Several Mathematical Problems in Investment Management

by

Ruihong Jiang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Actuarial Science

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Member: Rogemar S. Mamon
Professor
Dept. of Statistical and Actuarial Sciences
Western University

Internal-External Member: Jun Liu
Associate Professor
Dept. of Applied Mathematics
University of Waterloo

Internal Member: Ben Feng
Assistant Professor
Dept. of Statistics and Actuarial Science
University of Waterloo

Bin Li
Associate Professor
Dept. of Statistics and Actuarial Science
University of Waterloo

Supervisors: David Saunders
Associate Professor
Dept. of Statistics and Actuarial Science
University of Waterloo

Chengguo Weng
Professor
Dept. of Statistics and Actuarial Science
University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis studies four mathematical problems in investment management. All four problems arise from practical challenges and are data-driven.

Chapter 2 investigates the Kelly portfolio strategy. The full Kelly strategy's deficiency in the face of estimation errors in practice can be mitigated by fractional or shrinkage Kelly strategies. This chapter provides an alternative, the RL Kelly strategy, based on a reinforcement learning (RL) framework. RL algorithms are developed for the practical implementation of the RL Kelly strategy. Extensive simulation studies are conducted, and the results confirm the superior performance of the RL Kelly strategies.

In Chapter 3, we study the discrete-time mean-variance problem under an RL framework. The continuous-time problem was theoretically studied by the existing literature but was subject to a discretization error in implementations. We compare our discrete-time model with the continuous-time model in terms of theoretical results and numerical performance. In a daily trading market setting, we find both discrete-time and continuous-time models achieve comparable performance. However, the discrete-time model outperforms better than the continuous-time model when the trading is less frequent. Our discrete-time model is not subject to the discretization error.

Chapter 4 explores the valuation problem of large variable annuity (VA) portfolios. A computationally appealing methodology for the valuation of large VA portfolios is a metamodelling framework that evaluates a small set of representative contracts, fits a predictive model based on these computed values, and then extrapolates the model to estimate the values of the remaining contracts. This chapter proposes a new two-phase procedure for selecting representative contracts. The representatives from the first phase are determined using contract attributes as in existing metamodelling approaches, but those in the second phase are chosen by utilizing the information contained in the values of the representatives from the first phase. Two numerical studies confirm that our two-phase selection procedure improves upon conventional approaches from the existing literature.

Chapter 5 focuses on the capture ratio which is a widely-used investment performance measure. We study the statistical problem of estimating the capture ratio based on a finite number of observations of a fund's returns. We derive the asymptotic distribution of the estimator, and use it for testing whether one fund has a capture ratio that is statistically significantly higher than another. We also perform hypothesis tests with real-world hedge fund data. Our analysis raises concerns regarding the models and sample sizes used for estimating capture ratios in practice.

Acknowledgements

I want to express my sincere gratitude to my supervisors, Professor David Saunders and Professor Chengguo Weng, for their guidance, support, and encouragement throughout my research. Their insightful feedback and constructive criticism were instrumental in shaping my research projects.

I would like to thank my examining committee members Professor Rogemar S. Mamon, Professor Jun Liu, Professor Ben Feng, Professor Bin Li and Professor Marius Hofert (thesis proposal committee member) for their precious time and advice.

Many thanks also go to Professor Phelim Boyle, who is my Research Assistantship supervisor at the University of Waterloo. His knowledge and expertise helped me refine my research skills.

I would also like to extend my gratitude to the administrative staff at the Department of Statistics and Actuarial Science: Mary Lou Dufton, Greg Preston, Carla Daniel, and Carlos Mendes, for their help during my study at the University of Waterloo.

Last but not least, I would like to especially thank Keying Xu and my cats Oreo and Mochi. They bring unlimited happiness to my life and help me go through all the difficult times.

Dedication

This thesis is dedicated to my parents (Bin Jiang and Jinju Fang).

This thesis is also in memory of last Dr. Ken Seng Tan. When he was a professor at the University of Waterloo, Dr. Tan was in charge of the exchange student program at the Central University of Finance and Economics, China, through which I came to Waterloo for the first time. Dr. Tan also wrote my recommendation letter for my Ph.D. program which brought me here eventually.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Investment management covers a broad range of topics in quantitative finance and actuarial science. Many of these topics focus on practical problems and hence are data-driven. This thesis investigates four existing tasks of investment management in four separate chapters. All of them arise from real-world scenarios in stock, insurance and fund markets. To study these problems, we build rigorous mathematical and statistical frameworks which are able to work with real-world or simulated data. Below is a summary for each chapter.

Chapter 2 revisits the Kelly criterion problem of portfolio construction (MacLean et al., 2010) which has been well studied with theoretically optimal solutions developed. However, the practical implementation does not show the advantages of the optimal strategy, due to estimation error. Recently, a reinforcement learning (RL) based framework was proposed in the literature (Wang et al., 2019; Wang and Zhou, 2020) as a new approach to improve the empirical performance of the optimal mean-variance strategy. The learning scheme of balancing between exploration and exploitation achieves the mean-variance criterion better than the MLE-based model and deep deterministic policy gradient (DDPG) model, in a continuous-time setting. The RL framework was then applied to several different problems (Wang and Zhou, 2019; Dai et al., 2020; Guo et al., 2022; Firoozi and Jaimungal, 2022). In Chapter 2, we solve the continuous-time Kelly criterion problem with an RL solution. The new RL-based strategy outperforms the MLE-based strategies in different market scenarios. Different from the existing literature mentioned above, we develop an online learning framework. We also test the model convergence in an offline learning framework.

Chapter 3 investigates the RL solution for a discrete-time mean-variance portfolio strategy, in view of the fact that the model in Wang and Zhou (2020) is built in a continuous-time setting but implemented in a discretized scheme. Hence the implementation is affected by the discretization error. In this chapter, we investigate the discretization error by studying the discrete-time mean-variance problem under the RL framework. Through numerical examples, we find that the strategy following the discrete-time setting performs generally better than that discretized from the continuous-time setting.

Chapter 4 explores the problem of valuation for large variable annuities (VA) portfolios. A VA contract has an option-type payoff structure. It is usually not possible to calculate the value or Greeks of a VA contract analytically and one has to resort to simulation for an evaluation. However, to achieve a high level of accuracy, extensive simulations are required and thus computationally demanding. It is more challenging to evaluate a large VA portfolio which could contain more than 10,000 contracts. A popular valuation framework adopts a meta-modelling framework and takes two steps in the evaluation of a large VA portfolio (Gan (2013); Gan and Lin (2015); Hejazi and Jackson (2016); Gan and Valdez (2016, 2018, 2020); Hejazi et al. (2017); Xu et al. (2018); Liu and Tan (2021)). The first step is to select a small portion of contracts: representative contracts, based on the contract attributes. Evaluating such a small set of contracts is relatively computationally manageable. The second step is to build an interpolation model using the evaluation results from those selected representative contracts and estimate the values of the rest contracts in the VA portfolio using the interpolation model. The majority of the literature works on improving the estimation accuracy are subject to computational constraints. This chapter proposes a new selection procedure that has two phases. The motivation for the two-phase selection is to make those selected contracts more representative of the entire portfolio. This two-phase selection is also appealing in practice when there are existing representative contracts and the insurance company wants to select more representatives. In our proposed framework, the first phase (initial selection phase) is similar to the conventional selection of representative contracts in the existing meta-modelling method. The second phase (conditional selection phase) builds a posterior model for the target values (i.e., market values or Greeks) and determines a set of VA contracts that as a whole is most representative according to the established posterior model. We proposed a conditional k-means algorithm, which belongs to branch of machine learning called unsupervised learning, for the selection of the representative contracts in the second phase and proved certain convergence properties of the algorithm. Through two numerical studies, we demonstrate that the valuation procedure with this two-phase selection achieves significant improvement in estimation accuracy compared to the conventional valuation procedure.

Chapter 5 focuses on the capture ratio, a well-known performance measure used in the finance industry for fund management. It measures the relative performance of a fund in both the upside and downside markets. It has been used by ranking agencies such as Morningstar as well as in the literature (Cox and Goff, 2013; Chang and Krueger, 2013; Bello, 2014; Cline and Gilstrap, 2021). However, the heuristic definition has never been theoretically studied, or tested for robustness. We derive the asymptotic (joint) distributions of the capture ratio under different model assumptions. We also probe into the statistical power of hypothesis tests regarding the capture ratio. Our numerical examples raise concerns about the sample size and model assumptions in the practical use of the capture ratio.

Chapter 6 concludes this thesis and summarizes the future work.

The main results of the thesis have been published in journals or preprints. See Jiang et al. (2022a) for Chapter 2; Jiang et al. (2022c) for Chapter 4; Jiang et al. (2022b) for Chapter 5.

Chapter 2

Continuous-Time RL Kelly Criterion Problem

2.1 Introduction

In this chapter, we focus on portfolio selection in the stock market. How to optimize stock trading has always been an important task in practice and literature. In the classical Merton portfolio optimization model (Merton, 1971), an investor aims to maximize her utility by trading stocks and bonds. A common choice of utility function is log-utility. Maximizing the expected log-utility of the terminal portfolio value is the same as maximizing the expected log-return of the portfolio, and such an optimality target is known as the Kelly criterion (MacLean et al., 2010). For this criterion, the portfolio selection problem has been well studied, and closed-form solutions have been found in many models. It is well known that the full Kelly strategy, the optimal allocation strategy under the Kelly criterion, outperforms other strategies in terms of capital growth (MacLean et al., 2010). However, one important risk with the full Kelly strategy is that an investor may have to invest a large amount of money in stocks. This fact could lead to a substantial loss given a sequence of poor market returns.

Furthermore, the full Kelly strategy's optimality is sensitive to estimation errors. When estimation of model parameters is involved, the full Kelly strategy's empirical performance typically has a considerable deviation from the theoretical optimality results. As a remedy to mitigate the adverse effect of the estimation errors and improve the full Kelly strategy's performance, the fractional Kelly strategy was introduced. Under the fractional Kelly strategy, the portfolio weight is a fraction of that in the full Kelly strategy. Simulations and empirical evidence have shown the advantage of the fractional Kelly strategy over the full Kelly strategy (MacLean et al., 2010, 2011; Davis and Lleo, 2013). Although there is no universal choice for the weight, a straightforward and common example of the fractional Kelly strategy is the half Kelly strategy (Nekrasov, 2014; Han et al., 2019), where the portfolio weight is half of that in the full Kelly strategy. This simple strategy can reduce the portfolio risk in a bad scenario significantly (Ziemba, 2016). As a member of the fractional Kelly strategy family, the shrinkage Kelly strategy is also an alternative to adjust for parameter estimation error. One can either shrink the estimated expected stock return towards the risk-free return (Rising and Wyner, 2012) or directly shrink the portfolio weight (Han et al., 2019). Another potential modification to the Kelly strategy is applying machine learning methods in portfolio selection problems. Shen et al. (2019) improve the Kelly strategy by ensemble learning. They combine the bootstrap aggregating algorithm and random subspace method to reduce the estimation risk at a single step for multivariate portfolios. The algorithm is sequentially applied to empirical data and shown to outperform several competing strategies.

In this chapter, we apply RL to tackle the practical challenge of the Kelly strategy when faced with unknown model parameters. In RL, agents take action and receive rewards from the environment. They start with knowing very little about the environment and dynamically learn from interactions with the environment. Then they use the knowledge to maximize their rewards or objectives, e.g., expected log-return in the Kelly problem. The RL framework is more realistic than traditional portfolio selection models, where market parameters are assumed known a priori to investors. An important consideration in RL is to balance exploitation and exploration in the action process. At each decision point, the agent can either fully use the experience to execute the optimal action, i.e., exploiting the experience, or take a random action, i.e., exploring the environment. The benefit for the agent to explore is that more information about the environment is collected through the exploration to find a better path towards higher long-term rewards. Wang et al. (2019) formulate the exploration-exploitation trade-off in a control problem. In particular, they adopt an entropy-regularization method to regularize the efforts in exploration and apply it to linear-quadratic control problems.

