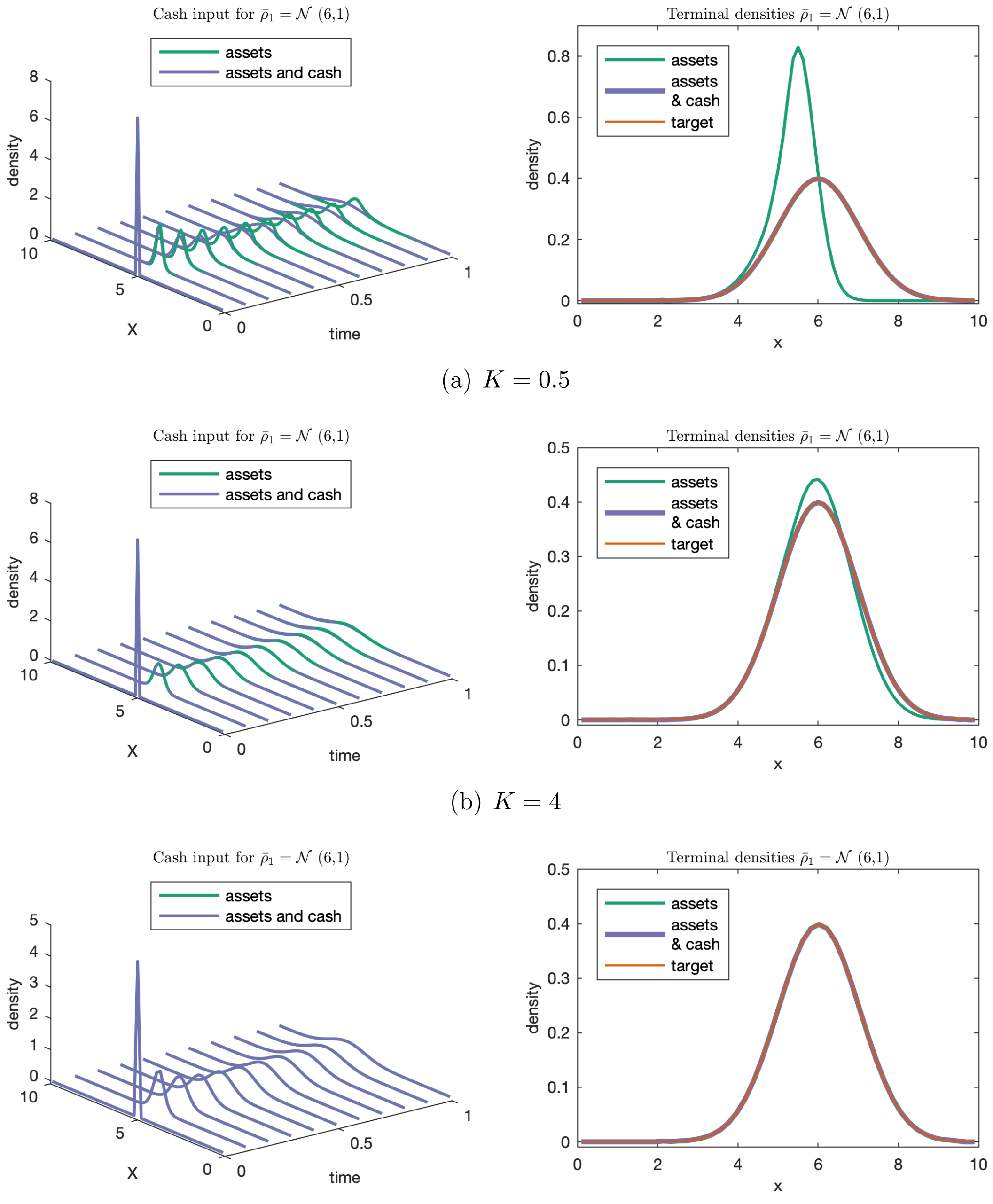
In the first example, we aim at the terminal distribution $\overline{\rho}_1 = \mathcal{N}(6,1)$, which is attainable by the self-financing portfolio. We use the squared Euclidean distance as the penalty functional and equation (3.49) as the cost function. Figure 3.6 demonstrates the time evolution of $q(t,x)$ (assets) and $\rho(t,x)$ (assets and cash input), and it compares $q_1(x), \rho_1(x)$ and $\overline{\rho}_1(x)$ for various K values. At the beginning, we set the coefficient $K = 0.5$ in Figure 3.6(a). There is a clear difference between the paths for *assets* and *assets and cash*, which means we have input a significant amount of cash over time. As we increase the value of K in Figure 3.6(b), the difference between $q(t,x)$ and $\rho(t,x)$ becomes less obvious. When $K = 6$, the paths with or without cash input coincide in Figure 3.6(c) because the high cost has prevented the cash input in this context. Since the target $\mathcal{N}(6,1)$ is attainable, we can still reach it even without cash input, as shown in the second plot of 3.6(c).

Unattainable target

To see the effect of cash input, here we demonstrate an example with an unattainable target. For instance, we may target at a terminal distribution with no left tail but a heavy right tail, in other words, there is very little risk for the wealth to fall below some level. Therefore, we set $\overline{\rho}_1 = \mathit{Weibull}\,(6,2)$ in Figure 3.7(a), where $P(x < 4)$ is almost zero. In this example, the coefficient K in (3.49) is not a constant anymore. Instead, we let $K(t) : [0,1] \to \mathbb{R}^+$ be a function of time so that we can control the cash input flow over time. We define $K(t) = 5$ for $t \in [0, 0.8]$ and $K(t) = 0.1$ for $t \in [0.8, 1]$. In the time-evolution plot (the left one of Figure 3.7(a)), we can see that the paths for *assets* and *assets and cash* start to differentiate from $t = 0.8$. Similarly, we can see the same effect in Figure 3.7(b), where we set $\overline{\rho}_1 = \mathcal{N}(6.5,1)$ and we define $K(t) = 5$ for $t \in [0, 0.95]$ and $K(t) = 0.1$ for $t \in [0.95, 1]$. In these two examples, the targets $\mathit{Weibull}\,(6,2)$ and $\mathcal{N}(6.5,1)$ are unattainable under the constraint $(\tilde{B}^+)^2 \leq \|\nu(t,x)\|^2 \tilde{A}$. However, we can make the empirical terminal density ρ_1 reach $\overline{\rho}_1$ by inputting cash wisely.

3.6 Conclusion

The ability to specify the whole distribution of final wealth of interest as portfolio optimization target gives a greater flexibility over classical objective functions such as expected utility or moment-based objectives such as the mean-variance framework and its extensions. In this chapter, we construct a portfolio and the dynamics of the portfolio wealth is a semimartingale. Starting from an initial wealth, by controlling the portfolio allocation process, we are able to steer the portfolio wealth to a prescribed distribution at the terminal time. This problem is closely related to optimal mass transport (OMT). In the problem formulation, in addition to the conventional cost function in OMT, we also design a penalty functional to measure the divergence of the empirical terminal density from the prescribed one. We take into consideration the possible cash saving during the investment process, and show that we can actually reach a better terminal density when there adding cash saving. When the target density is attainable, our problem can recover the classical OMT problem by choosing an indicator function as the penalty function. When the target terminal density is unattainable by the self-financing portfolio, we devise a strategy to reach it by allowing cash input during the investment process. We proved a duality result for the primal problem and solved it with a gradient descent based algorithm. Our numerical results verify the accuracy and validity of this algorithm.

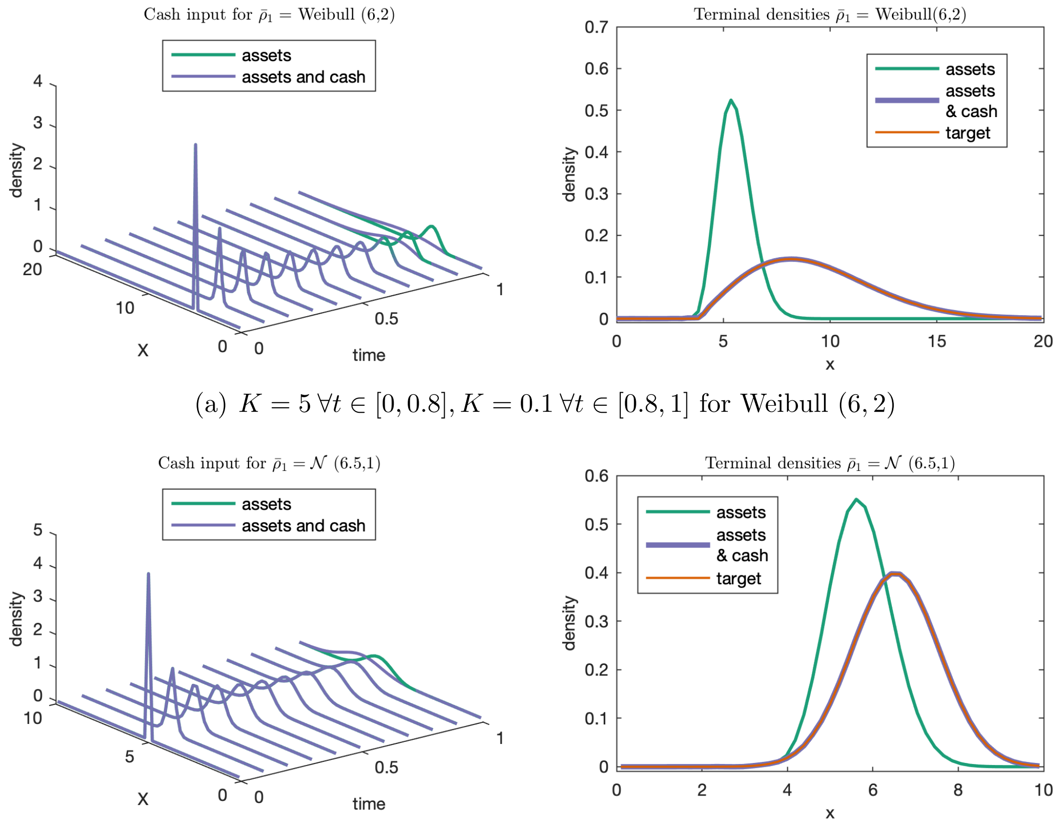


(a) $K = 0.5$

(b) $K = 4$

(c) $K = 6$

Figure 3.6: Fixed K for an attainable target: $\overline{\rho}_1 = \mathcal{N}(6, 1)$



(a) $K = 5\, \forall t \in [0, 0.8], K = 0.1\, \forall t \in [0.8, 1]$ for Weibull $(6, 2)$

(b) $K = 5\, \forall t \in [0, 0.95], K = 0.1\, \forall t \in [0.95, 1]$ for $\mathcal{N}(6.5, 1)$

Figure 3.7: $K(t)$ for unattainable targets

Chapter 4

Deep Semi-Martingale Optimal Transport

We propose two deep neural network-based methods for solving semi-martingale optimal transport problems. The first method is based on a relaxation/penalization of the terminal constraint, and is solved using deep neural networks. The second method is based on the dual formulation of the problem, which we express as a saddle point problem, and is solved using adversarial networks. Both methods are mesh-free and therefore mitigate the *curse of dimensionality.* We test the performance and accuracy of our methods on several examples up to dimension 10. We also apply the first algorithm to a portfolio optimization problem where the goal is, given an initial wealth distribution, to find an investment strategy leading to a prescribed terminal wealth distribution.

4.1 Introduction

The optimal transport problem goes back to the work of Monge (1781) and aims at transporting a distribution μ to another distribution ν under minimum transport cost. It was later revisited by Kantorovich (1942), leading to the so-called Monge–Kantorovich formulation. In recent years, a fast-developing phase was spurred by a wide range of extensions and applications of the Monge–Kantorovich problem; interested readers can refer to the books by Rachev and Rüschendorf (1998), Villani (2003) and Villani (2008) for a comprehensive review. Although we have gained tremendous theoretical insight, the numerical solution of the problem remains challenging. When the dimension is less or equal to three, many state-of-art approaches are able to compute the global solution effectively; see, for example, Chow et al. (2019), Haber and Horesh (2015), Li et al. (2018), and the review by Zhang et al. (2020). Readers can refer to the books by Santambrogio (2015) and Peyré and Cuturi (2019), and the references therein for an overview of these approaches. However, many traditional methods rely on Euclidean coordinates and require spatial discretization. When the distributions live in spaces of dimension four or more, these traditional methods suffer from the curse of dimensionality. Under this situation, solving optimal transport problems using deep neural networks looks very attractive since it can avoid space discretization.

Here is a general definition of Machine Learning:

> *Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.*
>
> — *Arthur Samuel, 1959*

Or a more technical definition is as follows.

> *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P improves with experience E.*
>
> *— Tom Mitchell, 1997*

The first application of Machine Learning was the *spam filter* back in the 1990s. Based on different criteria, Machine Learning can be classified into broad categories: supervised, unsupervised learning, semisupervised learning, and reinforcement learning; online versus batch learning; instance-based versus model-based learning. We call the examples the system uses to learn a training set, each training example in the training set is called a training instance. In a classification task, we usually use *accuracy* to measure the training performance, which is the ratio of correctly classified cases.

Machine Learning has many advantages compared to traditional programming. When a problem is complicated, traditional programming can be long and complex. If the environment of the problem changes, the programmer has to adjust the code accordingly; hence it is hard to maintain. Some problems are even too complex for the programmers to code, e.g., speech recognition. In contrast, Machine Learning techniques can automatically detect the pattern. It can update itself with the latest available information. In this case, the program will be easier to write and maintain. By digging into large amounts of data, Machine Learning can help humans discover previously unknown patterns.

Inspired by the biological neurons in human's brain, a new Machine Learning model emerged – artificial neural networks (ANNs). ANNs are at the core of Deep Learning. They were first introduced by McCulloch and Pitts (1943). There was a revival of interest in ANNs since the 1990s due to the tremendous increase in computing power and a huge quantity of available data. A *Perceptron* is the simplest ANN architecture; when stacking multiple Perceptions, the ANN is called a *Multilayer Perception*. A Multilayer Perceptron consists of one input layer, one or more hidden layers, and an output layer. A Deep Neural Network (DNN) is an ANN with two or more hidden layers. Thanks to the groundbreaking *backpropagation algorithm* introduced by Rumelhart et al. (1985), we can train DNNs successfully and efficiently. Mathematically, this algorithm is just Gradient Descent. But it uses an efficient technique to compute the gradient automatically, this is also referred to as *autodiff*. DNNs are preferred to shallow networks in general. As shown in Liang and Srikant (2016), for a given degree of approximation error, a shallow network will need exponentially more neurons than a deep network.

Usually the calibration of an ANN is done by minimizing a loss function. When the loss function is an expectation, Stochastic Gradient Descent (SGD) is a natural adaptation of Gradient Descent to stochastic optimization problems. The DNN-SGD paradigm is well suited for solving scientific computing problems such as stochastic control problems (e.g., Han and E 2016, Huré et al. 2018, Bachouch et al. 2018b) or partial differential equations (PDEs) in very high dimension (e.g., Weinan et al. 2017, Sirignano and Spiliopoulos 2018 and Huré et al. 2019), thanks to its ability to overcome the curse of dimensionality. A comprehensive review of the numerical and theoretical advances in solving PDE and BSDE with deep learning algorithms can be found in Han and Jentzen (2020).

Recent years witnessed the emergence of research on solving optimal transport problems with neural networks. Ruthotto et al. (2020) used a residual network (ResNet) to approximately solve high-dimensional mean-field games by combining Lagrangian and Eulerian viewpoints. In the numerical experiment, they solved a dynamical optimal transport as a potential mean-field game. Henry-Labordère (2019) introduced Lagrange multipliers associated with the two marginal constraints and proposed a Lagrangian algorithm to solve martingale optimal transport using neural networks. However, their algorithm will be applicable only to cost functions satisfying a martingale twist condition. Eckstein and Kupper (2019) studied optimal transport problem with an inequality constraint; their algorithm penalizes the optimization problem in its dual representation.

The main application of this work is in finance. The use of optimal transport in finance has been very popular in the recent years, and our work is particularly relevant for the problem of model calibration via optimal transport as in Guo et al. (2019c); Guo and Loeper (2018); Guo et al. (2019b, 2020b).

In this chapter, we study optimal transport by semi-martingales as introduced in Tan et al. (2013) and we use deep learning to estimate the optimal drift and diffusion coefficients, by two different methods:

- In the first one, we relax the terminal constraint by adding a penalty term to the loss function and solve it with a deep neural network. [1]
- In the second one, we introduce the dual formulation of the problem and express it as a saddle point problem. In this case, we utilize adversarial networks to solve it.

These methods can be widely applied to solving optimal transport as well as stochastic optimal control problems. The two methods do not need spatial discretization and hence can be potentially used for high-dimensional problems. We illustrate our method with an application in finance, where we implement the first algorithm to the problem of optimal portfolio selection with a prescribed terminal density studied in Guo et al. (2020a).

This chapter is organized as follows. In section 4.2, we formulate the optimal transport problem and introduce the primal problem, adding a penalization to the expected cost function. To solve this problem, we propose a deep neural network-based algorithm and present the corresponding numerical results in section 4.3. In section 4.4, we provide the dual representation of the primal problem and express it as a saddle point problem. Then we devise an algorithm using adversarial networks. We illustrate the numerical results in sections 4.4.1 and 4.4.2, including a 10-dimensional example. Finally, in section 4.5, we implement the above-mentioned (mentioned in section 4.3) deep learning algorithm in a financial application, solving the problem of steering the portfolio wealth to a prescribed terminal density.

4.2 Problem formulation

In this chapter, on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we consider a stochastic process $(X_t)_{t\in[0,1]}$ valued in $\mathbb{R}^d$, that solves the SDE

$$dX_t = B(t, X_t)dt + A(t, X_t)dW_t, \tag{4.1}$$

$$X_0 = x_0, \tag{4.2}$$

where $B : \mathcal{E} \to \mathbb{R}^d$ and $A : \mathcal{E} \to \mathbb{R}^{d\times d}$ is defined such that $AA^\intercal = \mathcal{A}$.

We denote by $\rho_t := \mathbb{P} \circ X_t^{-1} \in \mathcal{P}(\mathbb{R}^d)$ the distribution of X_t. In this problem, we are given the initial distribution of the state variable ρ_0, where ρ_0 can be a Dirac measure. and a prescribed terminal distribution $\bar{\rho}_1 \in \mathcal{P}(\mathbb{R}^d)$. We define a convex cost function $F : \mathbb{R}^d \times \mathbb{S}^d \to \mathbb{R}^+ \cup \{+\infty\}$ where $F(B, \mathcal{A}) = +\infty$ if $\mathcal{A} \notin \mathbb{S}^d_+$.

[1] This relaxation can be useful in the case where, depending on the constraints we put on the stochastic evolution, not all distributions are attainable, think for example of the case where we impose the process to be a martingale

For shorthand, we write $A_t(x)$ for $A(t,x)$ and $B_t(x)$ for $B(t,x)$. Given ρ_0 and suitable processes $(A_t)_{t\in[0,1]}, (B_t)_{t\in[0,1]}$, the distribution of X_1 is $\rho_1 := \mathbb{P} \circ X_1^{-1}$. We introduce the penalty function $C(\rho_1, \overline{\rho}_1)$, whose role will be to penalize the deviation of ρ_1 from the target $\overline{\rho}_1$.

Now we are interested in the following minimization problem:

Problem 4.2.1. With a given initial distribution ρ_0 and a prescribed terminal distribution $\overline{\rho}_1$, we want to solve the infimum of the functional

$$V(\rho_0, \overline{\rho}_1) = \inf_{\rho, B, \mathcal{A}} \left\{ \int_{\mathcal{E}} F(B_t, \mathcal{A}_t) d\rho(t,x) + C(\rho_1, \overline{\rho}_1) \right\}, \tag{4.3}$$

over all $(\rho, B, \mathcal{A}) \in \mathcal{P}(\mathbb{R}^d) \times \mathbb{R}^d \times \mathbb{S}^d$ satisfying the initial distribution

$$\rho(0,x) = \rho_0(x) \quad \forall x \in \mathbb{R}^d, \tag{4.4}$$

and the Fokker–Planck equation

$$\partial_t \rho(t,x) + \nabla_x \cdot (B(t,x)\rho(t,x)) - \frac{1}{2} \sum_{i,j} \partial_{ij} (\mathcal{A}_{i,j}(t,x)\rho(t,x)) = 0 \quad \forall (t,x) \in \mathcal{E}. \tag{4.5}$$

Because the objective function (4.3) is a trade-off between the convex cost function $F : \mathbb{R}^d \times \mathbb{S}^d \to \mathbb{R}$ and the penalty C, the optimal $\rho, \mathcal{A}, B$, if it exists, will not in general ensure that $\rho_1 = \overline{\rho}_1$, unless the penalty function is

$$C(\rho_1, \overline{\rho}_1) = \begin{cases} 0 & \text{if } \rho_1 = \overline{\rho}_1, \\ +\infty & \text{if } \rho_1 \neq \overline{\rho}_1, \end{cases} \tag{4.6}$$

and one recovers the "usual" semi-martingale optimal transport problem.

We make the following assumptions which will hold throughout the paper.

Assumption 4.2.1. The penalty function $C(\cdot, \overline{\rho}_1) : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}^+$ is convex, lower semi-continuous with respect to the weak-$*$ convergence of ρ_1, and $C(\rho_1, \overline{\rho}_1) = 0$ if and only if $\rho_1 = \overline{\rho}_1$.

Assumption 4.2.2. The cost function $F(p,q)$ is non-negative, lower semi-continuous and strictly convex in (p,q).