We apply the entropy-regularization RL framework to the Kelly portfolio problem. In this problem, we assume the underlying model dynamics are known to the investor (agent) to be a geometric Brownian motion, while the model parameters are unknown. The reward is the investment return from a given trading strategy, and the investor needs to learn how to find the optimal strategy to achieve the highest expected terminal log-return. We include a general time-varying temperature parameter in the regularization term to balance the degree of exploration and exploitation in the resulting RL algorithm. We consider both the Kelly portfolio problem for controlling the amount of investment in the stock and the portion of wealth invested in the stock. The equivalence between the two formulations is not as apparent as the problem under the classical formulation. Indeed, given the same temperature parameter, they lead to different investment strategies for the two exploratory versions. We derive the optimal exploratory solution as a Gaussian distribution with parameters depending on time and portfolio wealth. By virtue of the derived closed-form solutions, we identify a relationship in the temperature parameter for the two exploratory versions to yield the same exploratory investment strategy and the same exploratory wealth process. It is worth noting that the resulting value functions are not identical even when we set the temperature parameters to yield the same exploratory investment strategy from both.

In our study, we consider three specific functional forms for the temperature parameter in the exploratory Kelly portfolio problems and develop implementable RL algorithms with the aid of the obtained closed-form solutions and value functions. The variance term in the Gaussian distribution of the optimal control under the three functional forms of temperature parameter shows different time-varying patterns: increasing, constant, and decreasing over time. We call the resulting portfolio strategies the RL Kelly strategies. We apply the three RL algorithms in extensive simulation studies. In particular, we conduct a simulation study to confirm the convergence of our RL algorithms, and we then compare their performance with the MLE-based strategies, the fully Kelly, the fractional Kelly, and the shrinkage Kelly (by Han et al. (2019)) strategies. The simulation results show that the RL Kelly strategies yield significantly better and more robust performance than these benchmark Kelly strategies even under model misspecification. Thus, the RL Kelly strategy provides a practical improvement to those existing Kelly strategies.

The entropy-regularization RL framework has been applied to several investment problems in the recent literature including Wang and Zhou (2019, 2020) and Dai et al. (2020). Wang and Zhou (2019, 2020) apply the RL method to a mean-variance problem. This method benefits the investor in the mean-variance space by achieving the target expected return faster than several other estimation methods. Dai et al. (2020) also adopt a mean-variance framework but base their analysis on the log-return of the portfolio and study the equilibrium solution, instead of the pre-commitment solution that Wang and Zhou (2019, 2020) investigate. The exploratory mean-variance problem in Dai et al. (2020) reduces to an exploratory Kelly problem as a special case, but our study in this chapter differs from Dai et al. (2020) and makes contributions in several different aspects. First, while the control in Dai et al. (2020) is the investment portion of wealth, we study the exploratory Kelly problem by considering both the amount of investment in stock and the portion of wealth in stock as the control. We derive explicit solutions for both formulations. Second, while the discussion in Dai et al. (2020) mainly focuses on a constant temperature parameter and covers the case with an exponentially decaying temperature parameter, our study for both formulations is for a general time-varying temperature parameter. Third, we clarify the condition for the two formulations to have the same exploratory investment strategy, which otherwise is not as apparent as their equivalence under the classical formulation. When the temperature parameters from the two formulations satisfy a certain equation (see equation (2.29)), both formulations will yield the same exploratory investment strategy.

The RL algorithm of Wang and Zhou (2020) addresses the minimum variance of the terminal wealth problem when the terminal mean is targeted. The RL algorithm of Dai et al. (2020) finds the equilibrium strategy under the log-mean-variance criterion. Our RL algorithms borrow the same idea of using temporal difference error to update parameters as in Wang and Zhou (2020) and Dai et al. (2020), but have a different design. First, their algorithms follow an episodic framework, where the updated values of model parameters from one episode are used as the initial values of parameters for the next episode. The investment strategy rule is updated only at the start of each episode and remains unchanged throughout each episode. In our proposed algorithm, we interpret one episode as one investment time horizon considered for the Kelly problem. Our RL algorithms update the investment strategy over each trading period to make the strategy more practical. So, our algorithm is a one-step online algorithm. Second, in our simulation studies, we treat each episode independently and start from the same initial guess of model parameters in simulating each episode. The independence of all episodes run in the simulation allows us to assess the performance of an RL algorithm applied over one episode.

The remainder of the chapter is structured as follows. Section 2.2 introduces the classical Kelly criterion problem. Section 2.3 presents the exploratory Kelly problem that takes the dollar amount invested in stock as the control. Section 2.4 introduces the exploratory Kelly problem that uses the portion of wealth as the control. Section 2.5 creates RL algorithms, and Section 2.6 contains the simulation studies. Section 2.7 concludes the section.

2.2 Kelly Criterion Problem

We consider a frictionless market and a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0\leq t\leq T}, \mathbb{P})$ in a finite time horizon $[0, T]$. The market allows short-selling and leverage without extra cost. We assume that there are only two assets in the market: one riskless asset (bond) B_t and one risky asset (stock) S_t. The risk-free interest rate is r so that $dB_t = rB_t dt$. The stock price follows a geometric Brownian motion (GBM) with constant parameters μ and σ:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where $\{W_t, 0 \leq t \leq T\}$ is a standard Brownian motion.

An investor with an initial wealth of x_0 trades in the market to maximize her discounted terminal wealth. Denote the discounted amount invested in the stock at time t by $u_t \in \mathbb{R}$ and the corresponding discounted wealth process by $x_t \equiv x_t^u \in \mathbb{R}_+$. Under the self-financing condition, the discounted wealth process x_t satisfies:

$$dx_t^u = \rho\sigma u_t dt + \sigma u_t dW_t$$

where $\rho = \frac{\mu - r}{\sigma}$ is the Sharpe ratio of the stock. We assume the investor aims to establish a trading strategy according to the Kelly criterion. In other words, the investor's optimal trading strategy is solved from the following optimization problem:

$$\max_{u\in\mathcal{A}(0,x_0)} \mathbb{E}[U(x_T^u)] = \max_{u\in\mathcal{A}(0,x_0)} \mathbb{E}[\log x_T^u], \tag{2.1}$$

where U is the logarithmic function, i.e., $U(x) = \log x$ for $x \geq 0$, and $\mathcal{A}(0, x_0)$ is the set of admissible controls (i.e., $\mathbb{R}$-valued measurable $\mathcal{F}_t$-adapted and square integrable processes).

Through either the Hamilton-Jacobi-Bellman (HJB) equation (Merton (1971)), or the martingale method (e.g., Goll and Kallsen (2000)), we have the optimal strategy:

$$u_t^*(x) = \frac{\rho x}{\sigma}. \tag{2.2}$$

and the optimal wealth process:

$$dx_t^* = \rho^2 x_t^* dt + \rho x_t^* dW_t, \text{ or equivalently } x_t^* = x_0 e^{\frac{\rho^2}{2}t + \rho W_t}.$$

Thus, the optimal expected terminal log-return is given by:

$$\mathbb{E}[\log x_T^*] = \mathbb{E}\left[\log x_0 + \frac{\rho^2}{2}T + \rho W_T\right] = \log x_0 + \frac{\rho^2}{2}T. \tag{2.3}$$

2.3 Exploratory Kelly Amount Problem

Now we extend the classical Kelly criterion problem into an RL framework. We call it the exploratory version of the Kelly criterion problem, or simply, the exploratory (Kelly) problem. In this section, we use the amount of investment in stock as the control process and refer to the resulting exploratory problem as the "(exploratory Kelly) amount problem" when it becomes necessary or helpful to distinguish the exploratory formulation from the one using the portion of wealth as the control process.

2.3.1 Motivation

The exploratory formulation in Wang et al. (2019), and Wang and Zhou (2019, 2020) is motivated by the trade-off between exploitation and exploration. In RL, when an agent is going to take an action, they will either exploit the current knowledge or explore the environment. If exploitation is selected, they will choose an action that maximizes the short-term rewards based on the experience so far. This optimal short-term action is also called the *greedy* action. However, the greedy action is desirable only when the knowledge about the environment so far is sufficient. Consequently, it is preferable to occasionally explore the environment randomly to improve the level of knowledge. The agent always faces a trade-off between exploitation and exploration when trying to accumulate the largest long-term reward. One approach to solving the RL problem is to employ an ε-*greedy* policy, i.e., choosing the greedy action with a probability of $1-\varepsilon$ and random actions with a probability of ε, for some $\varepsilon \in (0,1)$. A larger ε leads to more exploration, while a smaller ε favours employing the greedy strategy.

For our Kelly criterion problem, it seems that there is no need to explore the environment since the full Kelly strategy dominates other strategies. However, domination has been found to fail in a practical scenario. Indeed, fractional Kelly strategies perform better than the full Kelly strategy in practice. This is partially due to the large bets of the full Kelly strategy, which would be very risky in a short period. Another reason is the estimation error of model parameters, i.e., μ and σ in our setting. Estimation errors of mean returns affect portfolio selection problems more than those of the covariances (Kallberg and Ziemba (1984)). A full Kelly strategy with biased estimates would be dominated by a fractional Kelly strategy with unbiased estimates (Han et al. (2019)). The deficient practical performance of the full Kelly strategy motivates us to apply exploration methods for higher rewards.

2.3.2 Exploratory Wealth Process

In the Kelly criterion problem, the greedy action is given by (2.2). In the exploratory problem, we consider random actions which could be formulated by a control distribution $\pi(u)$, i.e., every action is randomly drawn from the control distribution. Therefore, we are now interested in finding the optimal control distribution instead of greedy actions.

In RL, a policy is how an agent behaves in different states (Sutton and Barto (2018)). In our scenario, the state corresponds to the current (discounted) wealth, and the policy corresponds to the control distribution $\pi(u)$. Given a control distribution π, every draw from it is a classical control like one in equation (2.2). Every classical control will receive a reward from the environment. Then, we could estimate the rewarding mechanism of the environment by drawing N classical controls. As N goes to infinity, we would be very close to the true rewarding mechanism.

Suppose at time t, we have a control distribution π_t and N independent sample classical controls u^i, $i = 1, 2, ..., N$, drawn from π_t. $\{x_t^i, t \in [0, T]\}$ is the wealth process under the control $\{u_t^i, t \in [0, T]\}$ for $i = 1, 2, \cdots, N$. The key idea is to view x_t^i as independent samples drawn from a new wealth process X_t^π. We denote X_t^π as the exploratory version of the controlled wealth process. This new wealth process, by the idea of RL, can be approached by sample paths x_t^i, $i = 1, 2, ..., N$ as N goes to infinity. Following the procedure in section 2.1 of Wang et al. (2019), we get the dynamics of the exploratory wealth process X_t^π

$$\begin{aligned} dX_t^\pi &= \int_{\mathbb{R}} \rho\sigma u \pi_t(u)\, du dt + \sqrt{\int_{\mathbb{R}} \sigma^2 u^2 \pi_t(u) du}\, dW_t \\ &= \rho\sigma\mu_t\, dt + \sigma\sqrt{\mu_t^2 + \sigma_t^2}\, dW_t \\ &=: \alpha(\pi_t)\, dt + \beta(\pi_t)\, dW_t \end{aligned} \tag{2.4}$$

where

$$\alpha(\pi_t) = \rho\sigma\mu_t, \qquad \beta(\pi_t) = \sigma\sqrt{\mu_t^2 + \sigma_t^2} \tag{2.5}$$

and

$$\mu_t = \int_{\mathbb{R}} u\pi(u)\, du, \qquad \sigma_t^2 = \int_{\mathbb{R}} u^2 \pi_t(u)\, du - \mu_t^2. \tag{2.6}$$

We also assume $\mathbb{E}\left[\int_0^T \sqrt{\mu_t^2 + \sigma_t^2}\, dt\right] < \infty$ to ensure that X_t^π is well defined.

If an agent invests following an RL policy $\{\pi_t\}$, then, with the highest chance, the agent would execute μ_t, i.e., the mean of the control distribution, and meanwhile, the agent would have a chance to explore the environment by taking other possible actions. It is worth noting, however, that the agent's wealth is not fully explained by the exploratory wealth process. By empirically executing an RL policy, the wealth process is a realization from the draws of the control distribution and the stock price process. Both are random and independent of each other. The exploratory wealth process only incorporates the random effect from the stock price process, or equivalently from the Brownian motion W_t. The exploratory wealth process describes the average of wealth paths from a given exploratory investment strategy π_t.

2.3.3 Trade-Off Between Exploitation and Exploration

If the control distribution gives a larger probability mass to a single control rule, e.g., classical control, the agent will explore less and execute the single control rule more frequently. An extreme case of the control distribution is that it gives probability one to a single classical control. In this case, the agent would not explore anymore and the single control would be the optimal classical control. So, the control distribution needs to be regulated to maintain a certain degree of exploration in a learning procedure.