With Assumptions 4.2.1 and 4.2.2, we can get the following existence and uniqueness result for the minimizer of Problem 4.2.1 using similar convex minimization techniques as in Loeper (2006, Proposition 2). We refer the interested readers to Loeper (2006) for the detailed proof.

Theorem 4.2.1. Let us define

$$I(\rho, B, \mathcal{A}) = \int_{\mathcal{E}} F(B_t, \mathcal{A}_t) d\rho(t,x) + C(\rho_1, \overline{\rho}_1).$$

Under Assumptions 4.2.1 and 4.2.2, if $I(\rho, B, \mathcal{A})$ is finite, then there exists a unique minimizer $(\rho, B, \mathcal{A}) \in \mathcal{P}(\mathbb{R}^d) \times \mathbb{R}^d \times \mathbb{S}^d$ for Problem 4.2.1 satisfying constraints (4.4) and (4.5).

4.3 Deep neural network-based algorithm

In this section, we devise an algorithm to solve Problem 4.2.1 by using deep learning. Note that given processes $(B_t)_{t\in[0,1]}$, $(A_t)_{t\in[0,1]}$ and ρ_0, the density process $(\rho_t)_{t\in(0,1]}$ is fully determined (up to suitable regularity assumptions on B and A). Hence, our goal is to use neural networks to search for the optimal $(B_t)_{t\in[0,1]}$ and $(A_t)_{t\in[0,1]}$ which minimize the objective function.

We discretize the period $[0,1]$ into N constant time steps and construct a neural network θ_n for each time step $n \in [0, N-1]$, let X_n denote the state variable at time step n. We use multilayer feedforward neural networks in our application. At each time step, we wish to approximate the drift and diffusion coefficient with a neural network, i.e., $(B_n, A_n) \approx \theta_n(X_n)$, where $B_n \in \mathbb{R}^d$, $A_n \in \mathbb{R}^{d\times d}$. Feedforward neural networks approximate complicated nonlinear functions by a composition of simpler functions, namely

$$(B_n, A_n) \approx \theta_n(X_n) = g_n^J \circ g_n^{J-1} \circ \cdots \circ g_n^1(X_n).$$

For each layer, g_n^j is

$$g_n^j(X_n) = \sigma^j\left(X_n \mathbf{W}_n^j + \mathbf{b}_n^j\right), \tag{4.7}$$

where $\mathbf{W}_n, \mathbf{b}_n$ are the weight matrices and bias vectors, respectively. Here, $\sigma^j(\cdot)$ is a component-wise nonlinear activation function, such as sigmoid, ReLU, tanh, etc. In our paper, we use Leaky ReLU for all the hidden layers. For the output layer, $\sigma^j(\cdot)$ is the identity function.

We denote by M the number of Monte Carlo paths. For a particular path $m \in [1, M]$, with B_n^m, A_n^m and X_n^m, we can compute the state variable in the next time step from the dynamics

$$X_{n+1}^m = X_n^m + B_n^m \Delta t + A_n^m \Delta W_n^m, \quad \forall m \in [1, M], n \in [0, N-1].$$

At the final time step N, we can get M samples of terminal wealth X_N from the Monte Carlo paths. With these samples, we can estimate the empirical terminal distribution ρ_1 using kernel density estimation (KDE). In particular, we use the Gaussian kernel $K_{\mathbf{H}}(x) = (2\pi)^{-d/2}|\mathbf{H}|^{-\frac{1}{2}}e^{-\frac{1}{2}x^\intercal \mathbf{H}^{-1}x}$ with an appropriate bandwidth matrix $\mathbf{H} \in \mathbb{S}^d$. Then the terminal density is estimated as $\tilde{\rho}_1(x) := \frac{1}{M}\sum_{m=1}^{M} K_{\mathbf{H}}(x - X_N^m)$. Because the kernel density estimation is not an unbiased estimator of the true density, an error is generated from estimating ρ_1 with $\tilde{\rho}_1$. To address this issue, we also estimate $\bar{\rho}_1$ with the same KDE and use the estimated $\tilde{\bar{\rho}}_1$ as the target density in the training. To be precise, we first generate a sample $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_{M'})$ of size M' from the target density $\bar{\rho}_1(x)$. Then we estimate $\bar{\rho}_1(x)$ with $\tilde{\bar{\rho}}_1(x) := \frac{1}{M'}\sum_{i=1}^{M'} K_{\mathbf{H}}(x - \tilde{x}_i)$.

Finally, the objective function in (4.3) can be naturally used as the loss function in the training,

$$L(\theta_{n,n\in[0,N-1]}) = \mathbb{E}\left[\sum_{n=0}^{N-1} F(\theta_n(X_n))\Delta t\right] + C(\tilde{\rho}_1, \tilde{\bar{\rho}}_1).$$

Then the training will search for the optimal neurons $\hat{\theta}_{n,n\in[0,N-1]}$ where

$$\hat{\theta}_{n,n\in[0,N-1]} = \arg\inf_{\theta}\left\{\frac{1}{M}\sum_{m=1}^{M}\left\{\sum_{n=0}^{N-1} F(\theta_n(X_n^m))\Delta t\right\} + C(\tilde{\rho}_1, \tilde{\bar{\rho}}_1)\right\}.$$

We depict the above process in a flowchart (Figure 4.1), and the complete Deep Neural Network algorithm for Problem 4.2.1 is stated in Algorithm 5.

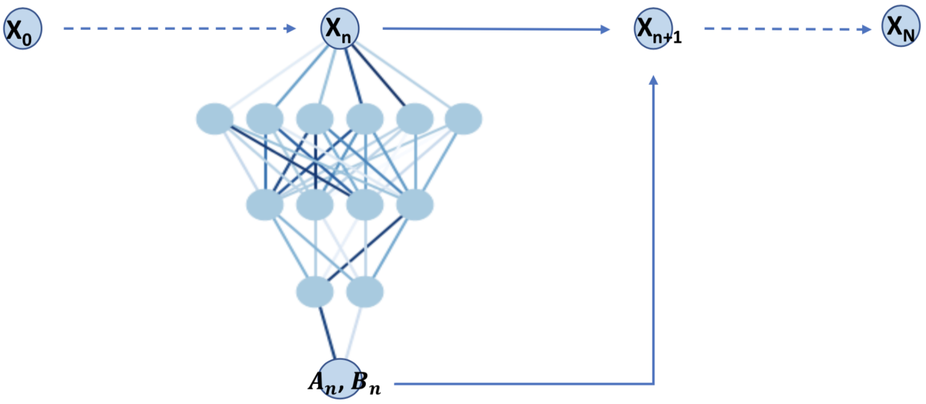


Figure 4.1: Structure of the DNN for one path

Algorithm 5 Deep Neural Network for Problem 4.2.1

```
Starting from the initial condition $X_0^m = x_0\,\forall m \in [1, M]$.
for $epoch = 1 : 100$ do
    for $n = 0 : N - 1$ do
        With neurons $\theta_n$, input $X_n^m$, output $(B_n^m, A_n^m) = \theta_n(X_n^m)$;
        $X_{n+1}^m = X_n^m + B_n^m \Delta t + A_n^m \Delta W_n^m$;
    end
    Estimate the terminal distribution of $X_N$ with KDE as $\tilde{\rho}_1$;
    Estimate $\tilde{\overline{\rho}}_1$ from $\overline{\rho}_1$ with the same KDE;
    Define Loss function $L = \frac{1}{M}\sum_{m=1}^{M}\left\{\sum_{n=0}^{N-1} F(B_n^m, A_n^m(A_n^m)^\intercal)\Delta t\right\} + C(\tilde{\rho}_1, \tilde{\overline{\rho}}_1)$;
    Train the neurons and update $\theta_{n,n\in[0,N-1]}$;
end
Get the optimal $\hat{\theta}_{n,n\in[0,N-1]}$.
```

4.3.1 Numerical Results

In this section, we validate Algorithm 5 with an example where

$$F(B, \mathcal{A}) = \|B\|^2, \tag{4.8}$$

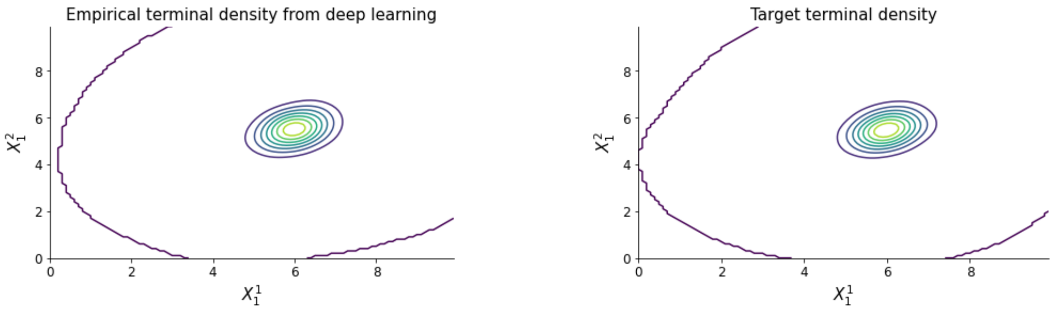
$$C(\rho_1, \overline{\rho}_1) = \frac{\lambda}{2}\int_{\mathbb{R}} (\rho_1 - \overline{\rho}_1)^2 dx. \tag{4.9}$$

The parameter λ in equation (4.9) is the intensity of the penalization. To push ρ_1 as close as possible to the target $\overline{\rho}_1$, we want λ be to large and we will let $\lambda = 5000$ in this case. We start from an initial state $x_0 = \begin{bmatrix} 5.0 \\ 5.0 \end{bmatrix}$, and the target distribution set to be a bivariate normal $\overline{\rho}_1 = \mathcal{N}\left(\begin{bmatrix} 5.5 \\ 6.0 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.10 \\ 0.10 & 0.25 \end{bmatrix}\right)$.

In the training, we construct a 5-layer network with 100, 80, 60, 60 and 40 neurons respectively. The output of the network is of size 2 × 3, out of which the 2 × 1 column represents the drift B_n and the 2 × 2 matrix represents the diffusion coefficient A_n. We use a batch size of 2000 and the Adam optimizer with a learning rate 1×10^{-4}. We trained the network for 100 epochs in total.

From the training, we get an empirical terminal density ρ_1 with mean $\begin{bmatrix}5.509\\5.994\end{bmatrix}$ and covariance matrix $\begin{bmatrix}0.253 & 0.098\\0.098 & 0.246\end{bmatrix}$. We can see that the mean and covariance of the empirical terminal density are very close to the target ones.

The contours of the empirical distribution and the target distribution are presented in Figure 4.2. In the later section, we will introduce a metric to measure the performance of the trained dataset.



(a) Density of X_1 from Algorithm 5

(b) Target Density of X_1

Figure 4.2: Contours with a bivariate normal target distribution

4.3.2 Merged network

The architecture described in Algorithm 5 consists of training N different feedforward neural networks (one per time step). This architecture generates a possibly high number of weights and biases to be estimated. Another possibility is to use one single feedforward neural network for all time steps. At each time step, besides the state variable X_n, we also feed the time n into the network. We refer to this alternative architecture as a *feedforward merged network*. The training process for this merged network is illustrated in Figure 4.3, and the algorithm is summarized in Algorithm 6. The advantage of this architecture is that, instead of training a list of networks $\theta_{n,n\in[0,N-1]}$, we only need to train one network θ in this algorithm, which significantly reduces the number of neurons and the complexity of the training.

The numerical experiments give us a similar result to the one from Algorithm 5; hence we are not presenting them repeatedly. When using the merged structure, as we observed from the experiment, each iteration costs 3.2 seconds, and it takes 190 epochs (each epoch contains 100 iterations) to reach the final value. When using Algorithm 5, each iteration takes 7.5 seconds and it needs 100 epochs to converge to the final value.

Algorithm 6 Deep Neural Network for Problem 4.2.1 with a merged network

```
Starting from the initial condition $X_0^m = x_0 \,\forall m \in [1, M]$.
for $epoch = 1 : 100$ do
    for $n = 0 : N-1$ do
        With neurons $\theta$, input $X_n^m$ and $n$, output $B_n^m, A_n^m = \theta(X_n^m, n)$;
        $X_{n+1}^m = X_n^m + B_n^m \Delta t + A_n^m \Delta W_n^m$;
    end
    Estimate the terminal distribution of $X_N$ with KDE as $\tilde{\rho}_1$;
    Estimate $\tilde{\bar{\rho}}_1$ from $\bar{\rho}_1$ with the same KDE;
    Loss function $L = \frac{1}{M}\sum_{m=1}^{M}\left\{\sum_{n=0}^{N-1} F(B_n^m, A_n^m (A_n^m)^\intercal)\Delta t\right\} + C(\tilde{\rho}_1, \tilde{\bar{\rho}}_1)$;
    Train the neurons with the Adam optimizer and update $\theta$;
end
Get the optimal $\hat{\theta}$.
```

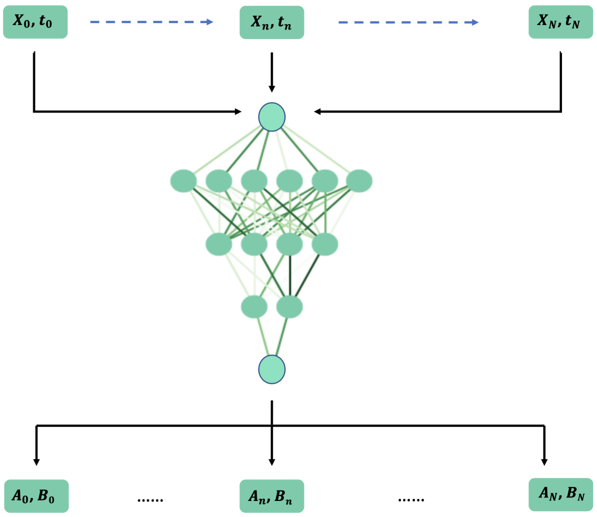


Figure 4.3: Structure of the merged network

4.4 Adversarial network algorithm for the dual problem

In this section, we first introduce the dual formulation of Problem 4.2.1 and express it as a saddle point problem. Then we propose an Adversarial Network-based algorithm to solve it, where we demonstrate high-dimensional numerical examples. This algorithm is inspired by Generative Adversarial Networks (GANs), which were first introduced in Goodfellow et al. (2014). GANs have enjoyed great empirical success in image generating and processing. The principle behind GANs is to interpret the process of generative modeling as a competing game between two (deep) neural networks: a generator and a discriminator. The generator network attempts to generate data that looks similar to the training data, and the discriminator network tries to identify whether the input sample is faked or true. GANs recently have attracted interests in the finance field, including simulating financial time-series data (e.g., Wiese et al. 2020, Zhang et al. 2019a) and for asset pricing models (Chen et al. 2019a).

To recover the optimal transport problem, we can use an indicator function (equation

(4.6)) as the penalty functional. Now we present the duality result for Problem 4.2.1. The following theorem is stated in Tan et al. (2013).

Theorem 4.4.1. When $C(\rho_1, \overline{\rho}_1)$ is defined as in equation (4.6), there holds

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{ \int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1 - \phi_0 d\rho_0 \right\}, \tag{4.10}$$

where the supremum is running over all $\phi_1 \in C_b^2(\mathbb{R}^d)$ and ϕ_0 is a viscosity solution of the Hamilton–Jacobi–Bellman equation

$$\begin{cases} -\partial_t \phi - \sup_{B \in \mathbb{R}^d, \mathcal{A} \in \mathbb{S}^d} \Big[B \cdot \nabla_x \phi + \frac{1}{2} tr(\mathcal{A} \nabla_x^2 \phi) - F(B, \mathcal{A}) \Big] = 0, & \text{in } [0,1) \times \mathbb{R}^d, \\ \phi(1, x) = \phi_1(x), & \text{on } [1] \times \mathbb{R}^d. \end{cases} \tag{4.11}$$

The function $\phi(0, x)$ can be expressed as

$$\phi(0, x) = \sup_{B \in \mathbb{R}^d, \mathcal{A} \in \mathbb{S}^d} \mathbb{E} \left[\phi_1(X_1) - \int_0^1 F(B, \mathcal{A}) dt \,\middle|\, X_0 = x_0 \right]. \tag{4.12}$$

Starting from an initial state $x_0 \in \mathbb{R}^d$, the initial density $\rho_0(x) = \delta(x - x_0)$, hence $\int_{\mathbb{R}^d} \phi_0 d\rho_0 = \phi_0(x_0)$. Then we can substitute the expression (4.12) into the dual form (4.10), and the dual formulation can be written as a saddle point problem:

$$V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \inf_{B \in \mathbb{R}^d, \mathcal{A} \in \mathbb{S}^d} \left\{ \int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1 - \mathbb{E} \left[\phi_1(X_1) - \int_0^1 F(B, \mathcal{A}) dt \,\middle|\, X_0 = x_0 \right] \right\}. \tag{4.13}$$

This dual saddle point formulation of the optimal transport problem is reminiscent of GANs: GANs can be interpreted as minimax games between the generator and the discriminator, whereas our problem is a minimax game between ϕ_1 and $(\mathcal{A}, B)$.

Inspired by this connection, we can use Adversarial Networks to estimate the value (4.13). GANs are also applied in Guo et al. (2019a) to solve a robust portfolio allocation problem. Our Adversarial Network consists of two neural networks; one generates ϕ_1 (referred to as ϕ_1-generator, denoted by Φ), the other generates A and B (referred to as AB-generator, denoted by θ). The two networks are trained iteratively: In the first phase, we train the AB-generator with a loss function

$$L_1 = -\mathbb{E} \left[\phi_1(X_1) - \int_0^1 F(B, \mathcal{A}) dt \,\middle|\, X_0 = x_0 \right].$$

In the second phase, given the output X_1 from the AB-generator, we train the ϕ_1-generator with a loss function

$$L_2 = -\int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1 + \mathbb{E} \left[\phi_1(X_1) - \int_0^1 F(B, \mathcal{A}) dt \,\middle|\, X_0 = x_0 \right].$$

When the dimension $d \geq 3$, the bottleneck in this algorithm is to compute $\int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1$ efficiently. To address this issue, we can use Monte Carlo integration. In particular, we sample $\mathfrak{I}$ points $\overline{x}_i \in \mathbb{R}^d$ $(i \in [1, \mathfrak{I}])$ from the target distribution $\overline{\rho}_1$. Then we can estimate $\int_{\mathbb{R}^d} \phi_1 d\overline{\rho}_1$ with $\frac{1}{\mathfrak{I}} \sum_{i=1}^{\mathfrak{I}} \phi_1(\overline{x}_i)$. In this case, this Adversarial Network-based scheme is mesh-free. It can now avoid the curse of dimensionality and has the potential to be applied to high-dimensional problems.

Algorithm 7 Adversarial Network algorithm with a merged network for the dual problem

```
Sample $\mathcal{I}$ points $\overline{x}_i \in \mathbb{R}^d$ $(i \in [1, \mathcal{I}])$ from the target distribution $\overline{\rho}_1$.
Starting from the initial condition $X_0^m = x_0 \, \forall m \in [1, M]$:
for epoch = 1 : number of epochs do
    Phase 1: train the $AB$-generator
    for time step $n = 1 : N - 1$ do
        With the network $\theta$, input $\{X_n^m\}_{m=1}^M$ and time step $n$, outputs $(B_n^m, A_n^m) = \theta(X_n^m, n)$;
        $X_{n+1}^m = X_n^m + B_n^m \Delta t + A_n^m \Delta W_n$;
    end
    Loss function $L_1 = -\frac{1}{M}\sum_{m=1}^M \left[\Phi(X_N^m) - \sum_{n=0}^{N-1} F(B_n^m, A_n^m (A_n^m)^\intercal)\Delta t\right]$;
    Train the neurons with the Adam optimizer and update $\theta$.

    Phase 2: train the $\phi_1$-generator
    With the network $\Phi$, input $\{X_N^m\}_{m=1}^M$, output $\phi_1(X_N^m) = \Phi(X_N^m)$;
    With the same network $\Phi$, inputs $\overline{x}_i, \forall i \in [1, \mathcal{I}]$, outputs $\Phi(\overline{x}_i)$;
    Loss function $L_2 = -\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \Phi(\overline{x}_i) + \frac{1}{M}\sum_{m=1}^M \left[\Phi(X_N^m) - \sum_{n=0}^{N-1} F(B_n^m, A_n^m (A_n^m)^\intercal)\Delta t\right]$;
    Train the neurons with the Adam optimizer and update $\Phi$.
end
```

We use a merged network for the AB-generator in this algorithm. A demonstration of the training process is illustrated in Figure 4.4. The detailed algorithm is summarized in Algorithm 7.

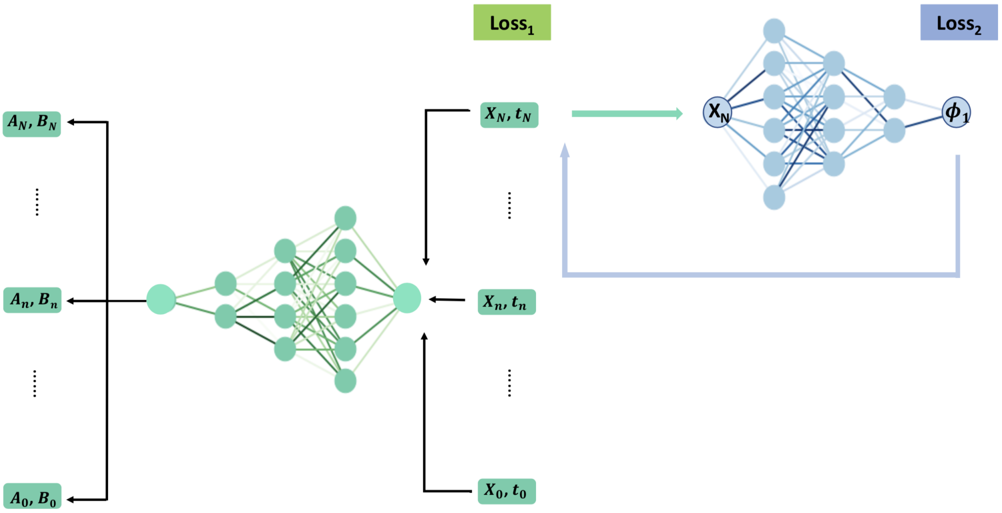
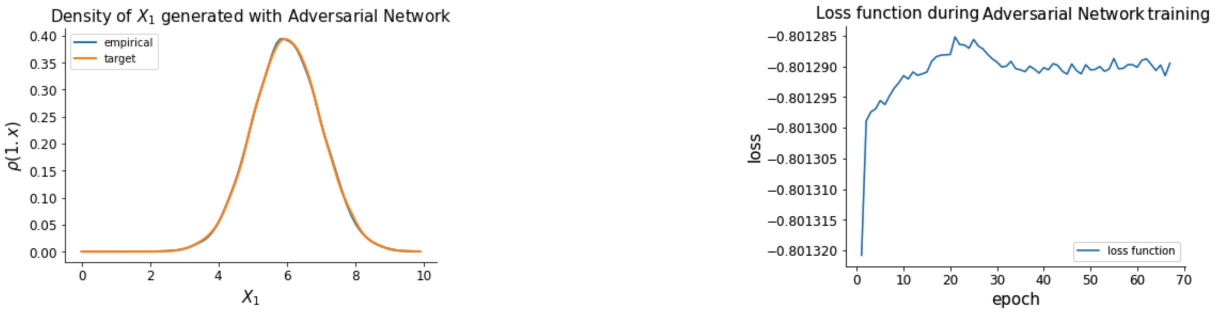


Figure 4.4: Adversarial Network algorithm with a merged network for the dual problem

4.4.1 Numerical Results

First, we start with a simple one-dimensional example to assess the quality of Algorithm 7. We choose a target distribution $\overline{\rho}_1 = \mathcal{N}(6, 1)$, a cost function $F(B, \mathcal{A}) = (\mathcal{A} - 0.1)^2$ and solve for $V(\rho_0, \overline{\rho}_1) = \sup_{\phi_1} \left\{\int_{\mathbb{R}} \phi_1 d\overline{\rho}_1 - \phi_0 d\rho_0\right\}$.

In Figure 4.5(a), we plot the target distribution and the distribution of X_1 learnt by the Adversarial Networks; in Figure 4.5(b), we show the corresponding loss function during the training. We can see that our algorithm works well in terms of attaining the target density.



(a) Density of X_1 learnt by Adversarial Networks

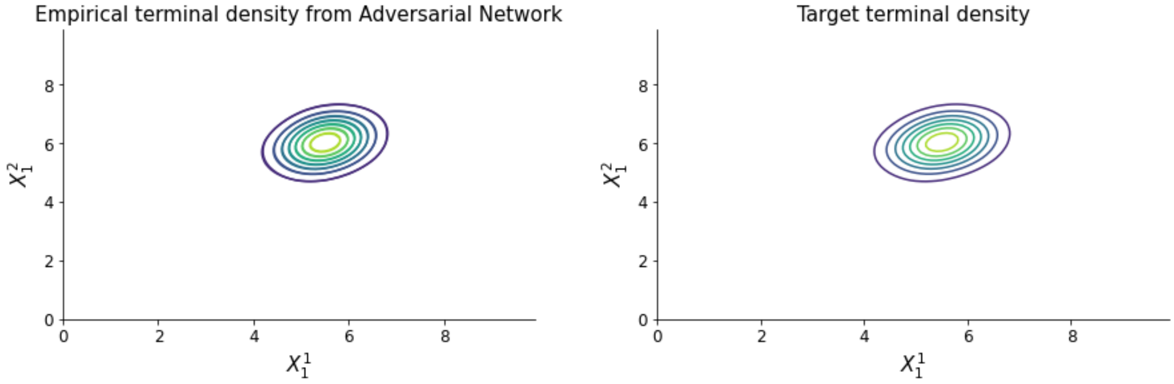
(b) Loss function

Figure 4.5: Dimension $d = 1$, $\overline{\rho}_1 = \mathcal{N}(6, 1)$

Next, we demonstrate a two-dimensional example. We let $X_n \in \mathbb{R}^2$, and we generate the drift $B_n \in \mathbb{R}^2$ and the diffusion coefficient $A_n \in \mathbb{R}^{2\times 2}$ with the deep neural network. We again start from an initial state $x_0 = \begin{bmatrix} 5.0 \\ 5.0 \end{bmatrix}$, and let the cost function $F(B, \mathcal{A}) = \|B\|^2$ and the target distribution set to be a bivariate normal $\overline{\rho}_1 = \mathcal{N}\left(\begin{bmatrix} 5.5 \\ 6.0 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.10 \\ 0.10 & 0.25 \end{bmatrix}\right)$.

We construct two networks for (A, B) and ϕ_1, respectively. The AB-generator has 4 layers with 40, 30, 20 and 10 neurons respectively and the ϕ_1-generator has 4 layers with 80, 60, 40 and 40 neurons respectively. We use a mini-batch SGD algorithm with a batch size of 1000 and Adam optimizer with learning rate 1×10^{-4}.

The empirical terminal density ρ_1 after the training has mean $\begin{bmatrix} 5.497 \\ 6.007 \end{bmatrix}$ and covariance matrix $\begin{bmatrix} 0.253 & 0.099 \\ 0.099 & 0.251 \end{bmatrix}$, which are very close to the target mean and covariance. The contours of the empirical and target distributions are shown in Figure 4.6. This result is similar to the one we got from Algorithm 5.



(a) Contour of ρ_1 from Adversarial Network

(b) Contour of $\overline{\rho}_1$

Figure 4.6: Dimension $d = 2$, bivariate normal target distribution

4.4.2 Higher dimensional examples

In this section, we apply the Adversarial Network-based algorithm to high-dimensional examples. When the dimension $d \geq 3$, Algorithm 5 will be less effective. This is due to how the kernel density estimation error increases with the dimension (see for example Wand and Jones 1994, Chapter 4). By contrast, the Adversarial Network-based algorithm for the dual problem does not rely on kernel density estimation, and is therefore less affected by an increase in dimension.

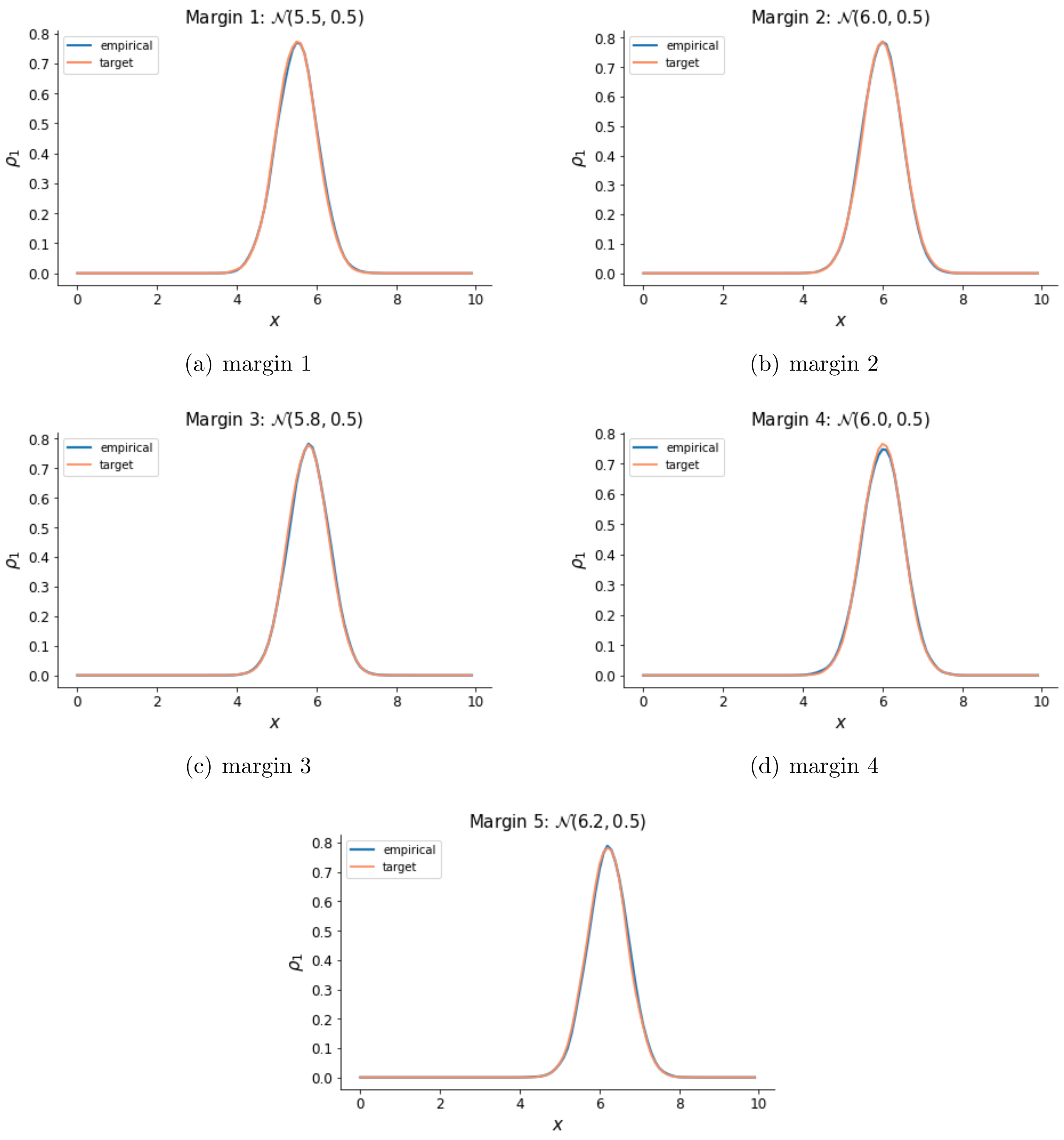
In the following, we use a cost function $F(B, \mathcal{A}) = \|B\|^2 + \|\mathcal{A}\|^2$ and solve the dual problem (4.13) with $d = 5$ and $d = 10$, respectively.

In the 5−d example, we set $x_0 = \begin{bmatrix} 5.0 \\ 5.0 \\ 5.0 \\ 5.0 \\ 5.0 \end{bmatrix}$, and let $\overline{\rho}_1$ be a multivariate normal distribution, where $\overline{\rho}_1 = \mathcal{N}\left(\begin{bmatrix} 5.5 \\ 6.0 \\ 5.8 \\ 6.0 \\ 6.2 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.10 & 0.10 & 0.10 & 0.10 \\ 0.10 & 0.25 & 0.10 & 0.10 & 0.10 \\ 0.10 & 0.10 & 0.25 & 0.10 & 0.10 \\ 0.10 & 0.10 & 0.10 & 0.25 & 0.10 \\ 0.10 & 0.10 & 0.10 & 0.10 & 0.25 \end{bmatrix} \right)$ [2].

Because the dimension is higher and the cost function is more complicated in this case, we need a bigger and deeper network. The AB-generator now has 5 layers with neurons $[400, 300, 200, 200, 150]$, the ϕ_1-generator keeps the same as before. Using 20,000 out-of-sample points of X_1 from the trained model, the empirical terminal density ρ_1 from Algorithm 7 has mean $\begin{bmatrix} 5.4929 \\ 5.9749 \\ 5.7855 \\ 5.9825 \\ 6.1944 \end{bmatrix}$ and covariance matrix $\begin{bmatrix} 0.2515 & 0.1041 & 0.1038 & 0.1037 & 0.1011 \\ 0.1041 & 0.2372 & 0.0926 & 0.0937 & 0.0885 \\ 0.1038 & 0.0926 & 0.2541 & 0.1103 & 0.0979 \\ 0.1037 & 0.0937 & 0.1103 & 0.2686 & 0.0906 \\ 0.1011 & 0.0885 & 0.0979 & 0.0906 & 0.2475 \end{bmatrix}$.

We use graphical tools to help us assess if the data plausibly come from the prescribed multivariate normal distribution. First, we compare the marginal distributions of ρ_1 with the ones of $\overline{\rho}_1$ in Figure 4.7. We can see that the empirical marginal distributions are consistent with the theoretical ones for all of the 5 marginals. Furthermore, we make a Q-Q plot for each margin, where we plot the quantiles of the empirical marginal distribution against the quantiles of the theoretical normal distribution. As we can see in Figure 4.8, for every margin, the points are close to a straight line. There are a few outliers at the two ends of the Q-Q plots, which is to be expected as Monte Carlo methods converge slowly for extreme values. Overall, the above two figures, in addition to the mean and covariance matrix, can verify the assumption that our dataset follows the target distribution $\overline{\rho}_1$ closely.

[2] For short, we write this prescribed distribution as $\overline{\rho}_1 = \mathcal{N}(\overline{\mu}, \overline{\Sigma})$.



(a) margin 1

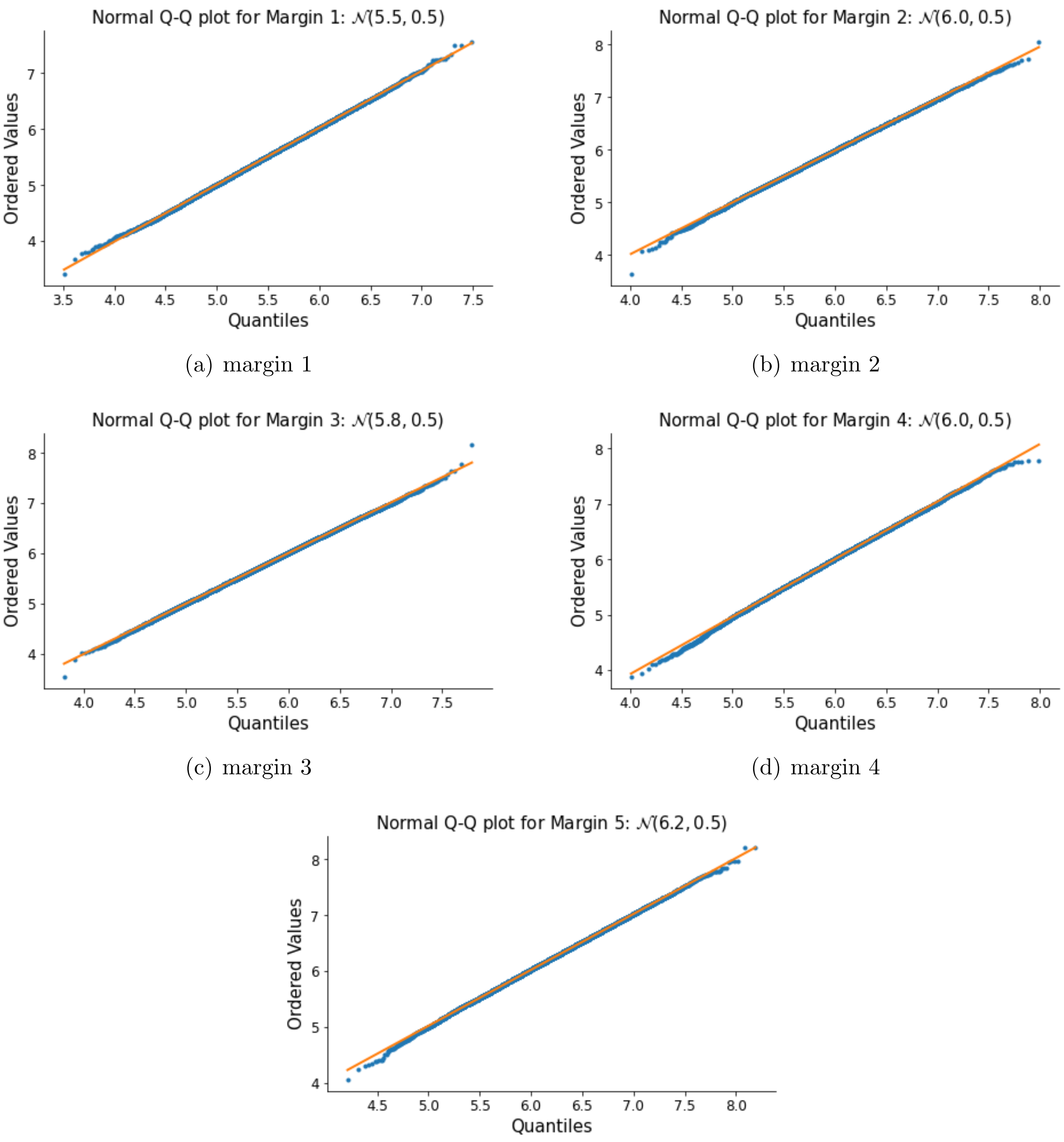
(b) margin 2

(c) margin 3

(d) margin 4

(e) margin 5

Figure 4.7: Marginal distributions of the 5-d sample



(a) margin 1

(b) margin 2

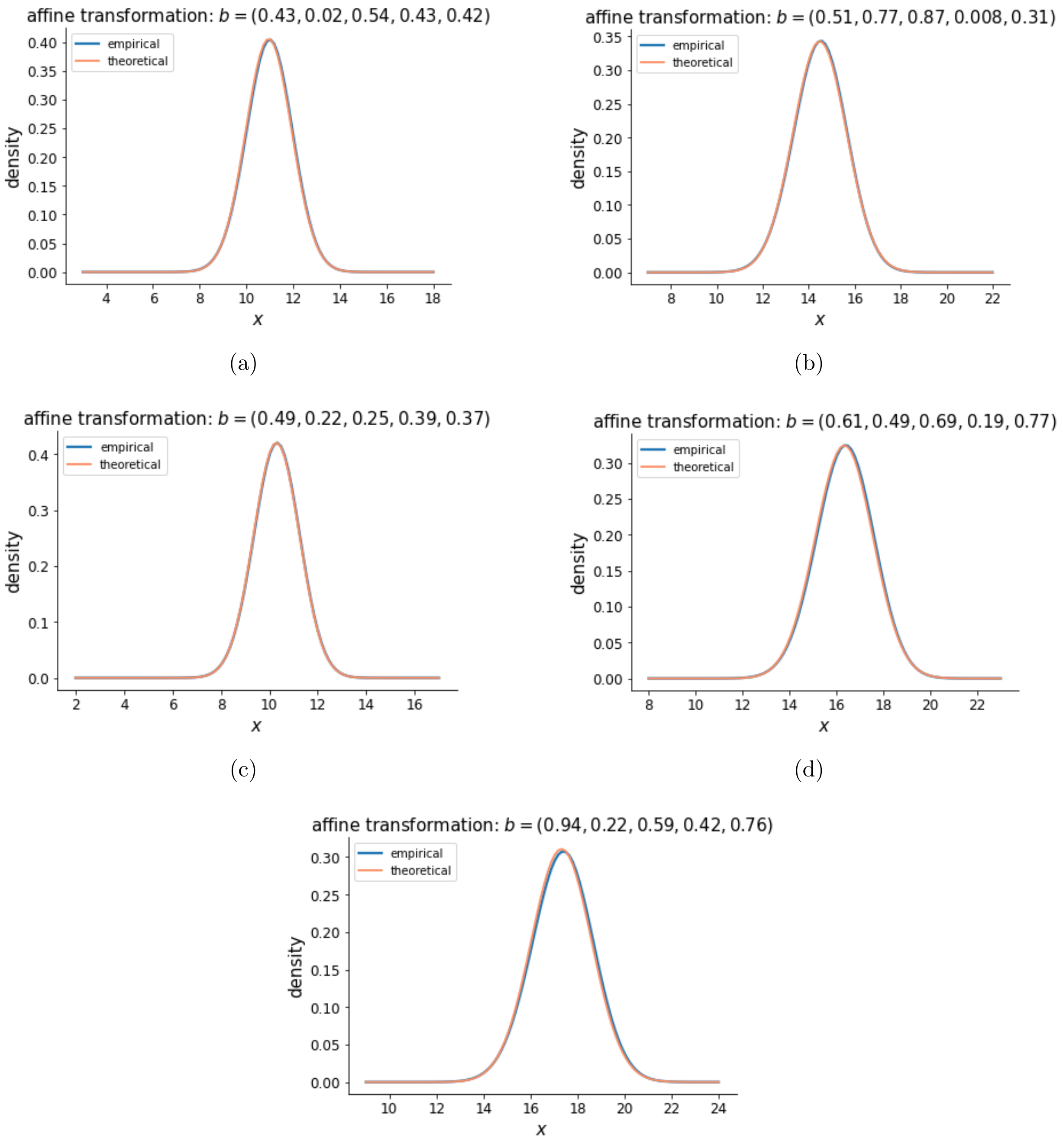
(c) margin 3

(d) margin 4

(e) margin 5

Figure 4.8: Normal Q-Q plot for margins of the 5-d sample

Next, we project the dataset X_1 on a random direction $b \in \mathbb{R}^5$. For a multivariate Gaussian distribution, the distribution after an affine transformation is univariate Gaussian. Let $X_{\text{affine}} = X_1 b$. After the transformation, X_{affine} should follow a normal distribution with mean $\mu_{\text{affine}} = \overline{\mu}^\intercal b$ and variance $\sigma^2_{\text{affine}} = b^\intercal \overline{\Sigma} b$. In the following figures, we illustrate the results of five random affine transformations. We compare the distributions of X_{affine} with the theoretical distributions (i.e., $\mathcal{N}(\mu_{\text{affine}}, \sigma^2_{\text{affine}})$) in Figure 4.9, and present their Q-Q plots in Figure 4.10. From these two figures, we can see that the dataset after the affine transformation follows the theoretical distribution closely.