The need of regulating the level of exploration leads to the application of differential entropy, which has been widely used in information theory to measure a random variable's average level of uncertainty or information (Cover and Thomas (1991)). More uncertainty of the control distribution corresponds to a larger entropy. The differential entropy has indeed been used by Wang et al. (2019), Wang and Zhou (2020, 2019) and Dai et al. (2020) to regularize exploration for linear-quadratic control problems with uncertainty. In particular, it has been used by Wang and Zhou (2020) and Dai et al. (2020) for a continuous-time mean-variance portfolio allocation problem.

The entropy for control π_t is defined as:

$$\mathcal{H}(\pi_t) = -\int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du. \tag{2.7}$$

Because we study the problem in the time horizon $[0, T]$, the aggregated entropy of the control distribution process $\{\pi_t, t \in [0, T]\}$ is the integral of the differential entropy over the whole investment time horizon $[0, T]$.

Our exploratory Kelly amount problem modifies the classical optimization problem (2.1) by using the exploratory wealth process and an entropy regularization term.

It is defined as follows:

$$\begin{aligned}
&\max_{\pi\in\mathcal{A}(0,x_0)} \mathbb{E}\left[\log X_T^\pi + \int_0^T \lambda_a(t)\mathcal{H}(\pi_t)\,dt\right] \\
=&\max_{\pi\in\mathcal{A}(0,x_0)} \mathbb{E}\left[\log X_T^\pi - \int_0^T \lambda_a(t)\int_{\mathbb{R}} \pi_t(u)\log\pi_t(u)\,du\,dt\right]
\end{aligned} \tag{2.8}$$

where the exogenous parameter $\lambda_a(t) > 0$ regularizes the level of exploration and is called the temperature parameter in the RL literature. The temperature parameter balances exploitation and exploration in an RL framework. A larger $\lambda_a(t)$ encourages more exploration as the entropy $\mathcal{H}(\pi_t)$ is larger for more dispersed distributional controls π_t. We attach the subscript "a" in the notation $\lambda_a(t)$ to indicate the temperature parameter is for the exploratory amount problem, as in the subsequent sections we need to distinguish it from the temperature parameter for the exploratory portion problem, which is formulated with the portion of wealth as the control.

$\mathcal{A}(0,x_0)$ in (2.8) is the set of admissible control distribution processes on $[0,T]$. For fixed $(s,x)\in[0,T]\times\mathbb{R}_+$, a control distribution process $\pi=\{\pi_t, s\le t\le T\}$ belongs to $\mathcal{A}(s,x)$ if (Wang and Zhou (2020))

(1) for each $s\le t\le T$, $\pi_t\in\mathcal{P}(\mathbb{R})$ almost surely, where $\mathcal{P}(\mathbb{R})$ denotes the set of $\mathbb{R}$-valued probability density functions;

(2) for each $\mathbb{A}\in\mathcal{B}(\mathbb{R})$, $\{\int_{\mathbb{A}}\pi_t(u)\,du,\ s\le t\le T\}$ is $\mathcal{F}_t$-progressively measurable;

(3) $\mathbb{E}\left[\int_s^T \sqrt{\mu_t^2+\sigma_t^2}\,dt\right] < \infty$, $0\le s\le T$;

(4) $\mathbb{E}\left[|\log X_T^\pi - \int_s^T \lambda_a(t)\int_{\mathbb{R}}\pi_t(u)\log\pi_t(u)\,du\,dt| \;\Big|\; X_s^\pi = x\right] < \infty$, $0\le s\le T$.

Since π_t is a probability density for all $t\in[0,T]$, it must satisfy

$$\pi_t(u)\ge 0,\ \text{for all } u\in\mathbb{R} \ \text{ and } \int_{\mathbb{R}}\pi_t(u)du = 1.$$

Below we discuss the solution to the exploratory amount problem (2.8). The value function of this optimization problem is defined as:

$$V^a(t,x;\lambda_a(t)) = \max_{\pi\in\mathcal{A}(t,x)} \mathbb{E}\left[\log X_T^\pi - \int_t^T \lambda_a(v)\int_{\mathbb{R}}\pi_v(u)\log\pi_v(u)\,du\,dv \;\middle|\; X_t^\pi = x\right].$$

A standard application of the Dynamic Programming Principle yields the following HJB equation for the value function:

$$v_t(t,x) + \max_{\pi_t \in \mathcal{P}(\mathbb{R})} \left\{ \alpha(\pi_t) v_x(t,x) + \frac{1}{2}\beta^2(\pi_t) v_{xx}(t,x) - \lambda_a(t) \int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du \right\} = 0 \tag{2.9}$$

with terminal condition $v(T,x) = \log x$. In the above, v_t, v_x and v_{xx} denote the corresponding partial derivatives of the function $v(t,x)$.

Theorem 2.3.1. *The maximization problem in equation (2.9) possesses the following density function as a solution:*

$$\pi_t^*(u; x, \lambda_a(t)) = \frac{\exp\left\{ \frac{1}{\lambda_a(t)} \left[\frac{1}{2}\sigma^2 v_{xx}(t,x) u^2 + \rho\sigma v_x(t,x) u \right] \right\}}{\int_{\mathbb{R}} \exp\left\{ \frac{1}{\lambda_a(t)} \left[\frac{1}{2}\sigma^2 v_{xx}(t,x) u^2 + \rho\sigma v_x(t,x) u \right] \right\} du}. \tag{2.10}$$

Proof. See Appendix A.1.1. □

Equation (2.10) indicates that $\pi_t^*(u; x, \lambda_a(t))$ is a Gaussian density if $v_{xx}(t,x) < 0$, which actually holds as we can tell shortly from Theorem 2.3.2.

Now that

$$\mu_t^* := \int_{\mathbb{R}} u \pi_t^*(u)\, du = -\frac{\rho v_x(t,x)}{\sigma v_{xx}(t,x)}$$

and

$$\sigma_t^* := \sqrt{\int_{\mathbb{R}} u^2 \pi_t^*(u) du - (\mu_t^*)^2} = \sqrt{-\frac{\lambda_a(t)}{\sigma^2 v_{xx}(t,x)}},$$

substituting these into the expressions for $\alpha(\pi_t)$ and $\beta(\pi_t)$ in (2.5) and using the control $\pi_t^*(u; x, \lambda_a(t))$ in equation (2.10), we can simplify the HJB equation (2.9) into:

$$v_t(t,x) - \frac{\rho^2 v_x^2(t,x)}{2 v_{xx}(t,x)} - \frac{\lambda_a(t)}{2} \log\left(-\frac{\sigma^2 v_{xx}(t,x)}{2\pi\lambda_a(t)} \right) = 0 \tag{2.11}$$

with $v(T,x) = \log x$. The above partial differential equation (PDE) is actually the Merton-type PDE in the classical optimization problem plus a term resulting from the entropy penalization. A similar PDE arises in the mean-variance portfolio allocation problem in Wang and Zhou (2020) but with a different terminal condition.

Define the real-valued function

$$f(t) = 1 + \int_t^T \lambda_a(s) ds, \quad t \in [0,T], \tag{2.12}$$

and let g_a be a real-valued function with $g_a(T) = 0$ and derivative:

$$g_a'(t) = -\frac{\rho^2}{2} f(t) + \frac{\lambda_a(t)}{2} \log \frac{\sigma^2 f(t)}{2\pi\lambda_a(t)}. \tag{2.13}$$

Theorem 2.3.2. *For the exploratory optimization problem (2.8),*

(a) the value function is

$$V^a(t, x; \lambda_a(t)) = f(t) \log x + g_a(t),$$

(b) the optimal control follows a Gaussian distribution

$$\pi_t^*(u; x, \lambda_a(t)) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{x^2 \lambda_a(t)}{\sigma^2 f(t)}\right), \tag{2.14}$$

(c) the exploratory wealth process $X_t^{\pi^*}$ under the optimal control π^* satisfies

$$dX_t^{\pi^*} = \rho^2 X_t^{\pi^*} dt + \sqrt{\rho^2 + \frac{\lambda_a(t)}{f(t)}} \cdot X_t^{\pi^*} dW_t, \tag{2.15}$$

or equivalently,

$$X_t^{\pi^*} = x_0 \exp\left\{\frac{\rho^2}{2} t + \frac{1}{2} \log \frac{1 + \int_t^T \lambda_a(s) ds}{1 + \int_0^T \lambda_a(s) ds} + \int_0^t \sqrt{\rho^2 + \frac{\lambda_a(s)}{f(s)}} dW_s\right\}, \tag{2.16}$$

(d) the expected terminal log-return is

$$\mathbb{E}[\log X_T^{\pi^*}] = \log x_0 + \frac{\rho^2}{2} T - \frac{1}{2} \log\left(1 + \int_0^T \lambda_a(s) ds\right), \tag{2.17}$$

(e) the relative loss of expected terminal log-return is

$$\frac{\mathbb{E}[\log x_T^*] - \mathbb{E}[\log X_T^{\pi^*}]}{\mathbb{E}[\log x_T^*]} = \frac{\log(1 + \int_0^T \lambda_a(s) ds)}{2 \log x_0 + \rho^2 T}, \tag{2.18}$$

where $\mathbb{E}[\log x_T^*]$ is the expected log-return of the terminal portfolio under the classical Kelly strategy and given in equation (2.3).

Proof. See Appendix A.1.2 □

The exploratory optimal control π^* for the amount problem (2.8) is centered at the classical optimal control u^* given in equation (2.2). The variance term in the optimal Gaussian distribution control depends on $\frac{\lambda_a(t)}{f(t)}$.

$$\frac{\lambda_a(t)}{f(t)} = \frac{\lambda_a(t)}{1 + \int_t^t \lambda_a(s)ds}.$$

The variance term determines the level of exploration. So, equation (2.14) sheds important light on how different time-varying properties of the control distribution can be designed by using different time-decaying $\lambda_a(t)$, or equivalently $f(t)$. However, not all time-decaying temperature processes lead to a time-decaying variance, for example, linearly decaying $\lambda_a(t)$ (see Appendix A.1.3) and exponentially decaying $\lambda_a(t)$ (see next subsection). The following proposition suggests the conditions of $f(t)$ for an appropriate temperature process.

Proposition 2.3.3. *A temperature process $\lambda_a(t)$ can be characterized by $f(t)$ as $\lambda_a(t) = -f^{'}(t)$. A necessary and sufficient condition for $\lambda_a(t)$ to be time-decaying and lead to a time-decaying variance is that $f(t)$ is strictly log-convex.*

Proof. The proof is straightforward from the definitions of $\lambda_a(t)$ and the variance of the control distribution, $\frac{x^2\lambda_a(t)}{\sigma^2 f(t)}$. □

2.3.4 Exploratory Solutions under Several Specific Temperature Parameters

In this section, we present results for three specific forms of the temperature parameter. The variance term shows different time-varying patterns over time under these three forms.