(a)

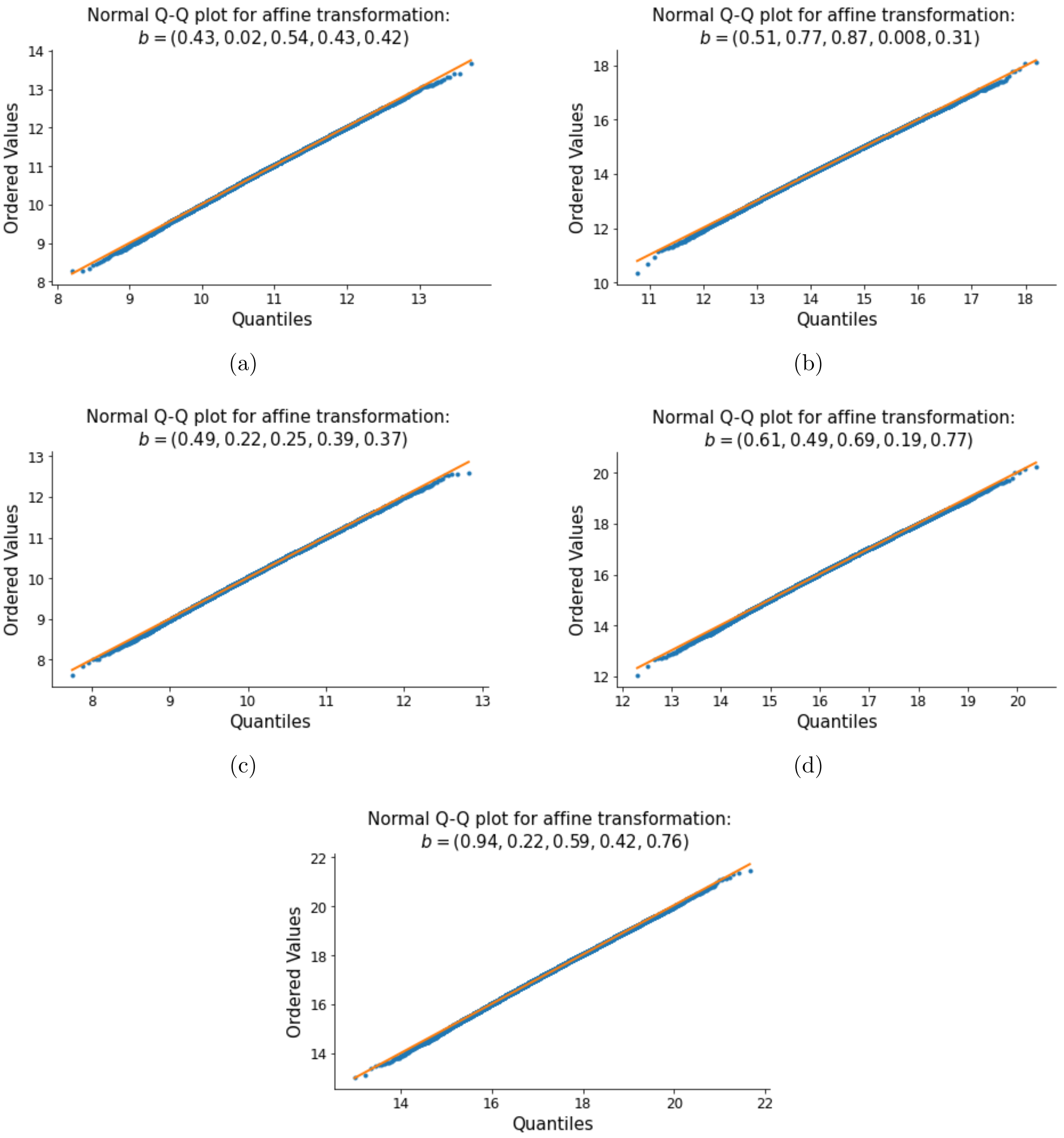
(b)

(c)

(d)

(e)

Figure 4.9: Densities after affine transformations



(a)

(b)

(c)

(d)

(e)

Figure 4.10: Normal Q-Q plots after affine transformations

We now introduce a loss metric to measure the quality of the dataset X_1. We will apply 40,000 random affine transformations on the 5-dimensional dataset we got from the Adversarial Network. For each affine transformation, we use the Wasserstein distance to measure the difference between the empirical distribution and the theoretical distribution after the transformation. The Wasserstein distance $W_p(P, Q)$ arises from the idea of optimal transport: intuitively, it measures how far you have to move the mass of P to turn it into Q. In particular, let $x_1, ..., x_n$ be an ordered dataset from the empirical distribution P, and $y_1, ..., y_n$ be an ordered dataset of the same size from the distribution Q, then the distance takes a very simple function of the order statistics: $W_2(P, Q) = (\sum_{i=1}^{n} ||x_i - y_i||^2)^{\frac{1}{2}}$. Inspired by this, we use a variation form of the 2-Wasserstein distance as the loss metric: for the k-th affine transformation, define the *average Wasserstein distance* between the empirical distribution and the theoretical distribution as $W(empirical, theoretical) := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\overline{\sigma}_k^2}||x_i - y_i||^2, \forall k \in [1, K]$, where $\overline{\sigma}_k$ is the standard deviation of the $k-$th theoretical distribution. In this case, we have $n =$ 20,000 and $K =$ 40,000. To be specific, for the k-th transformation, we get the 20,000 ordered sample points of X_1 and 20,000 quantiles from the theoretical distribution, then we compute the *average Wasserstein distance* between the empirical and theoretical distribution. In Figure 4.11, we show the histogram of the 40,000 average Wasserstein distances after affine transformations.

As a comparison, after each affine transformation, we also generate 20,000 random points from the theoretical distribution. Then we compute the average Wasserstein distances using the 20,000 quantiles from the theoretical distribution and the ordered samples generated from the theoretical distribution. We also present this histogram in Figure 4.12. We can use this 'correct answer' to evaluate the performance in Figure 4.11.

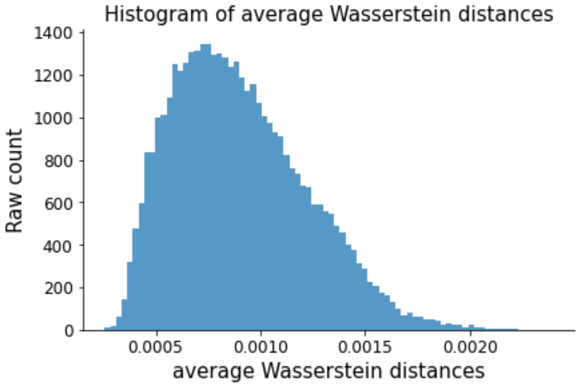


Figure 4.11: Wasserstein distances for the affine transformations of the 5 dimensional samples

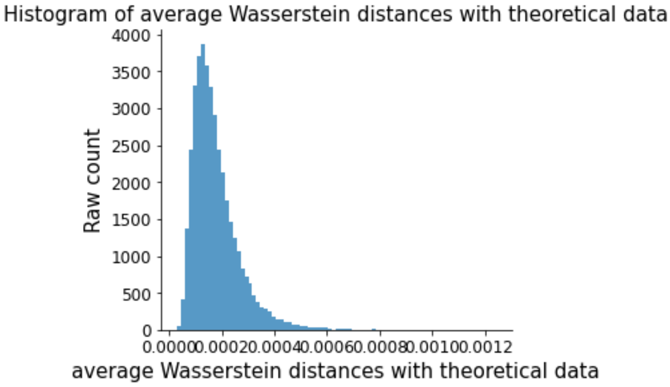


Figure 4.12: Wasserstein distances for the samples generated from theoretical distributions

Finally, we can go further to a 10-dimensional example, where we set $x_0 = \begin{bmatrix}5.0, & 5.0, & 5.0, & 5.0, & 5.0, & 5.0, & 5.0, & 5.0, & 5.0, & 5.0\end{bmatrix}^\intercal$ and the prescribed terminal distribution

$$\overline{\rho}_1 = \mathcal{N}\left(\begin{bmatrix}5.5\\6.0\\5.8\\6.0\\6.2\\5.5\\6.0\\5.8\\6.0\\6.2\end{bmatrix}, \begin{bmatrix}0.25 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.25 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.25 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.25 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.25 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.25 & 0.10 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.25 & 0.10 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.25 & 0.10 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.25 & 0.10\\ 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.25\end{bmatrix}\right).$$

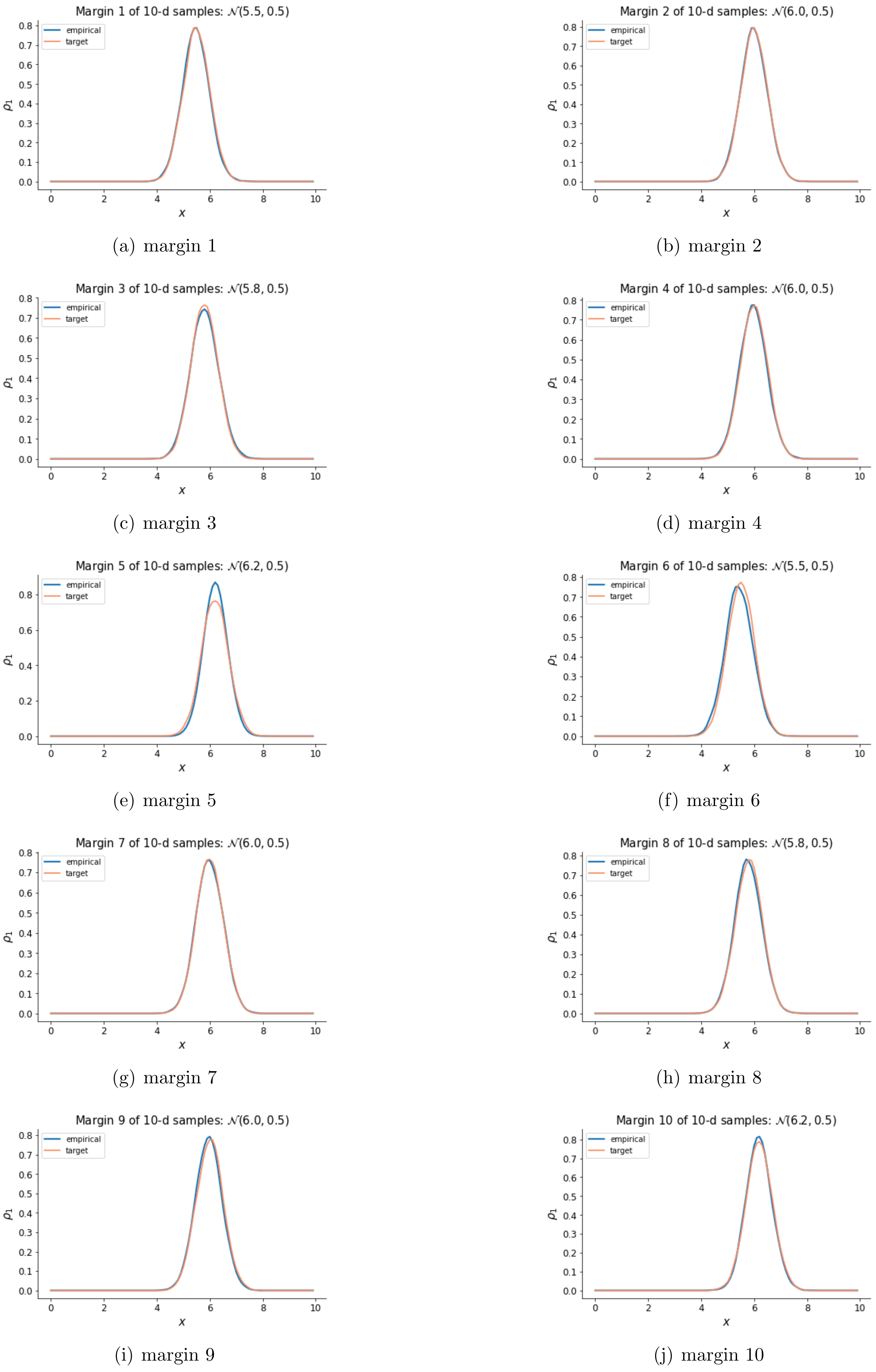
Using the same network constructed in the 5-d example, the empirical terminal density ρ_1 learnt from the Adversarial Network has mean

$$\begin{bmatrix}5.4711, & 5.9905, & 5.8023, & 5.9680, & 6.2131, & 5.4297, & 5.9932, & 5.7773, & 5.9596, & 6.1853\end{bmatrix}^{\intercal}$$

and covariance matrix

$$\begin{bmatrix}0.2522 & 0.1155 & 0.1891 & 0.1305 & 0.1288 & 0.1180 & 0.1016 & 0.1148 & 0.1114 & 0.1217\\ 0.1155 & 0.2494 & 0.1346 & 0.1275 & 0.0747 & 0.1170 & 0.1265 & 0.0837 & 0.1217 & 0.1042\\ 0.1891 & 0.1346 & 0.2763 & 0.1140 & 0.0939 & 0.1525 & 0.1211 & 0.1417 & 0.1298 & 0.0857\\ 0.1305 & 0.1275 & 0.1140 & 0.2622 & 0.1271 & 0.1163 & 0.0892 & 0.1094 & 0.1113 & 0.1067\\ 0.1288 & 0.0747 & 0.0939 & 0.1271 & 0.1858 & 0.1325 & 0.0901 & 0.0840 & 0.1120 & 0.0589\\ 0.1180 & 0.1170 & 0.1525 & 0.1163 & 0.1325 & 0.2696 & 0.1435 & 0.1121 & 0.1251 & 0.1178\\ 0.1016 & 0.1265 & 0.1211 & 0.0892 & 0.0901 & 0.1435 & 0.2578 & 0.1111 & 0.0751 & 0.0843\\ 0.1148 & 0.0837 & 0.1417 & 0.1094 & 0.0840 & 0.1121 & 0.1111 & 0.2548 & 0.1076 & 0.1185\\ 0.1114 & 0.1217 & 0.1298 & 0.1113 & 0.1120 & 0.1251 & 0.0751 & 0.1076 & 0.2455 & 0.0895\\ 0.1217 & 0.1042 & 0.0857 & 0.1067 & 0.0589 & 0.1178 & 0.0843 & 0.1185 & 0.0895 & 0.2368\end{bmatrix}.$$

We use a similar way to check the multivariate normality of the empirical distribution. In Figure 4.13, we plot the marginal distributions of the sample and the target marginal distributions. For most of the margins, the empirical distributions are consistent with the target ones. For margin 5, we can see the empirical one has a narrower shape (smaller variance), and the empirical margin 6 is shifted to the left (smaller mean). These imperfections can also be seen in the mean and covariance matrix of ρ_1. Then we present the Q-Q plots of the 10 marginal distributions in Figure 4.14. With the exception of some outliers at the two ends, the scatterplots are approximately straight. The Q-Q plots verify that the margins are all normally distributed.



(a) margin 1

(b) margin 2

(c) margin 3

(d) margin 4

(e) margin 5

(f) margin 6

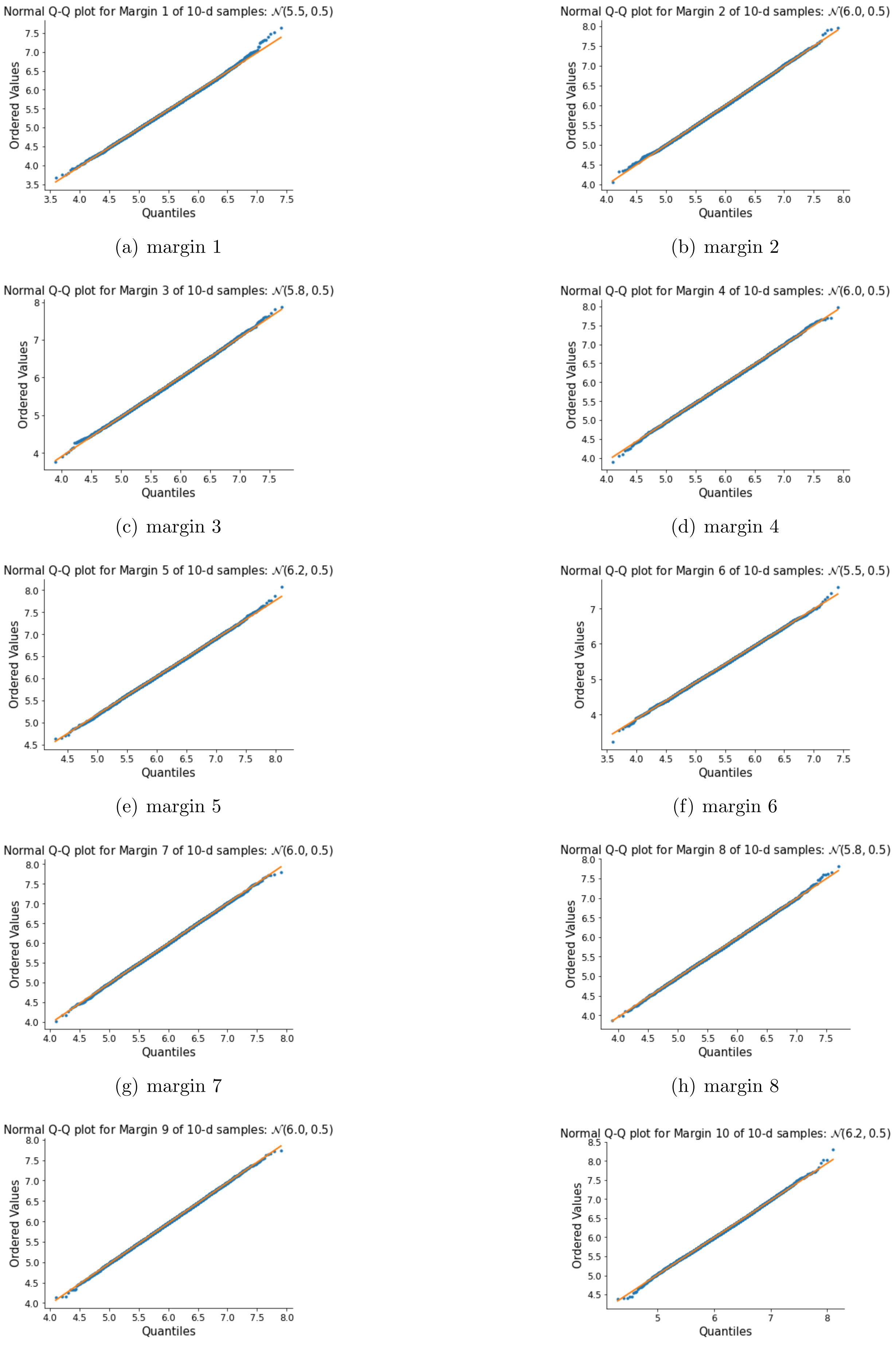
(g) margin 7

(h) margin 8

(i) margin 9

(j) margin 10

Figure 4.13: Marginal distributions for the 10-d sample



(a) margin 1

(b) margin 2

(c) margin 3

(d) margin 4

(e) margin 5

(f) margin 6

(g) margin 7

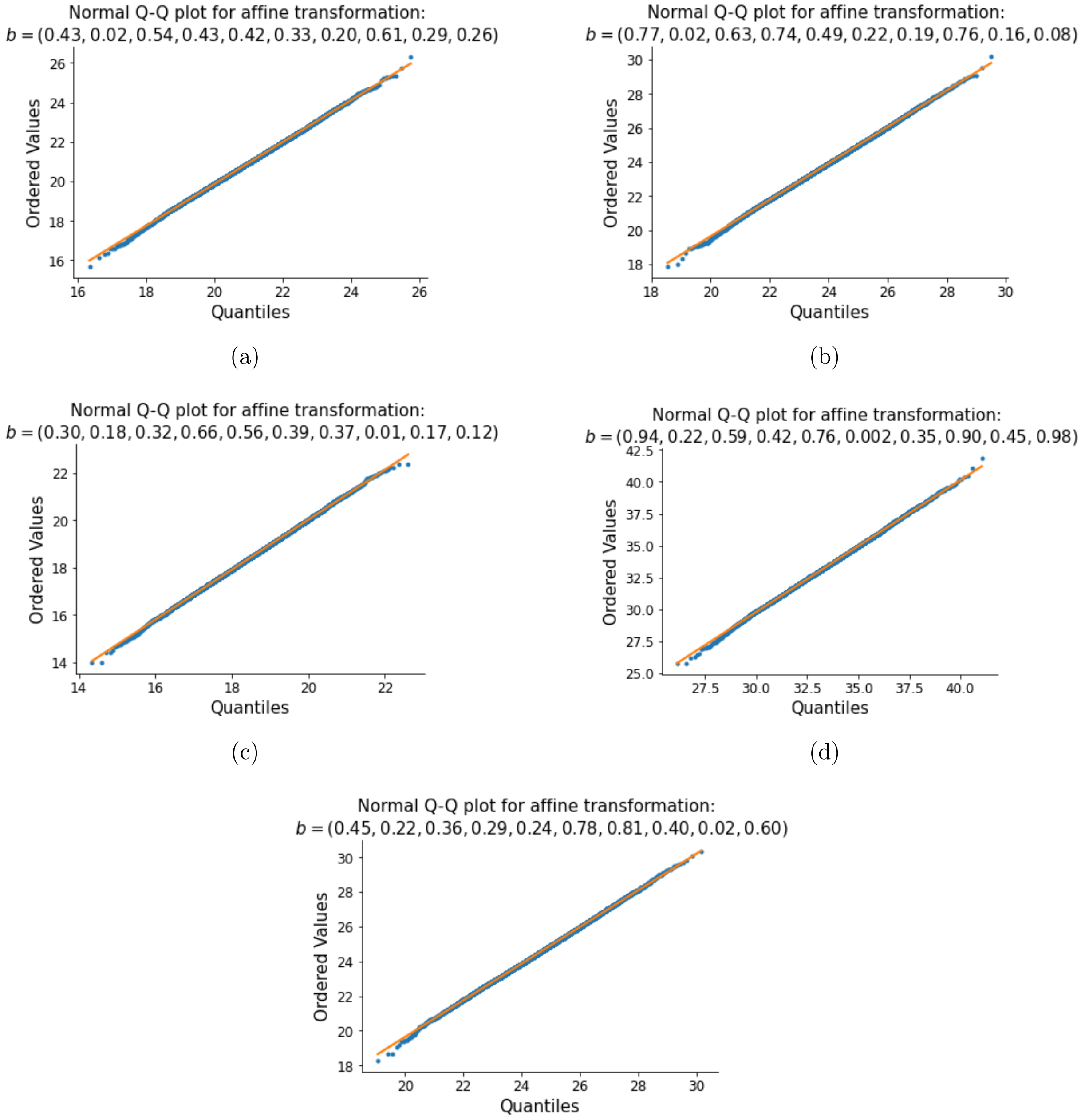
(h) margin 8

(i) margin 9

(j) margin 10

Figure 4.14: Normal Q-Q plots for margins of the 10-d sample

Next, we project the dataset $X_1 \in \mathbb{R}^{10}$ on a random direction $b \in \mathbb{R}^{10}$. We present their Q-Q plots of 5 random affine transformations in Figure 4.15.



(a)

(b)

(c)

(d)

(e)

Figure 4.15: Normal Q-Q plots after affine transformations for the 10-dimensional sample

Then, we apply 1,000,000 affine transformations on the 10-dimensional dataset we got from the Adversarial Network. Again, we use the previously defined *average Wasserstein distance* as a metric to quantify the difference between empirical distribution and the theoretical distribution after the affine transformation. In Figure 4.16 we plot the histogram of the 1,000,000 average Wasserstein distances for the transformed 10-d samples, and in Figure 4.17 we illustrate the average Wasserstein distances for the samples generated from theoretical distributions.

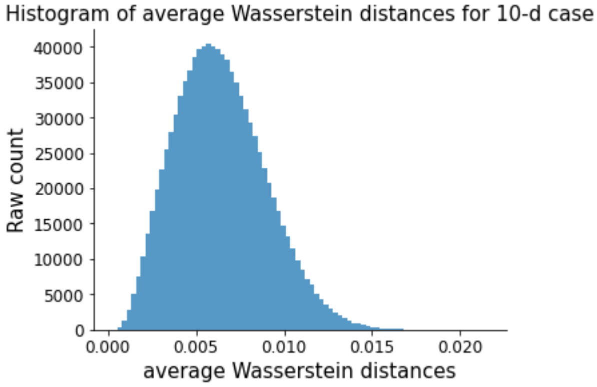


Figure 4.16: Wasserstein distances for the affine transformations of the 10-dimensional samples

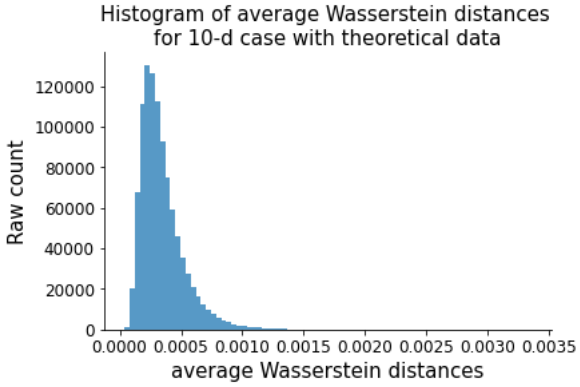


Figure 4.17: Wasserstein distances for the samples generated from theoretical distributions

We want to emphasize that, compared to other research where the prescribed distributions are restricted to Gaussian, our methods apply to a general choice of target distributions $\overline{\rho}_1$, such as heavy-tailed and asymmetric distributions. We choose multivariate normal target distributions only because the result is easier to be verified. In the following sections, we will demonstrate examples where $\overline{\rho}_1$ is not Gaussian.

4.5 Application to portfolio allocation

Now we are applying the deep learning algorithm (Algorithm 5) to the portfolio allocation problem introduced in Chapter 3, where the goal is to reach a prescribed wealth distribution $\overline{\rho}_1$ at the final time from an initial wealth x_0. Readers can refer to section 3.2.1 for the detailed description of this problem. We study this problem with the tools of optimal mass transport.

Let α be the portfolio allocation process valued in K. With a convex cost function $F(\alpha_t) : K \to \mathbb{R}$ and a penalty functional $C(\rho_1, \overline{\rho}_1)$, our objective function in this application is

$$\inf_{\alpha,\rho} \left\{ \int_{\mathcal{E}} F(\alpha_t) d\rho(t,x) + C(\rho_1, \overline{\rho}_1) \right\}, \tag{4.14}$$

where the feasible (α, ρ) in (4.14) should satisfy the initial distribution

$$\rho(0,x) = \rho_0(x) \quad \forall x \in \mathbb{R}, \tag{4.15}$$

and the Fokker–Planck equation

$$\partial_t \rho(t,x) + \partial_x(\alpha(t,x)^\intercal \mu(t,x) x \rho(t,x)) - \frac{1}{2}\partial_{xx}(\alpha(t,x)^\intercal \Sigma(t,x)\alpha(t,x)x^2\rho(t,x)) = 0 \quad \forall (t,x) \in \mathcal{E}. \tag{4.16}$$

If we write the drift $B(t,x)$ as $\alpha(t,x)^\intercal \mu(t,x)x$ and the diffusion $\mathcal{A}(t,x)$ as $\alpha(t,x)^\intercal \Sigma(t,x)\alpha(t,x)x^2$. It is easy to check that $\mathcal{A}$ and B are constrained in such a way that $\mathcal{A} \geq \frac{B^2}{\|\nu(t,x)\|^2}$, where $\nu(t,x) \coloneqq \Sigma(t,x)^{-\frac{1}{2}}\mu(t,x)$. Note that, due to the inequality constraint on the drift and diffusion, not all target distributions are attainable in this application. Therefore, we mainly utilize Algorithm 5 to solve this problem, it can be applied to all types of terminal distributions, attainable or unattainable.

4.5.1 Numerical Results

Now we implement Algorithm 5 to solve Problem 4.2.1 with various penalty functionals. In the experiment, we use $M = 512000$ Monte Carlo paths, and $N = 64$ time steps. We discretize the time horizon into constant steps with a step size $\Delta t = 1/N$ and the spatial domain into $I = 100$ constant grids. At each time step, the neural network θ_n consists of 3 layers, with neurons $[60, 40, 20]$. We feed the neural network with sequential mini-batches of size 1024 and trained it for 100 epochs.

Squared Euclidean Distance

In this example, we assume there is one risky asset in the portfolio, i.e., $\alpha_t \in \mathbb{R}$. We use the squared Euclidean distance as the penalty functional and a cost function $F(\alpha_t) = (\alpha_t - 0.5)^2$. The objective function is

$$V(\rho_0, \overline{\rho}_1) = \inf_{\alpha} \left\{ \mathbb{E}\Big[\int_0^1 (\alpha(t,x) - 0.5)^2 dt\Big] + \lambda \int_{\mathbb{R}} \frac{1}{2}\left(\rho_1(x) - \overline{\rho}_1(x)\right)^2 dx \right\},$$

where we choose $\lambda = 3000$, $x_0 = 5$ and $\overline{\rho}_1 = \mathcal{N}(6,1)$.

The discretized loss function is

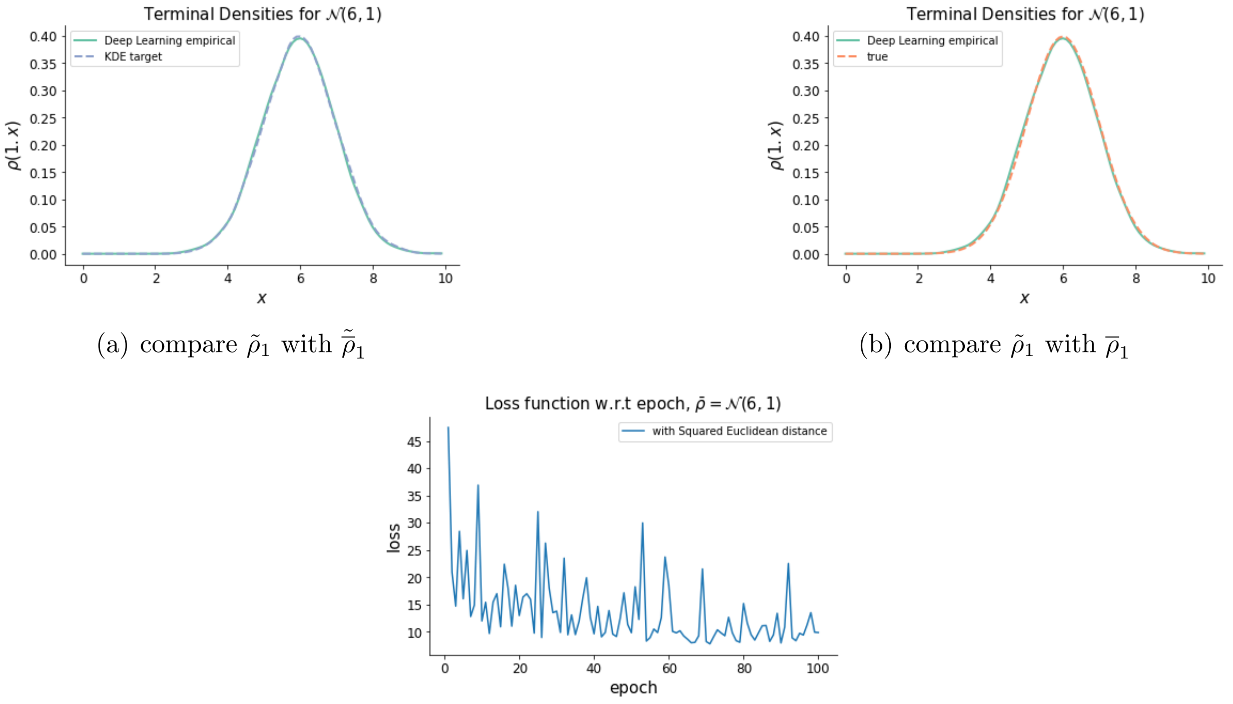
$$L = \frac{1}{M}\sum_{m=1}^{M}\left[\sum_{n=1}^{N}(\alpha_{n,m} - 0.5)^2 \Delta t\right] + \lambda \sum_{i=1}^{I} \frac{1}{2}\left(\tilde{\rho}_N(x_i) - \tilde{\overline{\rho}}_N(x_i)\right)^2 \Delta x.$$

We compare the KDE-estimated empirical terminal density $\tilde{\rho}_1$ with the KDE-estimated target density $\tilde{\overline{\rho}}_1$ in Figure 4.18(a); then we also compare $\tilde{\rho}_1$ with the true target density $\overline{\rho}_1$ in Figure 4.18(b). We can see that the portfolio allocation learned with the deep neural network can successfully steer the density to the prescribed one. We show the change of loss function through the training process in Figure 4.18(c). In training, we use a mini-batch gradient descent algorithm, which is a mix of batch gradient descent and stochastic gradient descent. In each step, it uses one mini-batch to compute the gradient and update the loss function. Therefore, as we can see in Figure 4.18(c), the loss function is not monotonic, but the overall trend is decreasing.

We can see the empirical density $\tilde{\rho}_1$ converges to the target $\tilde{\overline{\rho}}_1$ after the training. The current result is a trade-off between the cost function in α_t and the penalty functional in the terminal densities. In theory, we should set $\lambda = +\infty$ in the penalty functional to make $\rho_1 = \overline{\rho}_1$. Empirically, it is sufficient to set λ to be a large number for a similar effect.

The distance between $\tilde{\rho}_1$ and $\tilde{\overline{\rho}}_1$ comes from the imperfect training of the neural networks. There are two sources of error when we compare $\tilde{\rho}_1$ with the true target density

$\overline{\rho}_1$. One is the imperfectly trained model, and the other is the KDE estimation. To reduce the first kind of error, we can try different hyper-parameters in the neural network, e.g., different mini-batch sizes, different structures of the network, etc. On the other hand, we can reduce the second kind of error by choosing more appropriate kernels and bandwidth h for the KDE method.



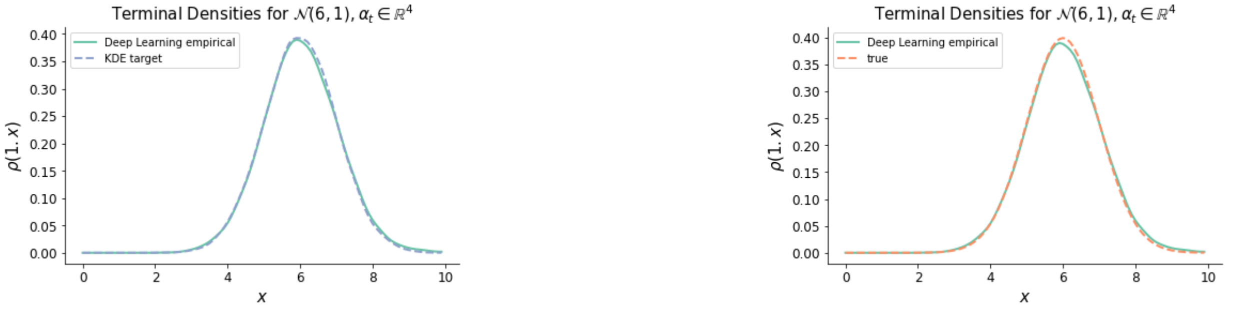
(a) compare $\tilde{\rho}_1$ with $\tilde{\overline{\rho}}_1$

(b) compare $\tilde{\rho}_1$ with $\overline{\rho}_1$

(c) Loss function during the training

Figure 4.18: Squared Euclidean: $\overline{\rho} = \mathcal{N}(6,1), \lambda = 3000, \alpha_t \in \mathbb{R}$

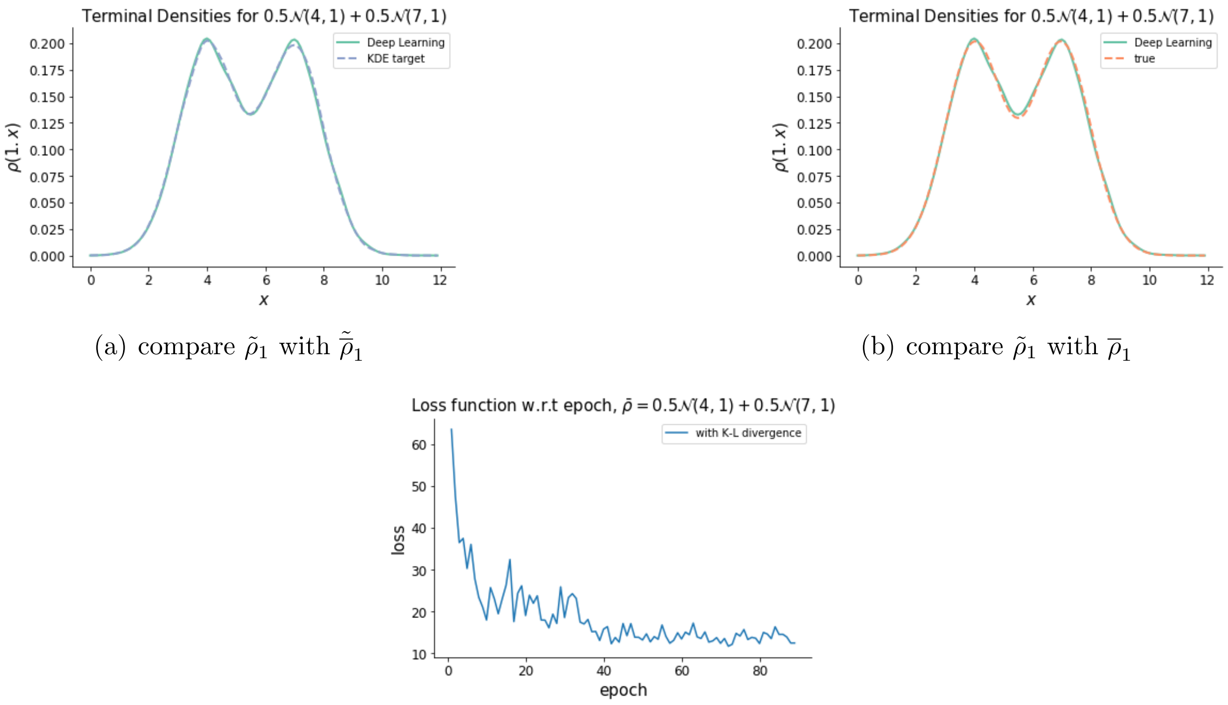
An advantage of this deep learning method is that it does not require much extra effort for multi-asset problems. We can easily obtain result for a 4-risky-asset case (or more) with good accuracy within computing time approximately the same as for the 1-risky-asset case. The result for such multi-asset example is shown in Figure 4.19.



(a) compare $\tilde{\rho}_1$ with $\tilde{\overline{\rho}}_1$

(b) compare $\tilde{\rho}_1$ with $\overline{\rho}_1$

Figure 4.19: Squared Euclidean: $\overline{\rho} = \mathcal{N}(6,1), \lambda = 3000, \alpha_t \in \mathbb{R}^4$



(a) compare $\tilde{\rho}_1$ with $\tilde{\bar{\rho}}_1$

(b) compare $\tilde{\rho}_1$ with $\bar{\rho}_1$

(c) Loss function during the training

Figure 4.20: K-L divergence: $\bar{\rho}_1 = 0.5\mathcal{N}(4,1) + 0.5\mathcal{N}(7,1), \lambda = 2000$

Kullback-Leibler divergence

In this second example, we use the Kullback–Leibler divergence to measure the difference between the two distributions. The Kullback–Leibler (K–L) divergence (Kullback and Leibler 1951) is also known as relative entropy or information deviation. In Machine Learning and neuroscience, the K–L divergence plays a leading role and is a widely used tool in pattern recognition (Mesaros et al. 2007), multimedia classification (Moreno et al. 2004), text classification (Dhillon et al. 2003) and so on. In the implementation, we may face $0 \log 0$ or *division by zero* cases in practice; to address this, we can replace zero with an infinitesimal positive value. In this case, the value function is defined as

$$V(\rho_0, \bar{\rho}_1) = \inf_{\alpha} \left\{ \mathbb{E}\Big[\int_0^1 (\alpha(t,x) - 0.5)^2 dt \Big] + \lambda \int_{\mathbb{R}} \rho_1(x) \log\left(\frac{\rho_1(x)}{\bar{\rho}_1(x)} \right) dx \right\},$$

and the discretized form of the loss function is

$$L = \frac{1}{M} \sum_{m=1}^{M} \left[\sum_{n=1}^{N} (\alpha_{n,m} - 0.5)^2 \, \Delta t \right] + \lambda \sum_{i=1}^{I} \tilde{\rho}_N(x_i) \log\left(\frac{\tilde{\rho}_N(x_i)}{\tilde{\bar{\rho}}_N(x_i)} \right) \Delta x.$$

To show that our method is not restricted to Gaussian target distributions, we use a mixture of two normal distributions as the target $\bar{\rho}_1 = 0.5\mathcal{N}(4,1) + 0.5\mathcal{N}(7,1)$ in this example. We provide the comparisons between $\tilde{\rho}_1$ and $\tilde{\bar{\rho}}_1$, $\tilde{\rho}_1$ and $\bar{\rho}_1$ in figures 4.20(a) and 4.20(b), respectively. The loss function is shown afterwards, in Figure 4.20(c).

We want to emphasize that our method is not restricted to constant parameters. The risky assets can have time-varying returns and covariances. We are not illustrating the results here because the output plots look very similar to the presented cases.

2-Wasserstein Distance

Here, we use 2-Wasserstein distance as the penalty functional. Wasserstein distance is closely related to the optimal transport problem. When we think of the optimal transport problem as an optimization problem, then the Wasserstein distance is simply the optimal objective value of this optimization problem, with some power transformation. We refer interested readers to Del Barrio et al. (2019), Panaretos and Zemel (2019) and Weed et al. (2019) for more properties of the Wasserstein distance. The Wasserstein distance is widely used in Machine Learning to formulate a metric for comparing clusters (Coen et al., 2010), and has been applied to image retrieval (Rubner et al., 2000), contour matching (Grauman and Darrell, 2004), and many other problems. The Wasserstein distance has some advantages compared to distances such as L^2, χ^2 or Hellinger. First of all, it can capture the underlying geometry of the space, which may be ignored by the Euclidean distance. Secondly, when we take average of different objects – such as distributions and images – we can get back to a similar object with the Wasserstein distance. Thirdly, some of the above distances are sensitive to small wiggles in the distribution, but the Wasserstein distance is insensitive to small wiggles.