Constant λ_a(t)

When the temperature parameter for problem (2.8), $\lambda_a(t)$, is a constant $\lambda > 0$, equations (2.12) and (2.13), together with the terminal condition $g_a(T) = 0$, imply

$$f(t) = 1 + (T - t)\lambda$$

and

$$g_a(t) = -\int_t^T \left[-\frac{\rho^2}{2}[1+\lambda(T-s)] + \frac{\lambda}{2}\log\frac{\sigma^2}{2\pi\lambda} + \frac{\lambda}{2}\log[1+(T-s)\lambda] \right] ds.$$

Computing the integral for g_a and applying Theorem 2.3.2, we get the value function given by

$$\begin{aligned} V^a(t,x;\lambda) =& [1+\lambda(T-t)]\log x - \frac{1+\lambda(T-t)}{2}\log[1+\lambda(T-t)] \\ & - \frac{\lambda\rho^2}{4}(T^2-t^2) + \left[\frac{\rho^2}{2} + \frac{\lambda}{2}\left(\rho^2 T - \log\frac{\sigma^2}{2\pi e\lambda} \right) \right](T-t), \end{aligned} \tag{2.19}$$

and the optimal exploratory amount control given by

$$\pi_t^*(u;x,\lambda) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{\lambda x^2}{\sigma^2(1+\lambda(T-t))} \right). \tag{2.20}$$

Power-Decaying $\lambda_a(t)$

In practice, as the agent collects more information from the environment, their attitude towards exploration may change over time. In light of this, state-dependent or time-dependent temperature parameters have been adopted in the literature (Ishii et al. (2002); Wang and Zhou (2020); Dai et al. (2020)). For our exploratory amount problem, one feasible temperature process is the power-decaying $\lambda_a(t)$ defined as follows:

$$\lambda_a(t) = \lambda_0 \frac{(T+\lambda_1)^{\lambda_0}}{(t+\lambda_1)^{\lambda_0+1}} \tag{2.21}$$

with constants $\lambda_0 > 0$ and $\lambda_1 > 0$. Its corresponding $f(t)$ is $f(t) = (\frac{T+\lambda_1}{t+\lambda_1})^{\lambda_0}$, which is log-convex (and therefore convex). Under this particular power-decaying temperature process, the value function for the amount problem (2.8) is given by

$$V^a(t,x;\lambda_a(t)) = \begin{cases} (\frac{T+\lambda_1}{t+\lambda_1})^{\lambda_0}\log x + \frac{\rho^2(T+\lambda_1)}{2(\lambda_0-1)}\left[(\frac{T+\lambda_1}{t+\lambda_1})^{\lambda_0-1} - 1 \right] \\ \quad -\frac{1}{2\lambda_0}\left[(\frac{T+\lambda_1}{t+\lambda_1})^{\lambda_0} - 1 \right] - \frac{1}{2}(\frac{T+\lambda_1}{t+\lambda_1})^{\lambda_0}\log\frac{\sigma^2(t+\lambda_1)}{2\pi\lambda_0} \\ \quad +\frac{1}{2}\log\frac{\sigma^2(T+\lambda_1)}{2\pi\lambda_0}, & \lambda_0 \neq 1 \\ \\ \frac{T+\lambda_1}{t+\lambda_1}\log x + \frac{\rho^2(T+\lambda_1)}{2}\log\frac{T+\lambda_1}{t+\lambda_1} - \frac{T+\lambda_1}{2(t+\lambda_1)} + \frac{1}{2} \\ \quad -\frac{T+\lambda_1}{2(t+\lambda_1)}\log\frac{\sigma^2(t+\lambda_1)}{2\pi} + \frac{1}{2}\log\frac{\sigma^2(T+\lambda_1)}{2\pi}, & \lambda_0 = 1 \end{cases} \tag{2.22}$$

and the optimal control is given by

$$\pi_t^*(u;x,\lambda_a(t)) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{\lambda_0 x^2}{\sigma^2(t+\lambda_1)}\right) \tag{2.23}$$

which has a time-decreasing variance $\frac{\lambda_0 x^2}{\sigma^2(t+\lambda_1)}$.

Exponentially Decaying $\lambda_a(t)$

Consider the optimization problem (2.8) with an exponentially decaying temperature parameter

$$\lambda_a(t) = \lambda_0 e^{\lambda_0(T-t)}, \ \text{ where } \ \lambda_0 > 0.$$

Applying Theorem 2.3.2, we get the value function

$$V^a(t,x;\lambda_a(t)) = e^{\lambda_0(T-t)}\log x + \left(\frac{\rho^2}{2\lambda_0} - \frac{1}{2}\log\frac{\sigma^2}{2\pi\lambda_0}\right)\left(e^{\lambda_0(T-t)} - 1\right),$$

and the optimal amount control

$$\pi_t^*(u;x,\lambda_a(t)) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{\lambda_0 x^2}{\sigma^2}\right)$$

which has a time-constant variance $\frac{\lambda_0 x^2}{\sigma^2}$.

2.4 Exploratory Kelly Portion Problem

In portfolio selection problems, the investor can either control the amount of wealth or the proportion of wealth invested in the stock. In the classical problem, these two choices are equivalent, producing the same optimal strategy, value function, and expected terminal log-return. In Section 2.3, the exploratory problem is formulated in terms of controlling the amount invested in the stock and is regularized using the differential entropy (2.7). Now, we revisit the problem with a regularization based on the portion of wealth invested in the stock. We call the resulting control problem the "exploratory (Kelly) portion problem" to distinguish it from the formulation based on the amount of investment.

Recall that $u_t \in \mathbb{R}$ represents the amount of wealth invested in the stock. Given that total wealth is x_t, the portion of wealth invested in the stock is therefore $z_t =$

$u_t/x_t \in \mathbb{R}$. We denote the associated control distribution for the portion of wealth by $\phi_t(\cdot)$. The entropy regularizing the new control distribution is:

$$\mathcal{H}(\phi_t) = -\int_{\mathbb{R}} \phi_t(z) \log \phi_t(z)\, dz = \mathcal{H}(\pi_t) - \log x_t, \tag{2.24}$$

which includes the extra term "$\log x_t$". For a solution to the exploratory problem with the entropy applied to ϕ_t, we can apply the above relationship and solve the problem via π_t:

$$\begin{aligned} V^p(t,x;\lambda_p(t)) := & \max_{\phi\in\mathcal{A}(t,x)} \mathbb{E}\left[\log X_T^{\phi} + \int_t^T \lambda_p(v)\mathcal{H}(\phi_v)\, dv \;\middle|\; X_t^{\phi} = x\right] \\ = & \max_{\pi\in\mathcal{A}(t,x)} \mathbb{E}\left[\log X_T^{\pi} + \int_t^T \lambda_p(v)(\mathcal{H}(\pi_v) - \log X_v^{\pi})\, dv \;\middle|\; X_t^{\pi} = x\right] \\ = & \max_{\pi\in\mathcal{A}(t,x)} \mathbb{E}\Bigg[\log X_T^{\pi} - \int_t^T \lambda_p(v) \int_{\mathbb{R}} \pi_v(z) \log \pi_v(z)\, dz\, dv \\ & \qquad - \int_t^T \lambda_p(v) \log X_v^{\pi}\, dv \;\Bigg|\; X_t^{\pi} = x\Bigg]. \end{aligned} \tag{2.25}$$

Thus, V^p satisfies the HJB equation

$$v_t(t,x) + \max_{\phi_t\in\mathcal{P}(\mathbb{R})} \left\{\alpha(\phi_t)x v_x(t,x) + \frac{1}{2}\beta^2(\phi_t)x^2 v_{xx}(t,x) - \lambda_p(t)\int_{\mathbb{R}} \phi_t(z)\log\phi_t(z)\, dz\right\} = 0$$

or equivalently,

$$v_t(t,x) + \max_{\pi_t\in\mathcal{P}(\mathbb{R})} \left\{\alpha(\pi_t) v_x(t,x) + \frac{1}{2}\beta^2(\pi_t) v_{xx}(t,x) - \lambda_p(t)\int_{\mathbb{R}} \pi_t(z)\log\pi_t(z)\, dz - \lambda_p(t)\log x\right\} = 0$$

with terminal condition $v(T,x) = \log x$.

Equations (2.24) and (2.25) demonstrate why the solutions turn out not to be equivalent when using the amount of investment and using the portion of wealth as the control in the exploratory Kelly problem while they are under the classical formulation. The distribution for the portion variable yields a smaller entropy compared with that for the amount variable.

Let g_p be a real function with derivative

$$g_p'(t) = -\frac{\rho^2}{2} + \frac{\lambda_p(t)}{2}\log\frac{\sigma^2}{2\pi\lambda_p(t)},$$

and terminal condition $g_p(T) = 0$.

Theorem 2.4.1. *For the exploratory optimization problem* (2.25),

(a) the value function is

$$V^p(t, x; \lambda_p(t)) = \log x + g_p(t),$$

(b) the optimal control follows a Gaussian distribution

$$\phi_t^*(u; x, \lambda_p(t)) \sim \mathcal{N}\left(\frac{\rho}{\sigma}, \frac{\lambda_p(t)}{\sigma^2}\right), \tag{2.26}$$

(c) the exploratory wealth process $X_t^{\phi^*}$ under the optimal control ϕ^* satisfies

$$dX_t^{\phi^*} = \rho^2 X_t^{\phi^*} dt + \sqrt{\rho^2 + \lambda_p(t)} \cdot X_t^{\phi^*} dW_t, \quad \textit{with } X_0^{\phi^*} = x_0,$$

or equivalently,

$$X_t^{\phi^*} = x_0 \exp\left\{\frac{\rho^2}{2}t - \frac{1}{2}\int_0^t \lambda_p(s)ds + \int_0^t \sqrt{\rho^2 + \lambda_p(s)}dW_s\right\}, \tag{2.27}$$

(d) the expected terminal log-return is

$$\mathbb{E}[\log X_T^{\phi^*}] = \log x_0 + \frac{\rho^2}{2}T - \frac{1}{2}\int_0^T \lambda_p(s)ds,$$

(e) the relative loss of expected terminal log-return is

$$\frac{\mathbb{E}[\log x_T^*] - \mathbb{E}[\log X_T^{\phi^*}]}{\mathbb{E}[\log x_T^*]} = \frac{\int_0^T \lambda_p(s)ds}{2\log x_0 + \rho^2 T}. \tag{2.28}$$

Proof. The proof is parallel to that of Theorem 2.3.2 and hence, omitted. □

Remark 1. *A comparison between Theorems 2.3.2 and 2.4.1 leads to the following interesting observations:*

(a) The variance term in the optimal control distribution is more directly related to the temperature parameter for the portion problem than the amount problem. The variance term is proportional to the temperature parameter for the portion problem.

When $\lambda_p(t)$ is set to be identical to $\lambda_a(t)$, the variance of the investment amount is larger in the portion problem than in the amount problem (see equations (2.14) *and* (2.26)*) since $f(t) = 1 + \int_t^T \lambda_a(s)ds \geq 1$. Accordingly, the expected terminal log-return is smaller, and the relative loss is larger in the portion problem. A smaller exploratory variance for the amount solution is attributed to its relatively smaller magnitude in the entropy term. As indicated in equation* (2.25)*, its entropy is smaller than that of the corresponding amount control by $\log x_t$, and therefore, its inclusion in the optimality objective discourages exploration compared with the formulation based on the amount variable.*

(b) Equations (2.18) *and* (2.28) *indicate that, as long as the temperature parameter is set to decrease to zero over time, the relative loss in expected terminal log-return diminishes to zero when the investment time horizon becomes infinitely long.*

(c) If we set temperature parameters in the two exploratory Kelly problems to satisfy

$$\lambda_p(t) = \frac{\lambda_a(t)}{1 + \int_t^T \lambda_a(s)ds}, \quad t \in [0, t], \tag{2.29}$$

then, both problems yield the same exploratory wealth process (see equations (2.16) *and* (2.27)*). Furthermore, under the condition* (2.29)*, the optimal amount control is equivalent to the optimal portion control in the sense that both are Gaussian distributions and the parameters for one are scaled by the portfolio wealth x from the other. However, it is worth noting that the value functions differ between the two exploratory Kelly problems even if the temperature parameters satisfy condition* (2.29)*.*

For the three temperature processes $\lambda_a(t)$ given in Section 2.3 for the amount problem, the equivalent temperature process $\lambda_p(t)$ for the portion problem is respectively as follows:

1. **Constant** $\lambda_a(t)$**.** When the temperature parameter for problem (2.8), $\lambda_a(t)$, is a constant $\lambda > 0$, the temperature parameter for the equivalent portion control problem (2.25) is given by

$$\lambda_p(t) = \frac{\lambda}{1 + \lambda(T - t)}.$$

In this case, the value function for the portion problem is given by

$$V^p(t,x;\lambda_p(t)) = \log x + \frac{\rho^2}{2}(T-t) + \frac{1}{4}\left[\left(\log\frac{\sigma^2}{2\pi\lambda}\right)^2 - \left(\log\frac{\sigma^2(1+\lambda(T-t))}{2\pi\lambda}\right)^2\right]$$

and the optimal exploratory portion control is

$$\phi_t^*(u;x,\lambda_p(t)) \sim \mathcal{N}\left(\frac{\rho}{\sigma},\ \frac{\lambda}{\sigma^2(1+\lambda(T-t))}\right).$$

2. **Power-Decaying $\lambda_a(t)$.** When the temperature for the amount problem is $\lambda_a(t) = \lambda_0\frac{(T+\lambda_1)^{\lambda_0}}{(t+\lambda_1)^{\lambda_0+1}}$ with constants $\lambda_0 > 0$ and $\lambda_1 > 0$, the equivalent temperature parameter for the portion problem is given by

$$\lambda_p(t) = \frac{\lambda_0}{t+\lambda_1}.$$

In this case, the value function for the portion problem is given by

$$V^p(t,x;\lambda_p(t)) = \log x + \frac{\rho^2}{2}(T-t) + \frac{\lambda_0}{4}\left[\left(\log\frac{\sigma^2(t+\lambda_1)}{2\pi\lambda_0}\right)^2 - \left(\log\frac{\sigma^2(T+\lambda_1)}{2\pi\lambda_0}\right)^2\right]$$

and the optimal exploratory portion control is

$$\phi_t^*(u;x,\lambda_p(t)) \sim \mathcal{N}\left(\frac{\rho}{\sigma},\ \frac{\lambda_0}{\sigma^2(t+\lambda_1)}\right).$$

3. **Exponentially Decaying $\lambda_a(t)$.** When the temperature for the amount problem is $\lambda_a(t) = \lambda_0 e^{\lambda_0(T-t)}$ with constant $\lambda_0 > 0$, the equivalent temperature parameter for the portion problem is given by $\lambda_p(t) = \lambda_0$. In this case, the value function for the portion problem is given by

$$V^p(t,x;\lambda_p(t)) = \log x + \left(\frac{\rho^2}{2} - \frac{\lambda_0}{2}\log\frac{\sigma^2}{2\pi\lambda_0}\right)(T-t),$$

and the optimal exploratory portion control is

$$\phi_t^*(u;x,\lambda_p(t)) \sim \mathcal{N}\left(\frac{\rho}{\sigma},\ \frac{\lambda_0}{\sigma^2}\right).$$

2.5 Exploratory RL Algorithms

2.5.1 Policy Improvement

In a common RL problem, an agent learns from the environment through iterations between policy evaluation and policy improvement (Sutton and Barto (2018)). We previously formulated the procedure to find the optimal value function using the exploratory version of the wealth process and entropy regularization. In this subsection, we study the policy improvement to complete the essential iterations in an RL framework. We will focus on the amount problem (2.8) and consider a constant temperature parameter because the results can be obtained in parallel for the portion problem (2.25) and for both problems with a general time-varying temperature parameter.

For a policy of a particular type, the following theorem guarantees that it could be improved to a Gaussian policy. The theorem is modified from Wang and Zhou (2020) to our scenario of the Kelly criterion problem.

Theorem 2.5.1. *Suppose π_t is an admissible control policy and $V^{\pi}(t,x)$, $(t,x) \in [0,T] \times \mathbb{R}_+$ is its corresponding value function satisfying $V^{\pi}_{xx}(t,x) < 0$. Suppose a new control policy defined as*

$$\tilde{\pi}_t(u;x) \sim \mathcal{N}\left(-\frac{\rho V^{\pi}_x(t,x)}{\sigma V^{\pi}_{xx}(t,x)},\ -\frac{\lambda}{\sigma^2 V^{\pi}_{xx}(t,x)}\right) \tag{2.30}$$

is also admissible under the same choice of λ. Then we have $V^{\tilde{\pi}}(t,x) \geq V^{\pi}(t,x)$. That is, we can improve policy π_t by an admissible Gaussian policy (2.30).

Proof. See Appendix A.1.4. □

Since Theorem 2.5.1 suggests that the improved policy is Gaussian, below we illustrate how exactly we improve Gaussian policies to the form of (2.20). The improvement could be achieved by updating only the parameters of the Gaussian control, i.e., the mean and the variance. Assume we start with a simple Gaussian control:

$$\pi^0_t(u;x) \sim \mathcal{N}\left(\beta_1 x,\ \frac{cx^2}{1+b(T-t)}\right) \tag{2.31}$$

with $c > 0$ and $b > 0$ guaranteeing a positive variance. To apply the update in (2.5.1), we shall calculate the value function $V^{\pi^0}(t,x)$ and its derivatives. We start from the PDE:

$$V_t^{\pi^0}(t,x) + \int_{\mathbb{R}} \left[\rho\sigma u V_x^{\pi^0}(t,x) + \frac{1}{2}\sigma^2 u^2 V_{xx}^{\pi^0}(t,x) - \lambda \log \pi_t^0(u) \right] \pi_t^0(u)\, du = 0,$$

with $V^{\pi^0}(T,x) = \log x$.

Substituting the form of the control distribution π^0 into the above PDE yields:

$$V_t^{\pi^0} + \rho\sigma\beta_1 x V_x^{\pi^0} + \frac{1}{2}\sigma^2 \left(\beta_1^2 + \frac{c}{1+b(T-t)} \right) x^2 V_{xx}^{\pi^0} + \frac{\lambda}{2} \log \frac{2\pi e c x^2}{1+b(T-t)} = 0 \tag{2.32}$$

with the terminal condition $V^{\pi^0}(T,x) = \log x$. Following the same procedure as for solving PDE (2.11), we can obtain a solution to the above PDE:

$$\begin{aligned} V^{\pi^0}(t,x) = {} & [1+\lambda(T-t)] \log x - \left[\frac{\lambda + \sigma^2 c(1-\lambda/b)}{2b} + \frac{\lambda}{2}(T-t) \right] \log\left[1+b(T-t)\right] \\ & + \left[\left(\rho\sigma\beta_1 - \frac{1}{2}\sigma^2\beta_1^2 \right)(1+\lambda T) + \frac{\lambda}{2} \log 2\pi c e^2 - \frac{\lambda\sigma^2 c}{2b} \right](T-t) \\ & - \frac{\lambda}{2}\left(\rho\sigma\beta_1 - \frac{1}{2}\sigma^2\beta_1^2 \right)(T^2 - t^2). \end{aligned} \tag{2.33}$$

Calculating and substituting the corresponding derivatives of the above $V^{\pi^0}(t,x)$ into equation (2.30), we obtain the update:

$$\pi_t^1(u;x) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{\lambda x^2}{\sigma^2(1+\lambda(T-t))} \right) \tag{2.34}$$

which is exactly the optimal control defined in equation (2.20).

However, it is worth noting that the above two-step procedure is not a directly implementable scheme for policy improvement since it requires the true model parameter values. The value function V^{π^0} depends on the true values of the parameters σ and ρ. This motivates us to develop iterative algorithms to update our belief on the model parameters over time. The iterative algorithms will be discussed in the following section.

2.5.2 Temporal Difference Error Minimization Algorithm

The previous discussion about policy evaluation and policy improvement completes the requirements of RL procedures. In this subsection, we build algorithms for exploratory optimization problems. Our discussion will be focused on the exploratory amount problem with a constant temperature parameter λ. Algorithms for extensions to a time-varying temperature parameter and to the portion problem follow in the same fashion. The design of the algorithms is adapted from Wang and Zhou (2020). However, our algorithm is a one-step online algorithm, different from the offline algorithm used in Wang and Zhou (2020).

Theorem 2.5.1 suggests that the main task is to update model parameters since the optimal controls are from the Gaussian distribution family. We parametrize the value function V as $V^{\pi}(t,x;\boldsymbol{\alpha})$ and control $\pi_t(u)$ as $\pi_t(u;\boldsymbol{\beta})$ to facilitate the discussion of parameter updating in the algorithm, where $\boldsymbol{\alpha}=(\alpha_1,\alpha_2)$ and $\boldsymbol{\beta}=(\beta_1,\beta_2)$ with each element of the two vectors specified later. The temperature parameter λ represents the weight an agent puts on exploration against exploitation. So λ is exogenous and pre-specified by the agent.

In view of the discussion following Theorem 2.5.1 in Section 2.5.1, we start with a simple Gaussian distribution $\pi_t(u;\boldsymbol{\beta})$ parametrized by $\boldsymbol{\beta}$, with mean $\beta_1 x$ and variance

$$\frac{x^2e^{-2\beta_2-1}}{2\pi[1+\lambda(T-t)]} \tag{2.35}$$

for some constants $\beta_1<0$ and $\beta_2>0$. The parametrization for the above variance is to get a neat expression for its entropy:

$$-\int_{\mathbb{R}}\pi_t(u;\boldsymbol{\beta})\log\pi_t(u;\boldsymbol{\beta})\,du=\log x-\frac{1}{2}\log\left[1+\lambda(T-t)\right]-\beta_2. \tag{2.36}$$

Recall that the value function under the Gaussian control in (2.31) is given by (2.33). So, for the control $\pi_t(u;\boldsymbol{\beta})$, we set $c=e^{-2\beta_2-1}$ and $b=\lambda$ in (2.33) to get the value function as follows:

$$\begin{aligned}&[1+\lambda(T-t)]\log x-\frac{1+\lambda(T-t)}{2}\log\left[1+\lambda(T-t)\right]\\&+\left[\left(\rho\sigma\beta_1-\frac{1}{2}\sigma^2\beta_1^2\right)(1+\lambda T)+\frac{\lambda}{2}\log 2\pi ce^2-\frac{\sigma^2\beta_2}{2}\right](T-t)\\&-\frac{\lambda}{2}\left(\rho\sigma\beta_1-\frac{1}{2}\sigma^2\beta_1^2\right)(T^2-t^2).\end{aligned} \tag{2.37}$$

On the other hand, equation (2.19) suggests the following form for the value function:

$$V^{\pi}(t,x;\boldsymbol{\alpha})=[1+\lambda(T-t)]\log x-\frac{1+\lambda(T-t)}{2}\log\left[1+\lambda(T-t)\right]+\alpha_1(t^2-T^2)+\alpha_2(t-T), \tag{2.38}$$

for some constants α_1 and α_2.

In the iterative algorithm, we start from some initialized values for the model parameters ρ and σ. The specification of these two parameter values would give us initial values for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The initial values for $\boldsymbol{\beta}$ can be obtained by comparing the mean $\beta_1 x$ and the variance term in (2.35) with the counterparts in equation (2.20). The initial values of $\boldsymbol{\alpha}$ can be derived by comparing the above parametric form of $V^{\pi}(t,x;\boldsymbol{\alpha})$ with that of $V(t,x;\lambda)$ in equation (2.19).

Given initial values for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we now discuss how to update the parameters iteratively. We update the parameter β_1 using a heuristic relationship with the other parameters α_1, α_2 and β_2. We update α_1, α_2 and β_2 through a minimization procedure using the gradient descent algorithm that we will describe in detail later.

For the update of β_1, we note that $\beta_1 x$ is the mean in the proposed Gaussian policy and $\frac{\rho}{\sigma}x$ is the mean in the optimal Gaussian policy. So, we work to find an expression for $\frac{\rho}{\sigma}$ in terms of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. We first calculate the entropy under the optimal control distribution π_t^* in (2.20):

$$-\int_{\mathbb{R}} \pi_t^*(u;\boldsymbol{\beta}) \log \pi_{t_i}^*(u;\boldsymbol{\beta})\,du = \log x - \frac{1}{2}\log\left[1+\lambda(T-t)\right] + \frac{1}{2}\log\frac{2\pi e\lambda}{\sigma^2} \tag{2.39}$$

and then compare its expression with (2.36) to get:

$$\beta_2 = -\frac{1}{2}\log\frac{2\pi e\lambda}{\sigma^2}, \quad \text{or equivalently} \;\; \sigma^2 = 2\pi\lambda e^{2\beta_2+1}. \tag{2.40}$$

We consequently equate the coefficients for t^2 in equations (2.37) and (2.38) to obtain

$$\alpha_1 = \frac{\lambda}{2}\left(\rho\sigma\beta_1 - \frac{1}{2}\sigma^2\beta_1^2\right)$$

which, along with (2.40), gives

$$\frac{\rho}{\sigma} = \frac{2\alpha_1}{\lambda\sigma^2\beta_1} + \frac{1}{2}\beta_1 = \frac{\alpha_1 e^{-2\beta_2-1}}{\pi\lambda^2\beta_1} + \frac{1}{2}\beta_1. \tag{2.41}$$

Since $\beta_1 x$ is the mean in the proposed Gaussian policy and $\frac{\rho}{\sigma}x$ is the mean in the optimal Gaussian policy, we apply equation (2.41) to make the update:

$$\beta_1 \leftarrow \left(\frac{\alpha_1 e^{-2\beta_2-1}}{\pi\lambda^2\beta_1} + \frac{1}{2}\beta_1\right). \tag{2.42}$$