The 2-Wasserstein distance between two distributions ρ_1 and $\overline{\rho}_1$ is defined by

$$W_2(\rho_1, \overline{\rho}_1) = \left(\inf_{\gamma \in \Gamma(\rho_1, \overline{\rho}_1)} \int_{\mathbb{R} \times \mathbb{R}} |x - y|^2 \, d\gamma(x, y) \right)^{\frac{1}{2}}.$$

For continuous one-dimensional probability distributions ρ_1 and $\overline{\rho}_1$ on $\mathbb{R}$, the distance has a closed form in terms of the corresponding cumulative distribution functions $F(x)$ and $\overline{F}(x)$ (see Rüschendorf 1985 for the detailed proof):

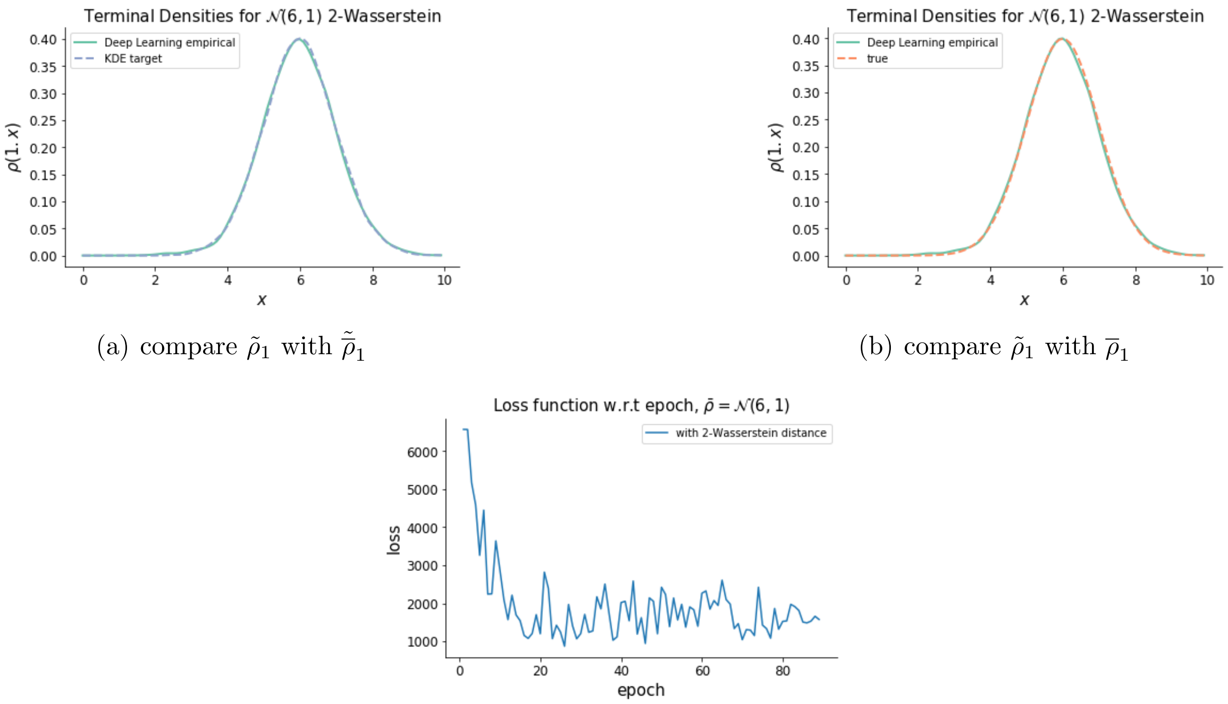
$$W_2(\rho_1, \overline{\rho}_1) = \left(\int_0^1 \left| F^{-1}(u) - \overline{F}^{-1}(u) \right|^2 du \right)^{\frac{1}{2}}.$$

Empirically, we can use the order statistics:

$$W_2(\rho_1, \overline{\rho}_1) = \left(\sum_{i=1}^{n} |X_i - Y_i|^2 \right)^{\frac{1}{2}},$$

where the dataset $X_1, X_2, ...X_n$ is increasingly ordered with an empirical distribution ρ_1, similarly the dataset $Y_1, Y_2, ...Y_n$ is increasingly ordered with an empirical distribution $\overline{\rho}_1$.

Using the same cost function and network structure as before, the numerical results for the densities and loss function are illustrated in Figure 4.21.



(a) compare $\tilde{\rho}_1$ with $\tilde{\bar{\rho}}_1$

(b) compare $\tilde{\rho}_1$ with $\bar{\rho}_1$

(c) Loss function during the training

Figure 4.21: 2-Wasserstein distance: $\bar{\rho}_1 = \mathcal{N}(6,1), \lambda = 4000$

4.6 Discussion and conclusion

In this chapter, we first devise a deep learning method to solve the optimal transport problem via a penalization method. In particular, we relax the classical optimal transport problem and introduce a functional to penalize the deviation between the empirical terminal density and the prescribed one. In section 4.5, we apply this deep learning method to the portfolio allocation problem raised in Guo et al. (2020a), where the goal is to reach a prescribed wealth distribution at the final time. We then provide numerical results for various choices of penalty functionals and target densities.

Then we investigate the dual representation and find it can be written as a saddle point problem. In this way, the optimal transport problem can be interpreted as a minimax game between the drift/diffusion and the potential function. We then solve this differential game with adversarial networks. Because this algorithm is free of spatial discretization, it can be applied to high dimensional optimal transport problems, where we illustrate an example up to dimension 10. These examples validate the accuracy and flexibility of the proposed deep optimal transport method.

In this work, the optimal transport problem starts from a point. This can be easily extended to more general situations. In future research, we will investigate more applications of deep learning in high-dimensional optimal transport problems, and we aim to propose more efficient neural network-based algorithms.

Bibliography

Bachouch, A., Huré, C., Langrené, N., and Pham, H. (2018a). Deep neural networks algorithms for stochastic control problems on finite horizon: numerical applications. *arXiv preprint arXiv:1812.05916*.

Bachouch, A., Huré, C., Langrené, N., and Pham, H. (2018b). Deep neural networks algorithms for stochastic control problems on finite horizon, part 2: Numerical applications.

Bajeux-Besnainou, I. and Portait, R. (1998). Dynamic asset allocation in a mean-variance framework. *Management Science*, 44(11-part-2):S79–S95.

Balata, A. and Palczewski, J. (2017). Regress-later Monte Carlo for optimal control of Markov processes. *arXiv preprint arXiv:1712.09705*.

Balata, A. and Palczewski, J. (2018). Regress-later Monte Carlo for optimal inventory control with applications in energy. *arXiv preprint arXiv:1703.06461*.

Baltas, I., Xepapadeas, A., and Yannacopoulos, A. N. (2019). Robust control of parabolic stochastic partial differential equations under model uncertainty. *European Journal of Control*, 46:1–13.

Bel Hadj Ayed, A., Loeper, G., and Abergel, F. (2017). Forecasting trends with asset prices. *Quantitative Finance*, 17(3):369–382.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, 23(4):769–805.

Benamou, J.-D. and Brenier, Y. (2000). A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393.

Benamou, J.-D., Carlier, G., and Nenna, L. (2019). Generalized incompressible flows, multi-marginal transport and Sinkhorn algorithm. *Numerische Mathematik*, 142(1):33–54.

Beutner, E., Pelsser, A., and Schweizer, J. (2013). Fast convergence of regress-later estimates in least squares Monte Carlo.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brenier, Y. (1991). Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417.

Brenier, Y., Frisch, U., Henon, M., Loeper, G., Matarrese, S., Mohayaee, R., and Sobolevskii, A. (2003). Reconstruction of the early universe as a convex optimization problem. arxiv (september 2003). *arXiv preprint astro-ph/0304214*.

Brezis, H. (2010). *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media.

Buckdahn, R. and Li, J. (2008). Stochastic differential games and viscosity solutions of Hamilton–Jacobi–Bellman–Isaacs equations. *SIAM Journal on Control and Optimization*, 47(1):444–475.

Cao, H., Guo, X., and Lauriere, M. (2020). Connecting GANs and MFGs. *arXiv preprint arXiv:2002.04112*.

Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19(1):19–30.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1.

Chalabi, Y. and Wuertz, D. (2012). Portfolio optimization based on divergence measures.

Chartrand, R., Wohlberg, B., Vixie, K., and Bollt, E. (2009). A gradient descent solution to the Monge-Kantorovich problem. *Applied Mathematical Sciences*, 3(22):1071–1080.

Chen, L., Pelger, M., and Zhu, J. (2019a). Deep learning in asset pricing. *Available at SSRN 3350138*.

Chen, Y., Georgiou, T. T., and Pavon, M. (2018). Steering the Distribution of Agents in Mean-Field Games System. *Journal of Optimization Theory and Applications*, 179(1):332–357.

Chen, Y., Georgiou, T. T., and Pavon, M. (2019b). Covariance steering in zero-sum linear-quadratic two-player differential games. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 8204–8209. IEEE.

Chow, Y. T., Li, W., Osher, S., and Yin, W. (2019). Algorithm for Hamilton–Jacobi equations in density space via a generalized Hopf formula. *Journal of Scientific Computing*, 80(2):1195–1239.

Clément, E., Lamberton, D., and Protter, P. (2002). An analysis of a least squares regression method for American option pricing. *Finance and Stochastics*, 6(4):449–471.

Coen, M. H., Ansari, M. H., and Fillmore, N. (2010). Comparing clusterings in space. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 231–238.

Cosso, A. and Pham, H. (2019). Zero-sum stochastic differential games of generalized McKean–Vlasov type. *Journal de Mathématiques Pures et Appliquées*, 129:180–212.

Crandall, M. G. and Lions, P.-L. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42.

Cuchiero, C., Khosrawi, W., and Teichmann, J. (2020). A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300.

Dai, M. and Zhong, Y. (2008). Penalty methods for continuous-time portfolio selection with proportional transaction costs. *Available at SSRN 1210105*.

Del Barrio, E., Loubes, J.-M., et al. (2019). Central limit theorems for empirical transportation cost in general dimension. *The Annals of Probability*, 47(2):926–951.

Denault, M. and Simonato, J.-G. (2017). Dynamic portfolio choices by simulation-and-regression: revisiting the issue of value function vs portfolio weight recursions. *Computers & Operations Research*, 79:174–189.

Dhillon, I. S., Mallela, S., and Kumar, R. (2003). A divisive information-theoretic feature clustering algorithm for text classification. *Journal of machine learning research*, 3(Mar):1265–1287.

Dolinsky, Y. and Soner, H. M. (2014). Martingale optimal transport and robust hedging in continuous time. *Probability Theory and Related Fields*, 160(1-2):391–427.

Eckstein, S. and Kupper, M. (2019). Computation of optimal transport and related hedging problems via penalization and neural networks. *Applied Mathematics & Optimization*, pages 1–29.

El Ghaoui, L. and Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on matrix analysis and applications*, 18(4):1035–1064.

Elie, R. and Touzi, N. (2008). Optimal lifetime consumption and investment under a drawdown constraint. *Finance and Stochastics*, 12(3):299.

ElKouri, N. (1981). Les aspects probabilistes du controle stochastique. *Lecture Notes in Math. Srpinger, New York*, 876.

Elliott, R. J. and Kalton, N. J. (1972). *The existence of value in differential games*, volume 126. American Mathematical Soc.

Elliott, R. J. and Siu, T. K. (2009). Robust optimal portfolio choice under Markovian regime-switching model. *Methodology and Computing in Applied Probability*, 11(2):145–157.

Fabozzi, F. J., Kolm, P. N., Pachamanova, D. A., and Focardi, S. M. (2007). *Robust portfolio optimization and management*. John Wiley & Sons.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014). Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882.

Fleming, W. H. (1961). The convergence problem for differential games. *Journal of Mathematical Analysis and Applications*, 3(1):102–116.

Fleming, W. H. (1964). The convergence problem for differential games ii. *Advances in game theory*, pages 195–210.

Fleming, W. H. and Soner, H. M. (2006). *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media.

Fleming, W. H. and Souganidis, P. E. (1989). On the existence of value functions of two-player, zero-sum stochastic differential games. *Indiana University Mathematics Journal*, 38(2):293–314.

Fouque, J.-P., Pun, C. S., and Wong, H. Y. (2016). Portfolio optimization with ambiguous correlation and stochastic volatilities. *SIAM Journal on Control and Optimization*, 54(5):2309–2338.

Friedman, A. (1971). *Differential games.* New York: Wiley-Interscience.

Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

Glasserman, P. and Xu, X. (2013). Robust portfolio control with stochastic factor dynamics. *Operations Research*, 61(4):874–893.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Grauman, K. and Darrell, T. (2004). Fast contour matching using approximate Earth Mover's Distance. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE.

Guo, I., Langrené, N., Loeper, G., and Ning, W. (2019a). Robust utility maximization under model uncertainty via a penalization approach. *arXiv preprint arXiv:1907.13345.*

Guo, I., Langrené, N., Loeper, G., and Ning, W. (2020a). Portfolio optimization with a prescribed terminal wealth distribution. *arXiv preprint arXiv:2009.12823.*

Guo, I. and Loeper, G. (2018). Path dependent optimal transport and model calibration on exotic derivatives. *arXiv preprint arXiv:1812.03526.*

Guo, I., Loeper, G., Obłój, J., and Wang, S. (2020b). Joint modelling and calibration of spx and vix by optimal transport. *Available at SSRN 3568998.*

Guo, I., Loeper, G., and Wang, S. (2017). Local volatility calibration by optimal transport. *arXiv preprint arXiv:1709.08075.*

Guo, I., Loeper, G., and Wang, S. (2019b). Calibration of local-stochastic volatility models by optimal transport. *arXiv preprint arXiv:1906.06478.*

Guo, I., Loeper, G., and Wang, S. (2019c). Local volatility calibration by optimal transport. In *2017 MATRIX Annals*, pages 51–64. Springer, Cham.

Haber, E. and Horesh, R. (2015). A multilevel method for the solution of time dependent optimal transport. *Numerical Mathematics: Theory, Methods and Applications*, 8(1):97–111.

Han, J. and E, W. (2016). Deep learning approximation for stochastic control problems. In *NIPS 2016, Deep Reinforcement Learning Workshop.*

Han, J. and Jentzen, A. (2020). Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to Machine Learning. *arXiv preprint arXiv:2008.13333.*

Hansen, L. P. and Sargent, T. J. (2007). Recursive robust estimation and control without commitment. *Journal of Economic Theory*, 136(1):1–27.

Hansen, L. P. and Sargent, T. J. (2008). *Robustness*. Princeton university press.

Hansen, L. P. and Sargent, T. J. (2011). Robustness and ambiguity in continuous time. *Journal of Economic Theory*, 146(3):1195–1223.

Hansen, L. P., Sargent, T. J., et al. (2005). Robust estimation and control under commitment. *Journal of economic Theory*, 124(2):258–301.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Henry-Labordère, P. (2017). *Model-free hedging: A martingale optimal transport viewpoint*. CRC Press.

Henry-Labordère, P. (2019). (Martingale) optimal transport and anomaly detection with neural networks: a primal-dual algorithm. *Available at SSRN 3370910*.

Henry-Labordère, P., Tan, X., and Touzi, N. (2016). An explicit martingale version of the one-dimensional Brenier's theorem with full marginals constraint. *Stochastic Processes and their Applications*, 126(9):2800–2834.

Hernández-Hernández, D. and Sîrbu, M. (2018). Zero-sum stochastic differential games without the Isaacs condition: random rules of priority and intermediate Hamiltonians. *SIAM Journal on Control and Optimization*, 56(3):2095–2119.

Huré, C., Pham, H., Bachouch, A., and Langrené, N. (2018). Deep neural networks algorithms for stochastic control problems on finite horizon: convergence analysis. *arXiv preprint arXiv:1812.04300*.

Huré, C., Pham, H., Bachouch, A., and Langrené, N. (2018). Deep neural networks algorithms for stochastic control problems on finite horizon, part i: convergence analysis. *arXiv preprint arXiv:1812.04300*.

Huré, C., Pham, H., and Warin, X. (2019). Some machine learning schemes for high-dimensional nonlinear pdes. *arXiv preprint arXiv:1902.01599*.

Ikeda, N. and Watanabe, S. (2014). *Stochastic differential equations and diffusion processes*. Elsevier.

Isaacs, R. (1965). *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. SIAM Series in Applied Mathematics. Wiley, New York.

Ismail, A. and Pham, H. (2019). Robust Markowitz mean-variance portfolio selection under ambiguous covariance matrix. *Mathematical Finance*, 29(1):174–207.

Jonsson, M. and Sircar, K. R. (2002a). Partial hedging in a stochastic volatility environment. *Mathematical Finance*, 12(4):375–409.

Jonsson, M. and Sircar, R. (2002b). Optimal investment problems and volatility homogenization approximations. In *Modern Methods in Scientific Computing and Applications*, pages 255–281. Springer.

Kantorovich, L. V. (1942). On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201.

Karatzas, I., Lehoczky, J. P., and Shreve, S. E. (1987). Optimal portfolio and consumption decisions for a small investor on a finite horizon. *SIAM journal on control and optimization*, 25(6):1557–1586.

Karatzas, I. and Shreve, S. (2012). *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media.

Karatzas, I. and Shreve, S. E. (1998). Brownian motion. In *Brownian Motion and Stochastic Calculus*, pages 47–127. Springer.

Karatzas, I., Shreve, S. E., Karatzas, I., and Shreve, S. E. (1998). *Methods of mathematical finance*, volume 39. Springer.

Kharroubi, I., Langrené, N., and Pham, H. (2014). A numerical algorithm for fully nonlinear HJB equations: an approach by control randomization. *Monte Carlo Methods and Applications*, 20(2):145–165.

Kharroubi, I., Langrené, N., and Pham, H. (2015). Discrete time approximation of fully nonlinear HJB equations via BSDEs with nonpositive jumps. *Annals of Applied Probability*, 25(4):2301–2338.

Klebaner, F. C. (2005). *Introduction to stochastic calculus with applications.* World Scientific Publishing Company.

Kraus, A. and Litzenberger, R. H. (1976). Skewness preference and the valuation of risk assets. *The Journal of finance*, 31(4):1085–1100.

Krylov, N. V. (2008). *Controlled diffusion processes*, volume 14. Springer Science & Business Media.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lakner, P. (1995). Utility maximization with partial information. *Stochastic Processes and their Applications*, 56(2):247–273.

Lee, C. F. (1977). Functional form, skewness effect, and the risk-return relationship. *Journal of financial and quantitative analysis*, 12(1):55–72.

Léonard, C. (2013). A survey of the Schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215.*

Li, W., Ryu, E. K., Osher, S., Yin, W., and Gangbo, W. (2018). A parallel method for Earth Mover's Distance. *Journal of Scientific Computing*, 75(1):182–197.

Liang, S. and Srikant, R. (2016). Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161.*

Lions, P.-L. (1983). Optimal control of diffusion processes and Hamilton–Jacobi–Bellman equations part 2: viscosity solutions and uniqueness. *Communications in partial differential equations*, 8(11):1229–1276.

Liu, H. and Loewenstein, M. (2002). Optimal portfolio selection with transaction costs and finite horizons. *The Review of Financial Studies*, 15(3):805–835.

Loeper, G. (2006). The reconstruction problem for the Euler-Poisson system in cosmology. *Archive for rational mechanics and analysis*, 179(2):153–216.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147.

Ludkovski, M. and Maheshwari, A. (2019). Simulation methods for stochastic storage problems: A statistical learning perspective. *Energy Systems*, pages 1–39.

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Markowitz, H. M. (1959). *Portfolio selection: Efficient diversification of investments*. John Wiley.

MathWorks (2020). fun.

Matoussi, A., Possamaï, D., and Zhou, C. (2015). Robust utility maximization in non-dominated models with 2BSDE: the uncertain volatility model. *Mathematical Finance*, 25(2):258–287.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, pages 247–257.

Merton, R. C. (1971). Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):373–413.

Mesaros, A., Virtanen, T., and Klapuri, A. (2007). Singer identification in polyphonic music using vocal separation and pattern recognition methods. In *ISMIR*, pages 375–378.

Mikami, T. (2015). Two end points marginal problem by stochastic optimal transportation. *SIAM Journal on Control and Optimization*, 53(4):2449–2461.

Mikami, T. and Thieullen, M. (2006). Duality theorem for the stochastic optimal control problem. *Stochastic processes and their applications*, 116(12):1815–1835.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris.*

Moreno, P. J., Ho, P. P., and Vasconcelos, N. (2004). A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in neural information processing systems*, pages 1385–1392.

Nisio, M. (2015). *Stochastic control theory*, volume 72. Springer Science & Business Media.

Noh, E.-J. and Kim, J.-H. (2011). An optimal portfolio model with stochastic volatility and stochastic interest rate. *Journal of Mathematical Analysis and Applications*, 375(2):510–522.

Oksendal, B. and Sulem, A. (2002). Optimal consumption and portfolio with both fixed and proportional transaction costs. *SIAM Journal on Control and Optimization*, 40(6):1765–1790.

Panaretos, V. M. and Zemel, Y. (2019). Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport: with applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Pham, H. (2009). *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media.

Rachev, S. T. and Rüschendorf, L. (1998). *Mass Transportation Problems: Volume I: Theory*, volume 1. Springer Science & Business Media.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

Rockafellar, R. T. (1970). *Convex analysis*, volume 36. Princeton university press.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media.

Rogers, L. C. (2013). *Optimal investment.* Springer.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The Earth Mover's Distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rüschendorf, L. (1985). The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129.

Ruthotto, L., Osher, S. J., Li, W., Nurbekyan, L., and Fung, S. W. (2020). A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In *Advances in neural information processing systems*, pages 2234–2242.

Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94.

Shen, Z. and Weng, C. (2019). A backward simulation method for stochastic optimal control problems. *arXiv preprint arXiv:1901.06715.*

Sirignano, J. and Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364.

Talay, D. and Zheng, Z. (2002). Worst case model risk management. *Finance and Stochastics*, 6(4):517–537.

Tan, X., Touzi, N., et al. (2013). Optimal transportation under controlled stochastic dynamics. *The Annals of Probability*, 41(5):3201–3240.

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

Villani, C. (2003). *Topics in optimal transportation.* Number 58. American Mathematical Soc.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Wand, M. and Jones, C. (1994). *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability.* CRC press.