For updating the other parameters β_2, α_1 and α_2, we follow the idea of Doya (2000) and Wang and Zhou (2020) by minimizing the cumulative continuous-time temporal difference (TD) error. Suppose π_t is the optimal control and V^π is the corresponding value function. Then V^π satisfies the following equation according to Bellman's principle:

$$V^\pi(t,x) = \mathbb{E}\left[V^\pi(s, X_s^\pi) - \lambda \int_t^s \int_{\mathbb{R}} \pi_\tau(u) \log \pi_\tau(u) \, du \, d\tau \, \middle| \, X_t^\pi = x \right], \quad s \in (t, T]. \tag{2.43}$$

The continuous-time TD error measures the difference between the two sides of the equation as s approaches t (Doya (2000)):

$$\varepsilon_t = \dot{V}^\pi(t, x; \boldsymbol{\alpha}) - \lambda \int_{\mathbb{R}} \pi_t(u; \boldsymbol{\beta}) \log \pi_t(u; \boldsymbol{\beta}) \, du \tag{2.44}$$

where $\dot{V}^\pi$ is the partial derivative of V^π with respect to time t. Then, the cumulative continuous-time TD error to time t is defined as

$$C_t(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\mathbb{E}\left[\int_0^t |\varepsilon_s|^2 \, ds \right] = \frac{1}{2}\mathbb{E}\left[\int_0^t |\dot{V}^\pi(s, x; \boldsymbol{\alpha}) - \lambda \int_{\mathbb{R}} \pi_s(u; \boldsymbol{\beta}) \log \pi_s(u; \boldsymbol{\beta}) du|^2 \, ds \right]. \tag{2.45}$$

which is a function of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

To implement the RL algorithm numerically, we need to get an approximation of the TD error. We partition the time interval $[0, T]$ into $\{t_i, \ i = 0, \ldots, n\}$ with $t_0 = 0$, $t_{i+1} = t_i + \Delta t$ and $t_n = T$ for a constant Δt. Let x_i denote the state value at time t_i for $i = 0, \ldots, n$, and write $\mathcal{S}_i = \{(t_j, x_j); j = 0, \ldots, i\}$ for the information up to time t_i. We approximate the TD error up to time t_i by (with a slight abuse of notation in the subscript of C):

$$C_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \sum_{(t_j, x_j) \in \mathcal{S}_i} \left(\dot{V}^\pi(t_j, x_j; \boldsymbol{\alpha}) - \lambda \int_{\mathbb{R}} \pi_{t_j}(u; \boldsymbol{\beta}) \log \pi_{t_j}(u; \boldsymbol{\beta}) \, du \right)^2 \Delta t. \tag{2.46}$$

This method that replaces the theoretical expectation by the expectation with respect to the empirical measure called the Sample Average Approximation (SAA) method (Kim et al. (2015)).

The derivative of the value function V^π at time t_i is approximated by

$$\begin{aligned}\dot{V}^\pi(t_i, x_i; \boldsymbol{\alpha}) &= \frac{V^\pi(t_{i+1}, x_{i+1}; \boldsymbol{\alpha}) - V^\pi(t_i, x_i; \boldsymbol{\alpha})}{\Delta t} \\ &= \frac{[1+\lambda(T-t_{i+1})] \log x_{i+1} - [1+\lambda(T-t_i)] \log x_i}{\Delta t} \\ &- \frac{[1+\lambda(T-t_{i+1})] \log [1+\lambda(T-t_{i+1})] - [1+\lambda(T-t_i)] \log [1+\lambda(T-t_i)]}{2\Delta t} \\ &+ \frac{\alpha_1(t_{i+1}^2 - t_i^2) + \alpha_2 \Delta t}{\Delta t}.\end{aligned} \tag{2.47}$$

Using the parametrized control and (2.36), we take the TD error approximation as follows:

$$C_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \sum_{(t_j, x_j) \in \mathcal{S}_i} \left(\dot{V}^\pi(t_j, x_j; \boldsymbol{\alpha}) + \lambda \log x_j - \frac{\lambda}{2} \log [1+\lambda(T-t_j)] - \lambda \beta_2 \right)^2 \Delta t. \tag{2.48}$$

We then use the Gradient Descent Algorithm in Goodfellow et al. (2016) to update α_1, α_2 and β_2 at time t_i. The gradients with respect to the parameters are calculated as follows:

$$\frac{\partial C_i}{\partial \alpha_1} = \sum_{(t_j, x_j) \in \mathcal{S}_i} D(t_j, x_j, \boldsymbol{\alpha}, \boldsymbol{\beta})(t_{j+1}^2 - t_j^2) \tag{2.49}$$

$$\frac{\partial C_i}{\partial \alpha_2} = \sum_{(t_j, x_j) \in \mathcal{S}_i} D(t_j, x_j, \boldsymbol{\alpha}, \boldsymbol{\beta}) \Delta t \tag{2.50}$$

$$\frac{\partial C_i}{\partial \beta_2} = \sum_{(t_j, x_j) \in \mathcal{S}_i} D(t_j, x_j, \boldsymbol{\alpha}, \boldsymbol{\beta})(-\lambda \Delta t) \tag{2.51}$$

where

$$D(t_j, x_j, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \dot{V}^\pi(t_j, x_j; \boldsymbol{\alpha}) + \lambda \log x_j - \frac{\lambda}{2} \log [1+\lambda(T-t_j)] - \lambda \beta_2.$$

Supposing θ_α and θ_β are learning rates for updating (α_1, α_2) and β_2, we update them by $(\alpha_1, \alpha_2)' - \theta_\alpha \nabla_\alpha C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $\beta_2 - \theta_\beta \nabla_\beta C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Once we obtain an update for $(\alpha_1, \alpha_2, \beta_2)$, we update β_1 using (2.42) and keep the iterative procedure until a termination criterion is satisfied. The pseudocode for the online updating procedure is summarized in Algorithm 1.

Algorithm 1: RL Algorithm with Amount Control

```
Input: Market parameters $(\mu, \sigma, r, \rho)$, learning rates $\theta_\alpha$, $\theta_\beta$, initial wealth $x_0$,
  investment horizon $T$, discretization $\Delta t$, exploration rate $\lambda$.
Initialization: $i = 1$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
while $i \leq \frac{T}{\Delta t}$ do
    Sample $(t_i, x_i)$ under $\pi(u; \boldsymbol{\beta})$
    Update set of samples $\mathcal{S}_i = \{(t_j, x_j); j = 0, \ldots, i\}$
    Update $(\alpha_1, \alpha_2)'$ as $(\alpha_1, \alpha_2)' - \theta_\alpha \nabla_\alpha C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (2.49) and (2.50)
    Update $\beta_2$ as $\beta_2 - \theta_\beta \nabla_\beta C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (2.51)
    Update $\beta_1$ using (2.42)
    Update $\pi_t(u; x, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as $\mathcal{N}\left(\beta_1 x, \frac{e^{-2\beta_2 - 1} x^2}{2\pi(1+\lambda(T-t))}\right)$
    $i = i + 1$
end
```

Algorithms for other exploratory problems are given in Appendix A.2.

2.5.3 Discussion

This section discusses the empirical application of the RL algorithm.

First, a multivariate problem needs to be studied to implement the strategy in practice. The above theoretical results could be extended to the multivariate case and the corresponding algorithms could be built in the same style. Previous literature on RL portfolio selection problems has developed a multivariate RL mean-variance optimization problem and implemented the resulting algorithm with 20 and 50 S&P500 stocks (Wang and Zhou (2019)). The main challenge of implementation is parameter updating in a high-dimensional framework, especially for the covariance matrix. However, this challenge remains for all multivariate portfolio selection problems.

Second, how to choose the temperature process $\lambda(t)$ is also important. As $\lambda(t)$ measures the weight between the classical optimal strategy and random actions, it could be linked to the investor's risk preference. A risk-averse investor may choose a small temperature parameter. At the same time, the value of the temperature parameter determines the value of the variance in the optimal distributional strategy π_t^*. A smaller variance leads to a more leptokurtic optimal distributional strategy. In this case, random trading positions are more likely to be close to the classical optimal strategy i.e. the mean of the distributional strategy. This could be a more conservative RL strategy.

Third, the transaction costs or turnover should be considered when implementing the strategy in practice. Under the RL framework, transaction costs and turnover could largely come from independent random actions between two trading points. One way to control that is to use a smaller temperature parameter which leads to a more leptokurtic optimal distributional strategy as discussed above. Another way is to add another regularization term to the objective of the optimization problem to penalize turnover.

2.6 Simulation Studies

This section implements the RL algorithms with simulated data and compares them with several benchmark Kelly portfolio strategies.

2.6.1 Portfolio Strategies and Simulation Setting

We consider the following seven portfolio strategies:

(1) Oracle: Kelly strategy with actual parameter values,

(2) Plug-in: Kelly strategy with maximum likelihood estimation (MLE),

(3) Shrinkage: Shrinkage Kelly strategy proposed by Han et al. (2019),

(4) Fractional: Fractional Kelly strategy with MLE,

(5) RL-Amount: RL algorithm 1 with amount control and constant λ,

(6) RL-Portion: RL algorithm 3 with portion control and constant λ, and

(7) RL-Decay: RL algorithm 4 with amount control and power-decaying $\lambda_a(t)$.

The Oracle strategy is defined in equation (2.2) with the true values used for the parameters ρ and σ. The Oracle strategy is not a legitimate investment strategy because it uses true parameter values which are unknown to us in practice. This strategy is expected to perform the best since it is subject to no estimation risk and no cost of exploration, although its performance is not attainable for practical use. The Plug-in strategy follows a growing window framework to form the MLE for the parameters μ and σ (or equivalently ρ and σ), and then substitute the resulting MLE into equation (2.2) for the portfolio weight. In the Shrinkage and Fractional Kelly strategies, portfolio weights are also based on MLE but multiplied with fractional weights. The fraction in the Shrinkage Kelly strategy is defined in equation (11) in Han et al. (2019). It is proposed to mitigate the estimation error from MLE. The Shrinkage Kelly strategy is validated to outperform several fractional Kelly strategies with empirical studies (Han et al. (2019)). When there are more sample data for estimation, the fraction becomes closer to 1. For the Fractional Kelly strategy, we test nine fractional candidates from 0.1 to 0.9 with a step size of 0.1, by $M = 2{,}000$ independent simulations of stock returns. We consider model parameters $(\mu, \sigma, r, \rho) = (0.2, 0.1, 0.02, 1.8)$, investment time horizon $T = 1$ year and the initial portfolio wealth $x_0 = 1$ as the benchmark setting in our simulation study. We discretize the investment time horizon into 252 sub-intervals (i.e., the discretization length $\Delta t = 1/252$) with each subinterval representing one trading day in the stock market. For all the fractional candidates along with a fraction of one (i.e., the full Kelly strategy), we plot the corresponding average terminal log-return in Figure 2.1. For the benchmark market setting, we choose the fraction of 0.7, under which the portfolio performance is the best, in terms of the average terminal log-return. For other market settings, we repeat the same selection procedure to choose the best fraction.

Other strategies are also evaluated using 2,000 independent paths of stock returns. Over each simulated path, all the strategies (2)-(7) start with an initial estimation of parameters and updates parameters through time based on observed data. We set the initial estimation of the model parameters to be the MLE from 100 simulated data points.

For the three RL based strategies, we set the default learning rates $\theta_\alpha = \theta_\beta = 0.0005$, following Wang and Zhou (2020). For strategies RL-Amount and RL-Portion with constant temperature parameters, we set $\lambda = 0.5$ as the default. For the RL-Decay strategy with power-decaying temperature parameter, we set $\lambda_0 = 0.1$ and $\lambda_1 = 0.236$ in equation (2.21) which gives $\lambda_a(0) = 0.5$. Similar to the fraction selection for the Fractional strategy, we choose the default λ from several candidates (see Table 2.1). From 2,000 simulations, the RL-Amount strategy, with λ varying from 0.05 to 0.5, yields similar performance in terms of the average terminal log-return and its standard error. The default λ of 0.5 is not the best temperature parameter for the benchmark market setting, but a fair choice in comparing with MLE based strategies (i.e., strategies Plug-in, Shrinkage, and Fractional).