Weed, J., Bach, F., et al. (2019). Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *Bernoulli*, 25(4A):2620–2648.

Weinan, E., Han, J., and Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380.

Wiese, M., Knobloch, R., Korn, R., and Kretschmer, P. (2020). Quant GANs: deep generation of financial time series. *Quantitative Finance*, pages 1–22.

Yong, J. and Zhou, X. Y. (1999). *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media.

Zeidler, E. (2013). *Nonlinear functional analysis and its applications: III: variational methods and optimization.* Springer Science & Business Media.

Zhang, J., Zhong, W., and Ma, P. (2020). A review on modern computational optimal transport methods with applications in biomedical research. *arXiv preprint arXiv:2008.02995.*

Zhang, K., Zhong, G., Dong, J., Wang, S., and Wang, Y. (2019a). Stock market prediction based on generative adversarial network. *Procedia computer science*, 147:400–406.

Zhang, R., Langrené, N., Tian, Y., Zhu, Z., Klebaner, F., and Hamza, K. (2019b). Dynamic portfolio optimization with liquidity cost and market impact: a simulation-and-regression approach. *Quantitative Finance*, 19(3):519–532.

Zhang, R., Langrené, N., Tian, Y., Zhu, Z., Klebaner, F., and Hamza, K. (2019c). Skewed target range strategy for multiperiod portfolio optimization using a two-stage least squares Monte Carlo method. *Journal of Computational Finance*, 23(1):97–127.

Appendix A

Appendix for Chapter 2

A.1

Proof of Proposition 2.3.2.

First of all, define $w(t,x) := \sup_{\alpha\in\mathcal{A}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta})\right]$, $(t,x)\in[0,T]\times\mathbb{R}$. All the assumptions on α, U, X_t hold for $w(t,x)$, except that we assume θ for time $u\in[t,T]$ is a fixed known process in $\mathcal{B}$. An argument used in Pham (2009, p.52) proved that, when the utility function $U(\cdot)$ is continuous, increasing and concave on $\mathbb{R}$, $w(t,\cdot)$ is also increasing and concave in x, $\forall t\in[0,T]$.

For any fixed process $\theta\in\mathcal{B}$, we define a function $q(t,x)$ by

$$\begin{aligned} q(t,x) &= \sup_{\alpha\in\mathcal{A}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0\int_t^T F(\theta_s)ds\right], \\ &= w(t,x) + \mathbb{E}\big[\lambda_0\int_t^T F(\theta_s)ds\big]. \end{aligned}$$

Then $q(t,x)$ is also concave in x for $t\in[0,T]$. We define

$$L(t,x,\mathbf{a},\Theta) := \lambda_0 F(\Theta) + (\mathbf{a}^\intercal(\mu-\mathbf{r})+r)x\frac{\partial q}{\partial x}(t,x) + \frac{1}{2}tr\left(\mathbf{a}^\intercal\Sigma\mathbf{a}x^2\frac{\partial^2 q}{\partial x^2}(t,x)\right).$$

In addition to Assumption 2.2.2, we know L is convex in Θ and concave in $\mathbf{a}$. By Zeidler (2013, Theorem 49.A), there exists a saddle point $(\mathbf{a}^*,\Theta^*)\in A\times B$, such that

$$\inf_{\Theta\in B} L(t,x,\mathbf{a}^*,\Theta) = L(t,x,\mathbf{a}^*,\Theta^*) = \sup_{\mathbf{a}\in A} L(t,x,\mathbf{a},\Theta^*). \tag{A.1}$$

We know from Pham (2009, Theorem 4.3.1) that $q(t,x)$ is a viscosity solution of the HJB equation

$$\frac{\partial q}{\partial t}(t,x) + \sup_{\mathbf{a}\in A} L(t,x,\mathbf{a},\Theta) = 0, \quad q(T,x) = U(x).$$

Then with the fixed process θ^*, whose value is denoted as Θ^* at each instant, the value

$$q^*(t,x) := \sup_{\alpha\in\mathcal{A}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta^*}) + \lambda_0\int_t^T F(\theta_s^*)ds\right]$$

is a viscosity solution of the PDE

$$\frac{\partial q}{\partial t}(t,x) + \sup_{\mathbf{a}\in A} L(t,x,\mathbf{a},\Theta^*) = 0,$$

which is equivalent to

$$\frac{\partial q}{\partial t}(t,x) + \inf_{\Theta \in B} L(t,x,\mathbf{a}^*,\Theta) = 0 \tag{A.2}$$

due to the saddle point property (A.1). Using arguments similar to the ones in Pham (2009, Chapter 4), the function $\inf_{\theta\in\mathcal{B}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha^*,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds\right]$ is the unique viscosity solution of the HJB equation (A.2). Therefore we have

$$\sup_{\alpha\in\mathcal{A}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta^*}) + \lambda_0 \int_t^T F(\theta_s^*)ds\right] = \inf_{\theta\in\mathcal{B}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha^*,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds\right].$$

With $J(t,x,\alpha,\theta) = \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds\right]$, then the inequality

$$\inf_{\theta\in\mathcal{B}}\sup_{\alpha\in\mathcal{A}} J(t,x,\alpha,\theta) \le \sup_{\alpha\in\mathcal{A}} J(t,x,\alpha,\theta^*) = \inf_{\theta\in\mathcal{B}} J(t,x,\alpha^*,\theta) \le \sup_{\alpha\in\mathcal{A}}\inf_{\theta\in\mathcal{B}} J(t,x,\alpha,\theta) \tag{A.3}$$

implies

$$\inf_{\theta\in\mathcal{B}}\sup_{\alpha\in\mathcal{A}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds\right] = \sup_{\alpha\in\mathcal{A}}\inf_{\theta\in\mathcal{B}} \mathbb{E}^{t,x}\left[U(X_T^{\alpha,\theta}) + \lambda_0 \int_t^T F(\theta_s)ds\right].$$

From Proposition 2.3.1, we have $\underline{u}(t,x) \le \underline{v}(t,x) \le \overline{v}(t,x) \le \overline{u}(t,x)$. Combining this with $\overline{u}(t,x) = \underline{u}(t,x)$, we obtained the required equalities

$$\underline{u}(t,x) = \underline{v}(t,x) = \overline{v}(t,x) = \overline{u}(t,x).$$

□

A.2

Proof of Proposition 2.3.3.

Let $X_T^{\theta,\Gamma}$ and $\overline{X}_T^{\theta,\Gamma}$ be the solutions of the SDE (2.2) with initial states (t,x) and $(t,\overline{x})$ respectively, they are both controlled by an arbitrary pair of admissible control and strategy processes (θ,Γ). From Assumption 2.2.1, we have

$$\left|U(X_T^{\theta,\Gamma}) - U(\overline{X}_T^{\theta,\Gamma})\right| \le Q(|X_T^{\theta,\Gamma}|, \left|\overline{X}_T^{\theta,\Gamma}\right|)\left|X_T^{\theta,\Gamma} - \overline{X}_T^{\theta,\Gamma}\right|.$$

We have

$$\left|\mathbb{E}^{t,x}\left[U(X_T^{\theta,\Gamma})\right] - \mathbb{E}^{t,\overline{x}}\left[U(\overline{X}_T^{\theta,\Gamma})\right]\right| \le \mathbb{E}\left|U(X_T^{\theta,\Gamma}) - U(\overline{X}_T^{\theta,\Gamma})\right|. \tag{A.4}$$

By the Cauchy-Schwarz inequality,

$$\left(\mathbb{E}\left|U(X_T^{\theta,\Gamma}) - U(\overline{X}_T^{\theta,\Gamma})\right|\right)^2 \le \mathbb{E}\left[Q(|X_T^{\theta,\Gamma}|, |\overline{X}_T^{\theta,\Gamma}|)^2\right]\mathbb{E}\left[\left|X_T^{\theta,\Gamma} - \overline{X}_T^{\theta,\Gamma}\right|^2\right]. \tag{A.5}$$

It is straightforward to check that there exist constants C, m_1, m_2 and β_0 such that

$$\mathbb{E}\left[Q(|X_T^{\theta,\Gamma}|, |\overline{X}_T^{\theta,\Gamma}|)^2\right]\mathbb{E}\left[\left|X_T^{\theta,\Gamma} - \overline{X}_T^{\theta,\Gamma}\right|^2\right] \le C\mathbb{E}\left[\left|X_T^{\theta,\Gamma}\right|^{2m_1} + \left|\overline{X}_T^{\theta,\Gamma}\right|^{2m_2}\right] e^{2\beta_0(T-t)}\left|x - \overline{x}\right|^2. \tag{A.6}$$

By the classical inequality $\mathbb{E}^{t,x}\Big[\max_{t\leq s\leq T}\left|X_s^{\theta,\Gamma}\right|^{2m}\Big] \leq C_T(1+x^{2m})$ (e.g., Pham (2009, Theorem 1.3.15)), for arbitrary control and strategy processes Γ, θ, we have

$$\left|\mathbb{E}^{t,x}\left[U(X_T^{\theta,\Gamma})\right] - \mathbb{E}^{t,\overline{x}}\left[U(\overline{X}_T^{\theta,\Gamma})\right]\right| \leq \Phi(|x|,|\overline{x}|)\,|x-\overline{x}|\,, \tag{A.7}$$

where C_T, m are constants, and Φ is a polynomial function.

Next, for all bounded functions $\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big]$ and $\mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]$ (by assuming the initial wealth x is finite, they are locally bounded because of Theorem 1.2.2),

$$\begin{aligned}
&\left|\inf_\theta \mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \inf_\theta \mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]\right| \\
\leq\ &\sup_\theta \left|\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]\right|, \qquad \text{(A.8)}\\
&\left|\sup_\Gamma \mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \sup_\Gamma \mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]\right| \\
\leq\ &\sup_\Gamma \left|\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]\right|. \qquad \text{(A.9)}
\end{aligned}$$

Under Assumption 2.2.1, $\overline{v}(t,x)$ is bounded. Then we can write the difference between the two value functions as:

$$\begin{aligned}
&\left|\overline{v}(t,x) - \overline{v}(t,\overline{x})\right| \qquad \text{(A.10)}\\
\leq\ &\sup_\Gamma \sup_\theta \left|\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \mathbb{E}^{t,\overline{x}}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(\overline{X}_T^{\Gamma,\theta})\Big]\right|. \qquad \text{(A.11)}
\end{aligned}$$

In addition to the inequality (A.7), the value function $\overline{v}(t,x)$ is locally Lipschitz continuous in x. □

A.3

Proof of Theorem 2.3.1.

We use localization techniques here. Let $B_k = \{x \in \mathbb{R}, x^2 < k^2\}$, let $\phi_k(x) \in C_b^\infty(\mathbb{R})$ be a function such that $\phi_k(x) = 1$ on B_k, and $\phi_k(x) = 0$ outside B_{k+1}, and the Lipschitz constant of ϕ_k is less than 2. Then we can define a new process

$$dX_s^k = \phi_k(X_s^k)X_s^k\Big[(\alpha_s^\intercal \mu_s + r - \alpha_s^\intercal \mathbf{r})ds + \alpha_s^\intercal \sigma_s dW_s\Big], \tag{A.12}$$

starting from an initial condition $(t,x) \in [0,T] \times \mathbb{R}$. Let $U^k(x) = \phi_{k+2}(x)U(x)$, then we can define the truncated value function by

$$\overline{v}^k(t,x) = \sup_{\Gamma\in\mathcal{N}} \inf_{\theta\in\mathcal{B}} \Big\{\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U^k(X_T^{k,\Gamma,\theta})\Big]\Big\}. \tag{A.13}$$

In the above setting, the drift and volatility functions in the SDE (A.12) are bounded, and the utility function in (A.13) is bounded and Lipschitz continuous. Since all assumptions of Fleming and Souganidis (1989) are satisfied, the localized value function $\overline{v}^k$ defined in (A.13) satisfies the dynamic programming principle: for $t \leq t+h \leq T$,

$$\overline{v}^k(t,x) = \sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}}\Big\{\mathbb{E}^{t,x}\Big[\lambda_0\int_t^{t+h}F(\theta_s)ds + \overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta})\Big]\Big\}. \tag{A.14}$$

In this proof, $X_{t+h}^{k,\Gamma,\theta}$ and $X_{t+h}^{\Gamma,\theta}$ are the solutions of SDE (A.12) and SDE (2.2) respectively, both starting from (t,x), controlled by processes Γ, θ for the time $u \in [t, t+h]$.

As $k \to \infty$, $\overline{v}^k(t,x)$ defined in (A.13) approaches $\overline{v}(t,x)$ defined in (2.7), then our problem reduces to proving that the right hand side of (A.14) converges to the right hand side of (2.11).

Note that if X_s^k is in $\overline{B_{k+1}}$, then X_u^k is in $\overline{B_{k+1}}$ $\forall u \in [s,T]$ almost surely. Define τ_k to be the first exit time of X_t^k from B_k. Thus, for $(t,x) \in [0,T]\times\mathbb{R}$, we have

$$\begin{aligned}
&\Big|\sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}}\Big\{\mathbb{E}^{t,x}\Big[\lambda_0\int_t^{t+h}F(\theta_s)ds + \overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta})\Big]\Big\} && \text{(A.15)}\\
&\quad - \sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}}\Big\{\mathbb{E}^{t,x}\Big[\lambda_0\int_t^{t+h}F(\theta_s)ds + \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\Big]\Big\}\Big|\\
&\leq \sup_{\Gamma\in\mathcal{N}}\sup_{\theta\in\mathcal{B}}\mathbb{E}^{t,x}\Big|\overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta}) - \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\Big|\\
&\leq \sup_{\Gamma\in\mathcal{N}}\sup_{\theta\in\mathcal{B}}\mathbb{E}^{t,x}\Big|\Big(\overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta}) - \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\Big)\mathbb{1}(\tau_k > T)\Big| && \text{(A.16)}\\
&\quad + \sup_{\Gamma\in\mathcal{N}}\sup_{\theta\in\mathcal{B}}\mathbb{E}^{t,x}\Big|\Big(\overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta}) - \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\Big)\mathbb{1}(\tau_k \leq T)\Big|. && \text{(A.17)}
\end{aligned}$$

If $\tau_k > T$, the term (A.16) is zero. For the term (A.17), for any arbitrary pair $(\overline{\Gamma}, \overline{\theta})$, we have

$$\begin{aligned}
&\Big(\mathbb{E}^{t,x}\Big|\overline{v}^k(t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}) - \overline{v}(t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}})\mathbb{1}(\tau_k \leq T)\Big|\Big)^2\\
&\leq \mathbb{E}^{t,x}\Big[\Big|\overline{v}^k(t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}) - \overline{v}(t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}})\Big|^2\Big] \times \mathbb{P}(\tau_k \leq T). && \text{(A.18)}
\end{aligned}$$

Finally our task is to show that the upper bound (A.18) converges to zero as k goes to infinity.

Let $X_T^{k,\Gamma,\theta}$ be the solution of SDE (A.12) starting from $(t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}})$, and $X_T^{\Gamma,\theta}$ be the solution of (2.2) starting from $(t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}})$, they are controlled by Γ, θ for the time $u \in [t+h, T]$.

Using arguments in equations (A.8) and (A.9),

$$\begin{aligned}
&\Big|\overline{v}^k(t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}) - \overline{v}(t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}})\Big|\\
&\leq \sup_{\Gamma\in\mathcal{N}}\sup_{\theta\in\mathcal{B}}\Big|\mathbb{E}^{t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}}\Big[\lambda_0\int_t^T F(\theta_s)ds + U^k(X_T^{k,\Gamma,\theta})\Big] - \mathbb{E}^{t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}}}\Big[\lambda_0\int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big]\Big|.
\end{aligned}$$

For any arbitrary controls (Γ, θ) for the time $u \in [t+h, T]$, it is easy to see that

$$\begin{aligned}
&\mathbb{E}^{t,x}\Big(\Big|\mathbb{E}^{t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}}\Big[U^k(X_T^{k,\Gamma,\theta})\Big] - \mathbb{E}^{t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}}}\Big[U(X_T^{\Gamma,\theta})\Big]\Big|\Big) \\
&\leq \mathbb{E}^{t,x}\Big(\mathbb{E}^{t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}}\Big|U^k(X_T^{k,\Gamma,\theta})\Big| + \mathbb{E}^{t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}}}\Big|U(X_T^{\Gamma,\theta})\Big|\Big) \\
&\leq \mathbb{E}^{t,x}\Big(\mathbb{E}^{t+h, X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}}\Big[C\Big|X_T^{k,\Gamma,\theta}\Big|^{2m_1}\Big] + \mathbb{E}^{t+h, X_{t+h}^{\overline{\Gamma},\overline{\theta}}}\Big[C\Big|X_T^{\Gamma,\theta}\Big|^{2m_2}\Big]\Big) \\
&\leq \mathbb{E}^{t,x}\Big(K_T\big(1 + \Big|X_{t+h}^{k,\overline{\Gamma},\overline{\theta}}\Big|^{2m_1}\big) + K_T\big(1 + \Big|X_{t+h}^{\overline{\Gamma},\overline{\theta}}\Big|^{2m_2}\big)\Big) \\
&\leq C_T\Big(1 + |x|^{2m}\Big),
\end{aligned}$$

where C, K_T, C_T, m_1, m_2, m are constants. Then there exists a polynomial Φ such that

$$\mathbb{E}^{t,x}\left[\left|\overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta}) - \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\right|^2\right] \leq \Phi(|x|), \tag{A.19}$$

and the Markov inequality yields

$$\mathbb{P}(\tau_k \leq T) \leq \frac{\mathbb{E}^{t,x}\left[\sup_{t \leq s \leq T}\left|X_s^{\Gamma,\theta}\right|^2\right]}{k^2} \leq \frac{C_T\,(1+x^2)}{k^2}, \tag{A.20}$$

where C_T is a constant independent of k. Therefore we have

$$\mathbb{E}^{t,x}\left|(\overline{v}^k(t+h, X_{t+h}^{k,\Gamma,\theta}) - \overline{v}(t+h, X_{t+h}^{\Gamma,\theta}))\mathbb{1}(\tau_k \leq T)\right| \leq \frac{K(\left|x\right|)}{k},$$

where $K(\left|x\right|)$ is a polynomial function in terms of x.

As $k \to \infty$, the term (A.17) goes to zero as well, therefore

$$\overline{v}(t,x) = \sup_{\Gamma \in \mathcal{N}} \inf_{\theta \in \mathcal{B}} \left\{\mathbb{E}^{t,x}\left[\lambda_0 \int_t^{t+h} F(\theta_s)ds + \overline{v}(t+h, X_{t+h}^{\Gamma,\theta})\right]\right\},$$

as the left and right hand sides of (A.14) converge to the left and right hand sides of equation (2.11) respectively. □

A.4

Proof of Corollary 2.3.1.

Let $X_s^{\Gamma,\theta}$ be the solution of the SDE (2.2) starting from x at time t, controlled by Γ, θ for time $u \in [t,s]$. By the Dynamic Programming Principle and inequality (A.7), for $t < s < T$,

$$\left|\overline{v}(t,x) - \overline{v}(s,x)\right| = \left|\sup_{\Gamma \in \mathcal{N}} \inf_{\theta \in \mathcal{B}} \left\{\mathbb{E}^{t,x}\left[\lambda_0 \int_t^s F(\theta_u)du + \overline{v}(s, X_s^{\Gamma,\theta})\right]\right\} - \overline{v}(s,x)\right|.$$

With any arbitrary control and strategy processes $(\hat{\Sigma}, \hat{\Gamma})$ for time $u \in [t,s]$, we have

$$\begin{aligned}
&\left|\mathbb{E}^{t,x}\left[\lambda_0 \int_t^s F(\hat{\theta}_u)du + \overline{v}(s, X_s^{\hat{\Gamma},\hat{\theta}})\right] - \overline{v}(s,x)\right| && \text{(A.21)} \\
&= \left|\mathbb{E}^{t,x}\left[\lambda_0 \int_t^s F(\hat{\theta}_u)du\right] + \mathbb{E}^{t,x}\left[\overline{v}(s, X_s^{\hat{\Gamma},\hat{\theta}}) - \overline{v}(s,x)\right]\right| && \text{(A.22)} \\
&\leq \left|\mathbb{E}^{t,x}\left[\lambda_0 \int_t^s F(\hat{\theta}_u)du\right]\right| + \mathbb{E}^{t,x}\left[\sup_{\Gamma}\sup_{\theta}\left(\mathbb{E}^{s, X_s^{\hat{\Gamma},\hat{\theta}}}[U(X_T^{\Gamma,\theta})] - \mathbb{E}^{s,x}[U(X_T^{\Gamma,\theta}]\right)\right]. && \text{(A.23)}
\end{aligned}$$

Referring to (A.7), there exist a polynomial function Φ and constants C, C_T, m_1, m_2 for an arbitrary pair of (Γ, θ) for time $u \in [s, T]$ such that

$$\begin{aligned}
&\mathbb{E}^{t,x}\left[\mathbb{E}^{s,X_s^{\hat{\Gamma},\hat{\theta}}}[U(X_T^{\Gamma,\theta})] - \mathbb{E}^{s,x}[U(X_T^{\Gamma,\theta})]\right] \\
\leq\ &\mathbb{E}^{t,x}\left[\Phi(\left|X_s^{\hat{\Gamma},\hat{\theta}}\right|, |x|)\left|X_s^{\hat{\Gamma},\hat{\theta}} - x\right|\right] \\
\leq\ &\mathbb{E}^{t,x}\left[\Phi(\left|X_s^{\hat{\Gamma},\hat{\theta}}\right|, |x|)^2\right]^{1/2}\mathbb{E}^{t,x}\left[\left|X_s^{\hat{\Gamma},\hat{\theta}} - x\right|^2\right]^{1/2} \\
\leq\ &\left(Cx^{2m_1} + C_T\left(1 + x^{2m_2}\right)\right)^{1/2}\mathbb{E}^{t,x}\left[\left|X_s^{\hat{\Gamma},\hat{\theta}} - x\right|^2\right]^{1/2}.
\end{aligned}$$

We know

$$\mathbb{E}^{t,x}\left[\left|X_s^{\hat{\Gamma},\hat{\theta}} - x\right|^2\right] \leq C_T(1 + x^2)(s - t).$$

Let $\eta = \max\left\{\left|F(\theta_u)\right| : \theta_u \in B\right\}$, therefore

$$\left|\overline{v}(t, x) - \overline{v}(s, x)\right| \leq \lambda_0\eta(s - t) + \Phi(|x|)(s - t)^{1/2}.$$

Hence $\overline{v}(t, x)$ is Hölder continuous in $t \in [0, T]$. □

A.5

Proof of Theorem 2.3.2.