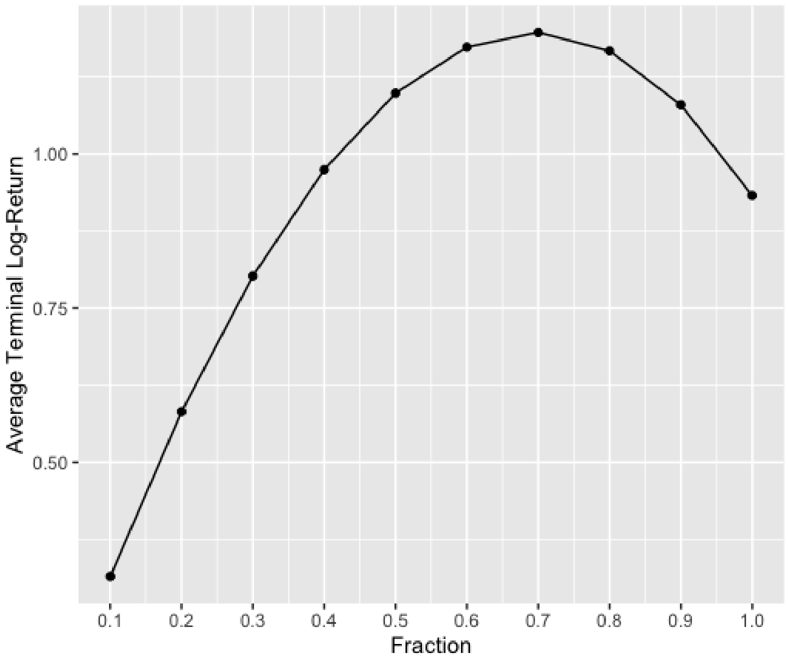

Figure 2.1: Fraction Selection for the Fractional Strategy

λ	0.025	0.05	0.1	0.25	0.5	1
Average Terminal Log-return	1.01	1.39	1.49	1.51	1.43	1.24
Standard Error	0.12	0.07	0.05	0.04	0.04	0.05

Table 2.1: Selection of Default λ: RL-Amount Strategy

2.6.2 Model Convergence

Before comparing our RL strategies with the MLE based strategies, we first investigate the convergence of the exploratory algorithm with simulated data. We focus on the RL-Decay strategy since it has the desirable time-decaying control variance. We run the Oracle strategy and RL-Decay strategy for different time horizons $T \in \{10/252, 1/12, 1/4, 1/2, 3/4, 1, 5/4, 3/2, 7/4, 2, 3, 4\}$, with other parameters set as default, and then calculate the loss of the RL-Decay strategy relative to the Oracle strategy based on 8,000 independent replications. Figure 2.2 illustrates the convergence of the relative loss to zero. The relative loss diminishes quickly as T increases. Particularly, it is close to 0 when the investment time horizon is longer than one year. It decreases to around 2% when the time horizon is two years. These simulation results mean that the relative performance of the RL strategy improves quickly over time and it performs almost as well as if we know the true parameter values in the Oracle strategy when the investment time horizon is as long as one year.

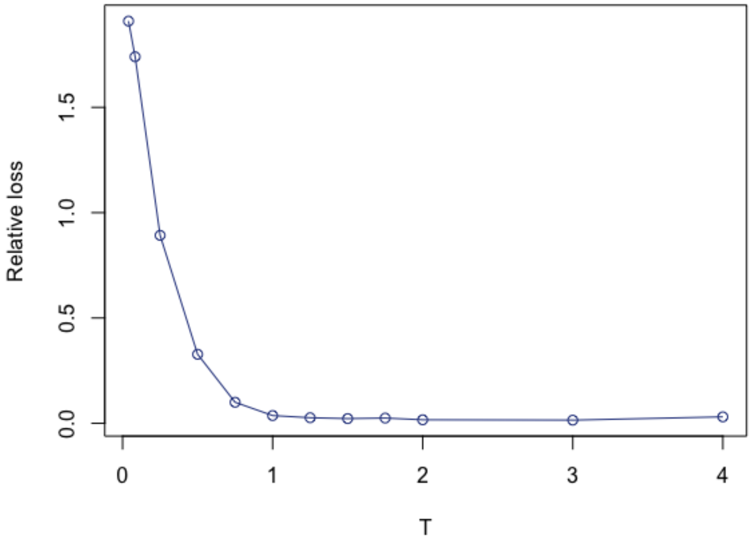

Figure 2.2: RL-Decay Model Convergence

We also study the convergence of the RL algorithm under an episodic framework. In contrast to the proposed online algorithms, an episodic algorithm only updates the model parameter values after one episode. The learned parameter values are then used throughout the next episode. For one simulation, we start from a random set of initial parameter values[1] and run for 150 episodes of length one year to get the terminal (year-end) utility at the end of each episode. Then in another independent simulation, we repeat this procedure to get another 150 terminal (year-end) utility

[1] The initial model parameter values are chosen as the MLEs from 100 simulated data points.

values. In total, we repeat for 4,000 independent simulations of 150 episodes. Hence, we have 4,000 values of the terminal (year-end) utility at the end of each episode. Their average is taken to estimate the expected terminal utility for each episode. If the RL algorithm works, then as k increases, the average terminal utility of the kth episode is anticipated to be close to the theoretically optimal value in (2.17). The result of the RL-Decay model is shown in Figure 2.3. The solid line is the average terminal utility at the end of each episode. Under the benchmark parameters, the theoretically optimal terminal utility is 1.54 (dashed line). As indicated by the graph, after six episodes, the average (year-end) terminal utility keeps fluctuating around the theoretically optimal one. This validates the rapid convergence of the algorithm under the episodic framework.

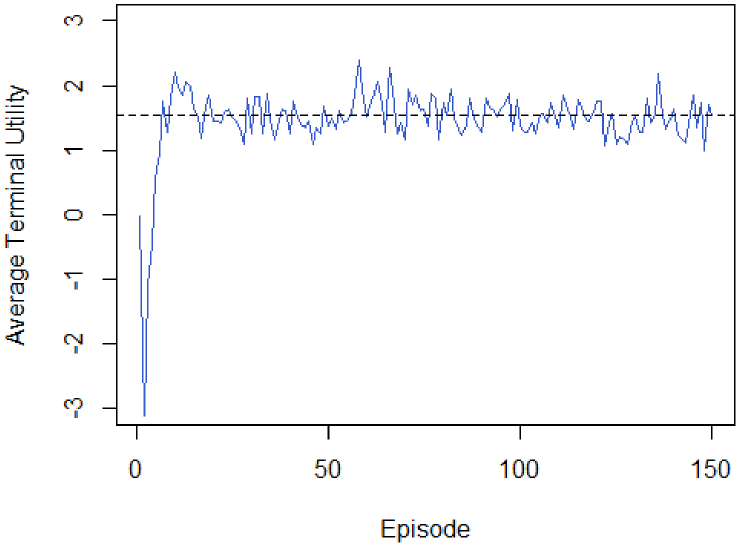

Figure 2.3: RL-Decay Model Convergence Under Episodic Framework

2.6.3 Simulation Results under the Benchmark Parameters

We now implement our RL Kelly strategies as well as some MLE based Kelly strategies with the online algorithms. We carry out these experiments because they are closer to the situation in practice where decisions are made frequently. Figure 2.4 shows the distributions of the terminal log-return under each of the strategies (2)-(7) compared to the Oracle strategy (1). Table 2.2 summarizes the average terminal log-return from each strategy. The first row is the theoretically optimal value of the expected terminal log-return in a classical Kelly criterion problem which assumes the true model parameter values are known. We also report the standard errors of the estimates. The last two columns are the theoretical and estimated values of the cost of the strategy, which is the relative difference between the average terminal log-return of a specific trading strategy and the theoretical optimal terminal log-return. Note that an inevitable issue in simulations is that the wealth may go negative, due to the uncertainty in either the stochastic stock process or the unconstrained short-selling or leverage of the RL Kelly strategy. In this case, the log-return is meaningless. We adopt the reflection approach to replace the negative wealth with its absolute value. The impact of the reflection method on our results is limited. Among 2,000 simulations, each with 252 values of wealth, only four values are reflected under the MLE based strategies. Under the RL-Amount strategy, only two values are reflected, both greater than -0.002. There are no cases for the other two RL strategies.

Model	*Mean*	*Std. Error*	*Cost*	$\widehat{Cost}$
Theoretical	1.62			
Oracle	1.68	0.04	0.00	0.04
Plug-in	0.93	0.05	0.00	0.42
Shrinkage	0.98	0.04	0.00	0.39
Fractional	1.20	0.04	0.00	0.26
RL-Amount	1.43	0.04	0.13	0.12
RL-Portion	1.40	0.04	0.15	0.14
RL-Decay	1.59	0.04	0.05	0.02

Table 2.2: Model Performance: Terminal Log-Return

The results in Figure 2.4 and Table 2.2 validate the practical merit of the fractional and shrinkage Kelly strategy over the full Kelly strategy. Furthermore, they also confirm the outperformance of the RL strategies over the three MLE based strategies. Figure 2.4 indicates that the distributions of terminal log-return from the three MLE based strategies are shifted to the left compared to those from RL strategies. The average terminal log-return reported in Table 2.2 also shows that the RL strategies yield a higher terminal log-return than the MLE based strategies on average. Moreover, the simulated results reported in Table 2.2 also confirm the

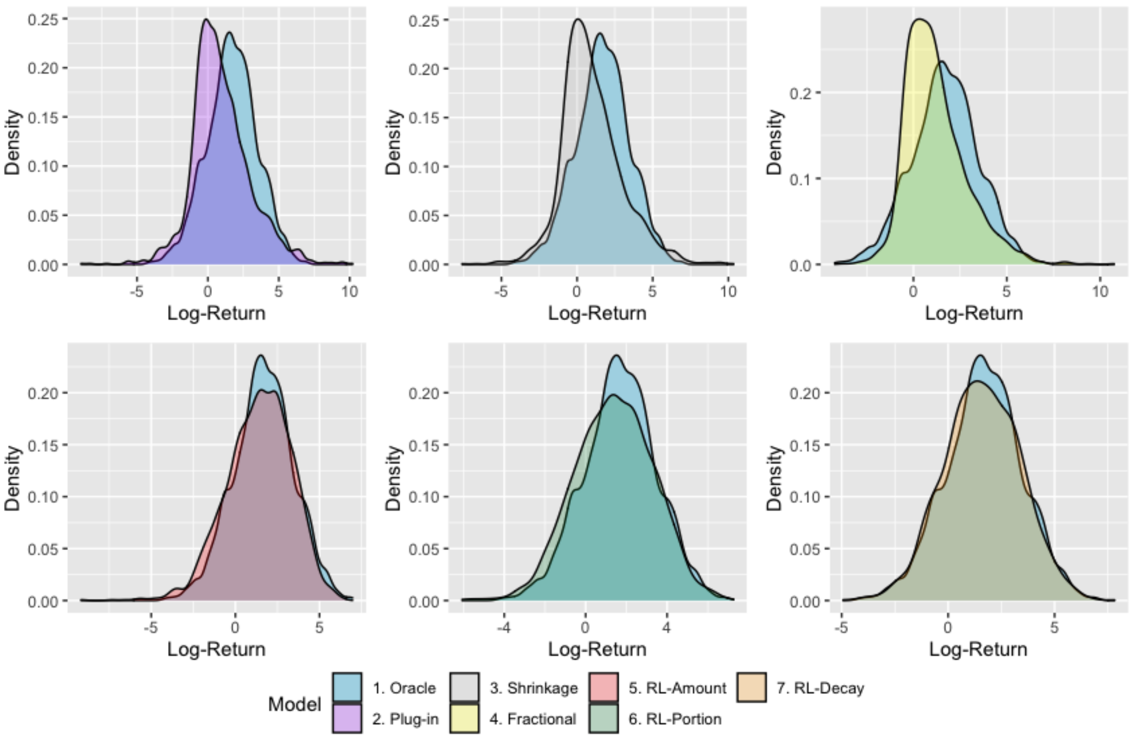

Figure 2.4: Model performance

benefit of using time-decaying λ. With a power-decaying λ, the agent is subject to less cost due to exploration and achieves a higher expected terminal log-return.