We again make use of the localized processes X_t^k, U^k and $\overline{v}^k$ from the proof of Theorem 2.3.1 in section 2.3.2. The HJBI equation associated with SDE (A.12) is

$$\begin{cases} \frac{\partial v}{\partial t}(t, x) + H^k(t, x, \frac{\partial v}{\partial x}(t, x), \frac{\partial^2 v}{\partial x^2}(t, x)) = 0 & \text{in } [0, T) \times \mathbb{R}, \\ v(T, x) = U^k(x) & \text{on } [T] \times \mathbb{R}, \end{cases} \tag{A.24}$$

where

$$H^k(t, x, p, M) = \inf_{\Theta \in B}\sup_{\mathbf{a} \in A}\left\{\lambda_0 F(\Theta) + \phi_k(x)(\mathbf{a}^\intercal\mu + r - \mathbf{a}^\intercal\mathbf{r})xp + \frac{1}{2}tr\left(\phi_k^2(x)\mathbf{a}^\intercal\Sigma\mathbf{a}x^2 M\right)\right\}. \tag{A.25}$$

All the assumptions in Fleming and Souganidis (1989) are satisfied, so $\overline{v}^k(t, x)$ (A.13) is a viscosity solution of the HJBI equation (A.24).

Now we introduce another value function

$$\tilde{v}^k(t, x) = \begin{cases} \overline{v}^k(t, x) & \forall (t, x) \in [0, T] \times B_{k+1} \\ U(x) + \inf_{\theta \in \mathcal{B}}\mathbb{E}^{t,x}[\lambda_0\int_t^T F(\theta_s)ds] & \forall (t, x) \in [0, T] \times (\mathbb{R}\backslash B_{k+1}) \end{cases}.$$

In the first case where $x \in B_{k+1}$, we have $\left(X_T^k\right)^2 < (k + 2)^2$ almost surely. Therefore

$$\tilde{v}^k(t, x) = \sup_{\Gamma \in \mathcal{N}}\inf_{\theta \in \mathcal{B}}\mathbb{E}^{t,x}[\lambda_0\int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})],\ \ \forall (t, x) \in [0, T] \times B_{k+1}.$$

Then $\tilde{v}^k(t, x), \forall (t, x) \in [0, T] \times B_{k+1}$ is a viscosity solution of

$$\begin{cases} \frac{\partial v}{\partial t}(t, x) + H^k(t, x, \frac{\partial v}{\partial x}(t, x), \frac{\partial^2 v}{\partial x^2}(t, x)) = 0 & \text{in } [0, T) \times \mathbb{R}, \\ v(T, x) = U(x) & \text{on } [T] \times \mathbb{R}. \end{cases} \tag{A.26}$$

Since the drift and diffusion of X_t^k are zero outside of B_{k+1}, then $X_T^{k,\Gamma,\theta} = x$ for $x \in (\mathbb{R}\backslash B_{k+1})$ and

$$\tilde{v}^k(t,x) = \sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}} \mathbb{E}^{t,x}[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})],\ \ \forall(t,x) \in [0,T]\times(\mathbb{R}\backslash B_{k+1})\,.$$

It is easy to check that $\tilde{v}^k(t,x), \forall(t,x) \in [0,T]\times(\mathbb{R}\backslash B_{k+1})$ is also a viscosity solution of HJBI equation (A.26) with $\phi_k(x) = 0$. Combining the two cases, we have

$$\tilde{v}^k(t,x) = \sup_{\Gamma\in\mathcal{N}}\inf_{\theta\in\mathcal{B}} \mathbb{E}^{t,x}[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})], \quad \text{on } [0,T]\times\mathbb{R},$$

and $\tilde{v}^k(t,x)$ is a viscosity solution of (A.26).

Since $H^k \to H$ as $k \to \infty$, if we can prove $\tilde{v}^k \to \overline{v}$ as $k \to \infty$, then it shows that $\overline{v}$ is a viscosity solution of equation (2.12). We will prove the convergence in the following way: first of all, we have

$$\left|\overline{v} - \tilde{v}^k\right| \le \sup_{\Gamma\in\mathcal{N}}\sup_{\theta\in\mathcal{B}}\left|\mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\Big] - \mathbb{E}^{t,x}\Big[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})\Big]\right|.$$

For any arbitrary pair of control and strategy processes (Γ,θ), we have

$$\mathbb{E}^{t,x}\Big[\Big(\big(\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta})\big) - \big(\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})\big)\Big)\mathbb{1}(\tau_k \ge T)\Big] = 0. \quad (A.27)$$

Using Assumption 2.2.1, we can write

$$\begin{aligned}&\left|\mathbb{E}^{t,x}[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{\Gamma,\theta}])] - \mathbb{E}^{t,x}[\lambda_0 \int_t^T F(\theta_s)ds + U(X_T^{k,\Gamma,\theta})]\right| \\ &= \left|\mathbb{E}^{t,x}\Big[\big(U(X_T^{\Gamma,\theta}) - U(X_T^{k,\Gamma,\theta})\big)\mathbb{1}(\tau_k < T)\Big]\right| \\ &\le \mathbb{E}^{t,x}\Big[Q(\left|X_T^{\Gamma,\theta}\right|, \left|X_T^{k,\Gamma,\theta}\right|)(\left|X_T^{\Gamma,\theta}\right| - \left|X_T^{k,\Gamma,\theta}\right|)\mathbb{1}(\tau_k < T)\Big]. \end{aligned} \quad (A.28)$$

Applying the Cauchy-Schwarz inequality on the upper bound (A.28), with similar arguments in (A.20), we obtain

$$\begin{aligned}&\left(\mathbb{E}^{t,x}\Big[Q(\left|X_T^{\Gamma,\theta}\right|, \left|X_T^{k,\Gamma,\theta}\right|)(\left|X_T^{\Gamma,\theta}\right| - \left|X_T^{k,\Gamma,\theta}\right|)\mathbb{I}(\tau_k < T)\Big]\right)^2 \\ &\le C_T\left(1 + x^{2m}\right) \times \frac{C_T\left(1+x^2\right)}{k^2}. \end{aligned} \quad (A.29)$$

Hence

$$\mathbb{E}^{t,x}\Big[Q(\left|X_T^{\Gamma,\theta}\right|, \left|X_T^{k,\Gamma,\theta}\right|)(\left|X_T^{\Gamma,\theta}\right| - \left|X_T^{k,\Gamma,\theta}\right|)\mathbb{1}(\tau_k < T)\Big] \le \frac{\Phi(|x|)}{k}, \quad (A.30)$$

where $\Phi(|x|)$ is a polynomial function independent of k. Since (Γ,θ) are arbitrary, combining (A.27), (A.28) and (A.30), we deduce that

$$\left|\overline{v} - \tilde{v}^k\right| \le \frac{\Phi(|x|)}{k}.$$

So $\tilde{v}^k$ converges to $\overline{v}$ as $k \to \infty$. Thus $\overline{v}$ is a viscosity solution of the HJBI equation (2.12). □

A.6

Proof of Theorem 2.3.3.

For the case $\lambda \neq 0$, we can use the dedoubling variable technique and Ishii's lemma to prove that $U(t,x) \leq V(t,x)$ on $[0,T] \times \mathbb{R}$. This proof is a natural adaptation of Pham (2009), interested readers can refer to it. Here we illustrate the specific case $\lambda = 0$ in the PDE (2.14).

We multiply the penalty function in equation (2.14) by $e^{\lambda t}$, then we have

$$-\frac{\partial v}{\partial t}(t,x) + \lambda v(t,x) - \inf_{\Theta \in B} \sup_{\mathbf{a} \in A} \Big\{ e^{\lambda t} \lambda_0 F(\Theta) + (\mathbf{a}^\intercal(\mu - \mathbf{r}) + r)x \frac{\partial v}{\partial x}(t,x) + \frac{1}{2} tr\left(\mathbf{a}^\intercal \Sigma \mathbf{a} x^2 \frac{\partial^2 v}{\partial x^2}(t,x)\right)\Big\} = 0, \quad (t,x) \in [0,T) \times \mathbb{R}, \lambda \in \mathbb{R}\backslash\{0\}. \quad (A.31)$$

We assume w.l.o.g. that $\lambda > 0$. We have $U(t,x)$ and $V(t,x)$ as the subsolution and supersolution of (A.31) and $U(T,x) \leq V(T,x)$. Choose an arbitrary test function $e^{\lambda t}\phi \in C^\infty([0,T) \times \mathbb{R})$ and $(t_0,x_0) \in [0,T) \times \mathbb{R}$ be the maximum point of $U - e^{\lambda t}\phi$, let $(U - e^{\lambda t}\phi)(t_0,x_0) = 0$. By the viscosity subsolution property of $U(t,x)$, we have

$$\begin{aligned}
& -\lambda e^{\lambda t}\phi(t_0,x_0) - e^{\lambda t}\frac{\partial \phi}{\partial t}(t_0,x_0) + \lambda e^{\lambda t}\phi(t_0,x_0) \\
& - \inf_{\Theta \in B} \sup_{\mathbf{a} \in A} \left\{ e^{\lambda t}\lambda_0 F(\Theta) + e^{\lambda t}(\mathbf{a}^\intercal(\mu - \mathbf{r}) + r)x\frac{\partial \phi}{\partial x}(t_0,x_0) + e^{\lambda t}\frac{1}{2} tr\left(\mathbf{a}^\intercal \Sigma \mathbf{a} x^2 \frac{\partial^2 \phi}{\partial x^2}(t_0,x_0)\right)\right\} \\
=& e^{\lambda t}\left[-\frac{\partial \phi}{\partial t}(t_0,x_0) - \inf_{\Theta \in B} \sup_{\mathbf{a} \in A}\left\{\lambda_0 F(\Theta) + (\mathbf{a}^\intercal(\mu - \mathbf{r}) + r)x\frac{\partial \phi}{\partial x}(t_0,x_0) + \frac{1}{2} tr\left(\mathbf{a}^\intercal \Sigma \mathbf{a} x^2 \frac{\partial^2 \phi}{\partial x^2}(t_0,x_0)\right)\right\}\right] \\
\leq & 0. \qquad (A.32)
\end{aligned}$$

Let $\tilde{U}(t,x) = e^{-\lambda t}U(t,x)$ and $\tilde{V}(t,x) = e^{-\lambda t}V(t,x)$. Note that (t_0,x_0) is also the maximum point of $\tilde{U} - \phi$. The inequality (A.32) implies that

$$-\frac{\partial \phi}{\partial t}(t_0,x_0) - \inf_{\Theta \in B} \sup_{\mathbf{a} \in A}\left\{\lambda_0 F(\Theta) + (\mathbf{a}^\intercal(\mu - \mathbf{r}) + r)x\frac{\partial \phi}{\partial x}(t_0,x_0) + \frac{1}{2} tr\left(\mathbf{a}^\intercal \Sigma \mathbf{a} x^2 \frac{\partial^2 \phi}{\partial x^2}(t_0,x_0)\right)\right\} \leq 0.$$

Hence $\tilde{U}(t,x)$ is a subsolution of

$$-\frac{\partial v}{\partial t}(t,x) - \inf_{\Theta \in B} \sup_{\mathbf{a} \in A}\left\{\lambda_0 F(\Theta) + (\mathbf{a}^\intercal(\mu - \mathbf{r}) + r)x\frac{\partial v}{\partial x}(t,x) + \frac{1}{2} tr\left(\mathbf{a}^\intercal \Sigma \mathbf{a} x^2 \frac{\partial^2 v}{\partial x^2}(t,x)\right)\right\} = 0, \quad (t,x) \in [0,T) \times \mathbb{R}. \quad (A.33)$$

A similar calculation applies to $\tilde{V}(t,x)$ and $\tilde{V}(t,x)$ is a supersolution of (A.33). From the proof for the case $\lambda \neq 0$, we know $U(t,x) \leq V(t,x)$ on $(t,x) \in [0,T] \times \mathbb{R}$. Therefore we conclude that $\tilde{U}(t,x) \leq \tilde{V}(t,x)$ on $[0,T] \times \mathbb{R}$. This completes the proof. □

A.7

Explicit solution of equation (2.23)**:**

For completeness, we express the real positive root of equation (2.23) explicitly. Let $c = \dfrac{(\mu - r)^2}{2\lambda_0}$, the discriminant of the equation $\Delta = -256c^3 - 27\sigma_0^4 c^2$ is less than zero, meaning there are two distinct real roots. It is easy to check that there is one positive and one negative root, and the real positive one is

$$
\begin{aligned}
\hat{\sigma} = \frac{\sigma_0}{4} + \frac{1}{2}\left[\frac{\sigma_0^2}{4} + \frac{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}}\right]^{\frac{1}{2}} \\
+\frac{1}{2}\Biggl[\frac{\sigma_0^2}{2} - \frac{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}}{\sqrt[3]{2}3^{2/3}} + \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}} \\
+\frac{\sigma_0^3}{4\sqrt{\frac{\sigma_0^2}{4} + \frac{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{\sqrt{3}\sqrt{27\sigma_0^4c^2+256c^3}-9\sigma_0^2c}}}}\Biggr]^{\frac{1}{2}}.
\end{aligned}
$$

Appendix B

Appendix for Chapter 3

B.1

Proof of proposition 3.2.1.

Firstly, we prove the necessity. We can use eigen-decomposition and write the covariance matrix $\Sigma(t,x) = Q\Lambda Q^\intercal = Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^\intercal$, where $Q \in \mathbb{R}^{d\times d}$ and the ith column of Q is the eigenvector q_i of $\Sigma(t,x)$, and $\Lambda \in \mathbb{R}^{d\times d}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$.

For any given $\alpha(t,x) \in \mathbb{R}^d$, $A = \alpha(t,x)^\intercal Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^\intercal\alpha(t,x)x^2\rho$. We define $\beta \coloneqq \alpha(t,x)^\intercal Q\Lambda^{\frac{1}{2}}$, then $A = \beta\beta^\intercal x^2\rho = \|\beta\|^2 x^2\rho$, where $\|\cdot\|$ denotes the L^2 norm. Similarly,

$B = \alpha(t,x)^\intercal Q\Lambda^{\frac{1}{2}}(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)x\rho = \beta(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)x\rho$. Therefore we have the relationship between A and B as

$$B^2 = \left(\beta(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)\right)^2 x^2\rho^2 \le \|\beta\|^2 \left\|(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)\right\|^2 x^2\rho^2 = A\rho\left\|(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)\right\|^2,$$

$$\left\|(Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x)\right\|^2 A \ge \frac{B^2}{\rho}.$$

Define $\nu(t,x) \coloneqq (Q\Lambda^{\frac{1}{2}})^{-1}\mu(t,x) = \Sigma(t,x)^{-\frac{1}{2}}\mu(t,x)$, we can write the above inequality as $A \ge \frac{B^2}{\|\nu(t,x)\|^2\rho}$.

For given (ρ, B, A) satisfying $A \ge \frac{B^2}{\|\nu(t,x)\|^2\rho}$, we want to show that there exists $\alpha(t,x) \in \mathbb{R}^d (d > 1)$, such that $A = \alpha(t,x)^\intercal\Sigma(t,x)\alpha(t,x)x^2\rho$, $B = \alpha(t,x)^\intercal\mu(t,x)x\rho$. First of all, since $\frac{A}{\rho} \ge 0$, there exists a vector $\beta \in \mathbb{R}^{1\times d}$ whose norm satisfies $\|\beta\|^2 = \frac{A}{x^2\rho}$. Then $\frac{B^2}{\|\nu(t,x)\|^2\rho} \le A$ will be equivalent to $B^2 \le \|\beta\|^2\|\nu(t,x)\|^2 x^2\rho^2$. With Cauchy–Schwarz inequality, $(\beta\nu(t,x))^2x^2\rho^2 \le \|\beta\|^2\|\nu(t,x)\|^2x^2\rho^2$ holds. Therefore there exists a vector $\beta \in \mathbb{R}^{1\times d}(d > 1)$ such that $B = \beta\nu(t,x)x\rho$ and $\|\beta\|^2 = \frac{A}{x^2\rho}$. With this β, there exists an $\alpha(t,x) = Q\Lambda^{-\frac{1}{2}}\beta^\intercal$.

The case for dimension $d = 1$ is trivial, hence omitted here. □

B.2

Proposition B.2.1. We denote K_0 the set of $(u,b,a,r) \in C_b(\mathcal{E}, \mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})$ that can be represented by $\phi \in C_b^{1,2}(\mathcal{E})$. Then we have

$$\begin{aligned}\inf_{(\rho,B,A,\rho_1)\in C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} &\{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\} \\ &= \inf_{(\rho,B,A,\rho_1)\in\mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})}\{\alpha^*(\rho,B,A,\rho_1) + \beta^*(\rho,B,A,\rho_1)\}. \quad \text{(B.1)}\end{aligned}$$

Proof. Following closely the argument in Villani (2003, Section 1.3), we define $C_0(\mathcal{E})$ the space of continuous functions on $\mathcal{E}$, going to 0 at infinity. For $(\rho, B, A, \rho_1) \in C_b^*(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})$, we decompose $(\rho, B, A, \rho_1) = (\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) + (\delta\rho, \delta B, \delta A, \delta\rho_1)$, where $(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) \in \mathcal{M}(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})$. For any $(u, b, a, r) \in C_0(\mathcal{E}; \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})$, we have

$$\langle (u, b, a, r), (\delta\rho, \delta B, \delta A, \delta\rho_1) \rangle = 0.$$

Because $\mathcal{M}(\mathcal{E}; \mathbb{R})$ is a subset of $C_b^*(\mathcal{E}; \mathbb{R})$, we naturally have

$$\begin{aligned}
&\inf_{(\rho,B,A,\rho_1) \in C_b^*(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho, B, A, \rho_1) + \beta^*(\rho, B, A, \rho_1)\} \\
&\qquad \leq \inf_{(\rho,B,A,\rho_1) \in \mathcal{M}(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \{\alpha^*(\rho, B, A, \rho_1) + \beta^*(\rho, B, A, \rho_1)\}. \qquad \text{(B.2)}
\end{aligned}$$

Notice that the right hand side of the inequality (B.2) can be written as

$$\inf_{(\hat{\rho},\hat{B},\hat{A},\hat{\rho}_1) \in \mathcal{M}} \left\{ \alpha^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) + \beta^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) \right\}.$$

Now we look at the opposite direction of inequality (B.2). For α^*, we have

$$\begin{aligned}
&\alpha^*(\rho, B, A, \rho_1) \\
&= \sup_{(u,b,a,r) \in C_b(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{ \int_{\mathcal{E}} u d\rho + b dB + a dA + \left[\int_{\mathbb{R}} r d\rho_1 - C^*(r) \right] : u + F^*(b, a) \leq 0 \right\} \\
&\geq \sup_{(u,b,a,r) \in C_0(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{ \int_{\mathcal{E}} u d\rho + b dB + a dA + \left[\int_{\mathbb{R}} r d\rho_1 - C^*(r) \right] : u + F^*(b, a) \leq 0 \right\} \\
&= \sup_{(u,b,a,r) \in C_0(\mathcal{E};\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R})} \left\{ \int_{\mathcal{E}} u d\hat{\rho} + b d\hat{B} + a d\hat{A} + \left[\int_{\mathbb{R}} r d\hat{\rho}_1 - C^*(r) \right] : u + F^*(b, a) \leq 0 \right\} \\
&= \alpha^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1).
\end{aligned}$$

We know $\beta^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) = 0$ if $(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1)$ satisfies (3.23), and $\beta^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) = +\infty$ otherwise. When β^* is finite,

$$\int_{\mathcal{E}} u d\hat{\rho} + b d\hat{B} + a d\hat{A} + \int_{\mathbb{R}} r d\hat{\rho}_1 - \phi_0 d\rho_0 = 0 \qquad \forall (u, b, a, r) \in K_0.$$

Then we have

$$\begin{aligned}
\beta^*(\hat{\rho}, \hat{B}, \hat{A}, \hat{\rho}_1) &= \sup_{(u,b,a,r) \in C_0 \cap K_0} \int_{\mathcal{E}} u d\hat{\rho} + b d\hat{B} + a d\hat{A} + \int_{\mathbb{R}} r d\hat{\rho}_1 - \phi_0 d\rho_0 \\
&= \sup_{(u,b,a,r) \in C_0 \cap K_0} \int_{\mathcal{E}} u d\rho + b dB + a dA + \int_{\mathbb{R}} r d\rho_1 - \phi_0 d\rho_0 \\
&\leq \sup_{(u,b,a,r) \in K_0} \int_{\mathcal{E}} u d\rho + b dB + a dA + \int_{\mathbb{R}} r d\overline{\rho}_1 - \phi_0 d\rho_0 \\
&= \beta^*(\rho, B, A, \rho_1).
\end{aligned}$$

This completes the proof. □