Table 2.3 reports the statistical summary of the terminal wealth from the 2,000 simulations: average value, standard deviation, skewness, and quantiles at levels of 0.1%, 1%, 5%, 95%, 99% and 99.9%. To obtain wealth, we modify the reflection approach. For example, when the wealth becomes -0.1, we build an additional account to borrow 0.2 from the bank. The total wealth is still -0.1 but the investment wealth becomes 0.1. Then we invest based on the new wealth of 0.1. At the end of the period, we report the total wealth. It is still possible to have negative terminal wealth. From the result, MLE based strategies have higher average terminal wealth, as well as significantly large standard deviations. They also have thicker tails of wealth distribution, especially the Plug-in and Shrinkage strategies. These are due to the nature of the (fractional) Kelly strategy, in particular, that it usually bets a large amount of money. Hence, in a few extreme cases where a sequence of the simulated stock returns is relatively high, the MLE based strategies benefit greatly from the aggressive investment. However, on the other hand, aggressive investment could lead to negative wealth when the stock returns are relatively low. The Fractional strategy mitigates these effects through smaller investments. As a result, the distribution is centred at a smaller mean and becomes more leptokurtic and skewed. Compared with the MLE based strategies, the three RL strategies learn the entire wealth distribution better. Their wealth distributions are closer to that under the Oracle strategy. Particularly, the RL strategy with the power-decaying temperature parameter has close quantiles to those under the Oracle model.

Model	*Mean*	*Std. Dev.*	*Skw.*	$q_{0.001}$	$q_{0.01}$	$q_{0.05}$	$q_{0.95}$	$q_{0.99}$	$q_{0.999}$
Theoretical	25.53	126.47	136.38	0.02	0.08	0.26	97.59	332.76	1316.09
Oracle	23.14	63.54	8.26	0.02	0.07	0.27	96.12	265.81	936.50
Plug-in	68.29	830.54	23.61	-2.48	0.02	0.15	98.51	694.85	12862.84
Shrinkage	70.14	897.16	25.70	-0.73	0.03	0.17	99.68	691.45	13208.59
Fractional	54.72	1088.69	39.64	0.05	0.16	0.46	72.02	415.23	4526.27
RL-Amount	17.80	41.21	6.33	0.00	0.03	0.16	74.36	212.80	403.23
RL-Portion	21.69	69.22	9.96	0.00	0.04	0.17	85.79	333.25	692.60
RL-Decay	22.24	77.68	18.51	0.02	0.07	0.28	89.90	302.69	765.29

Table 2.3: Model Performance: Terminal Wealth

The MLE based strategies and RL strategies are essentially updating market parameters over time. For each strategy, one simulation yields one pair of estimates $(\hat{\mu}, \hat{\sigma}^2)$ by the end of the investment time horizon. Since we have run 2,000 independent simulations for each strategy, we obtained 2,000 pairs of estimates from each strategy. Table 2.4 shows the mean and its standard error of those 2,000 estimates (for both μ and σ^2) from each strategy.[2] Not surprisingly, all are close to the true parameter values after taking the average over the 2,000 estimates. But the estimates of parameter μ from the RL strategies are consistently more robust than those from the MLE based strategies, noting that the MLE based strategies have a higher standard error.

[2] MLE based strategies (2), (3) and (4) share the same estimates. Hence, only one set of results is reported, named Plug-in.

Model	μ		σ^2	
	Mean	*Std. Error*	*Mean*	*Std. Error*
Oracle	0.2		0.1	
Plug-in	0.2030	0.0018	0.0998	0.0001
RL-Amount	0.2025	0.0004	0.1025	0.0002
RL-Portion	0.2056	0.0005	0.1024	0.0002
RL-Decay	0.2054	0.0003	0.1023	0.0001

Table 2.4: Model Performance: Parameter Estimation

2.6.4 Sensitivity Tests

To assess the robustness of the outperformance of the RL strategies over the three MLE based strategies, we repeat simulations for all strategies under different market settings, i.e., different values of μ and σ. We report the results in Table 2.5.

We choose four different values for σ and seven different values for μ, which yield 28 market scenarios in total, including the benchmark setting where $(\mu, \sigma) = (0.2, 0.1)$. The fractions used in the Fractional strategy are again chosen from 9 candidates, by repeating the selection procedure under the benchmark setting. The temperature parameters for the RL strategies are still the same. For each scenario, we compare the average terminal log-return between the RL strategies and the MLE based strategies. The best performance under each scenario is labelled with a superscript asterisk. Among all the 28 scenarios, the RL strategies outperform MLE based strategies under 24 settings. Particularly, the RL-Decay strategy outperforms all three MLE based strategies in all 24 cases. The other two RL strategies beat all MLE based strategies in 19 cases, even though the fractions for strategy (4) are chosen based on ex-post information.

Four exceptions where RL strategies do not outperform are under settings of $(\mu, \sigma) = (-0.1, 0.01)$, $(0.2, 0.01)$, $(0, 0.1)$ and $(0, 0.15)$ which yield extreme values of ρ^2. However, the performance of the RL-Decay strategy is still comparable with the best one in these cases. Under the setting of $(\mu, \sigma) = (0, 0.1)$ and $(0, 0.15)$, the differences between the RL-Decay strategy and the best strategy are less than 0.004. Under the other two cases, the relative differences are less than 5%.

Model	$\sigma = 0.01$						
	$\mu = -0.2$	$\mu = -0.1$	$\mu = -0.05$	$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$
Theoretical	242.00	72.00	24.50	2.00	4.50	32.00	162.00
Oracle	229.97	72.72	21.41	1.91	4.53	29.05	165.87
Plug-in	231.18	73.55*	20.86	1.12	3.61	29.12	166.63*
Shrinkage	227.59	72.38	21.03	1.18	3.68	29.04	164.50
Fractional	216.63	68.92	21.28	1.40	3.89	28.77	155.94
RL-Amount	232.07*	71.05	21.75	1.80	4.17	29.39*	156.78
RL-Portion	230.88	70.97	21.70	1.78	4.17	28.99	157.73
RL-Decay	229.70	71.17	21.89*	1.97*	4.40*	28.95	158.74
Model	$\sigma = 0.05$						
	$\mu = -0.2$	$\mu = -0.1$	$\mu = -0.05$	$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$
Theoretical	9.68	2.88	0.98	0.08	0.18	1.28	6.48
Oracle	9.02	2.76	0.92	0.06	0.20	1.34	6.43
Plug-in	7.80	1.92	0.19	-0.62	-0.48	0.61	5.34
Shrinkage	7.96	1.99	0.24	-0.58	-0.44	0.65	5.46
Fractional	8.34	2.18	0.53	0.00	0.04	0.88	5.73
RL-Amount	8.78	2.65	0.80	-0.13	0.00	1.09	5.94
RL-Portion	8.91	2.63	0.77	-0.14	-0.02	1.06	5.99
RL-Decay	9.23*	2.83*	0.96*	0.06*	0.16*	1.26*	6.27*
Model	$\sigma = 0.1$						
	$\mu = -0.2$	$\mu = -0.1$	$\mu = -0.05$	$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$
Theoretical	2.42	0.72	0.25	0.02	0.04	0.32	1.62
Oracle	2.31	0.67	0.22	0.01	0.06	0.35	1.68
Plug-in	1.51	-0.04	-0.47	-0.67	-0.63	-0.34	0.93
Shrinkage	1.56	0.00	-0.43	-0.63	-0.58	-0.30	0.98
Fractional	1.77	0.34	0.05	-0.01*	0.00	0.12	1.20
RL-Amount	2.21	0.54	0.06	-0.18	-0.14	0.15	1.43
RL-Portion	2.19	0.51	0.04	-0.28	-0.16	0.12	1.40
RL-Decay	2.38*	0.70*	0.23*	-0.01	0.03*	0.30*	1.59*
Model	$\sigma = 0.15$						
	$\mu = -0.2$	$\mu = -0.1$	$\mu = -0.05$	$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$
Theoretical	1.08	0.32	0.11	0.01	0.02	0.14	0.72
Oracle	1.02	0.29	0.09	0.00	0.03	0.16	0.77
Plug-in	0.28	-0.41	-0.60	-0.68	-0.65	-0.52	0.06
Shrinkage	0.33	-0.36	-0.55	-0.63	-0.61	-0.48	0.10
Fractional	0.61	0.09	0.01	-0.01*	0.00	0.03	0.40
RL-Amount	0.89	0.16	-0.11	-0.14	-0.16	-0.02	0.55
RL-Portion	0.87	0.12	-0.12	-0.22	-0.20	-0.06	0.51
RL-Decay	1.05*	0.30*	0.09*	-0.01	0.00*	0.12*	0.70*

Note: $*$: the best model among MLE based and RL models.

Table 2.5: Average Terminal Log-Return under Different Market Settings

To sum up, RL strategies have robust performance, in terms of the relatively high average terminal log-return, under different market scenarios. In cases where MLE based strategies fail to achieve relatively high terminal log-return, RL strategies outperform them. On the other hand, in extreme cases of ρ^2 where MLE based strategies obtain relatively high average log-return, RL strategies also have comparable performance.

2.6.5 Performance under Heston's Model

The simulation studies in the preceding subsections confirm the outperformance of our RL strategies over the MLE based strategies under the correctly specified stock price model (i.e., the geometric Brownian motion). To test the practical feasibility of the RL strategies, we consider Heston's model for the stock price:

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{L_t} S_t dW_t \\ dL_t &= \kappa(\nu - L_t)dt + \xi\sqrt{L_t} d\tilde{W}_t \end{aligned}$$

where $\tilde{W}_t$ is a Brownian motion correlated with W_t: $\text{Cov}(W_t, \tilde{W}_t) = \tilde{\rho}t$.

The experimental procedure is the same as before except that the stock price paths are simulated from Heston's model with parameters $\mu = 0.2$, $\nu = 0.01$, $\tilde{\rho} = -0.3$, $\kappa = 2$ and $\xi \in \{0.001, 0.01, 0.05, 0.1, 0.15, 0.2\}$. We exclude the Oracle strategy from the analysis because we implement investment strategies derived based on the geometric Brownian motion but test them on data from Heston's model. We use the Plug-in strategy as the benchmark and report the relative performance of the average terminal log-return for the other five strategies in Table 2.6. The average terminal log-returns are computed based on 2,000 independent replications. Zero for the relative performance measure means an equivalent performance with the Plug-in strategy and the larger the relative performance, the higher the average terminal log-return for a strategy. The issue that simulated wealth goes negative is not very significant in the results, and we reflect the negative values as we did in the previous simulation studies. For each different ξ value, at most five wealth values out of the 2,000 simulations are reflected, and they are all greater than -0.06. The issue is even less significant for a smaller ξ in Heston's model. From the results, the RL-Decay strategy still has the best performance compared to the other strategies. RL-Amount and RL-Portion strategies have better or comparable performance to the MLE based strategies in all scenarios.

Strategy	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.05$	$\xi = 0.1$	$\xi = 0.15$	$\xi = 0.2$
Shrinkage	0.06	0.07	0.08	0.09	0.13	0.22
Fractional	0.33	0.35	0.43	0.59	0.85	1.05
RL-Amount	0.56	0.57	0.63	0.69	0.76	1.01
RL-Portion	0.52	0.54	0.61	0.70	0.84	1.10
RL-Decay	0.66	0.69	0.78	0.92	1.12	1.53

Table 2.6: Performance Relative to the Plug-in Strategy: Heston's Model

2.7 Conclusion

The performance of the full Kelly strategy in practice is not as superior as claimed in theory due to estimation errors in market parameters. Two alternatives to the full Kelly strategies are fractional and shrinkage Kelly strategies. Motivated by the practical deficiency, we extend the classical Kelly criterion problem to an RL framework. Based on the novel exploratory formulation (Wang et al. (2019), Wang and Zhou (2020)), we build two exploratory Kelly criterion problems, taking the amount of investment and the portion of wealth as the control. The resulting optimal strategies, the RL Kelly strategies, are sequences of normal distributions that centre on the classical optimal allocation.

We establish learning algorithms to implement the RL Kelly strategies. We use simulated data to compare the performance of our strategies with three MLE based strategies. Our results validate the practical advantage of the fractional and shrinkage Kelly strategies against the full Kelly strategy with plug-in MLE. The RL Kelly strategies perform even significantly better than the fractional and shrinkage strategies. They achieve higher average terminal log-return. Particularly, the RL strategy with a time-decaying $\lambda_a(t)$ is the best in that it not only achieves the highest average terminal log-return but also learns the entire terminal wealth distribution more precisely. Furthermore, the performance of the RL strategies is robust under different market settings. When the MLE based strategies perform well, the RL strategies also have comparable performance. When the MLE based strategies perform poorly, the RL strategies outperform them significantly.

Chapter 3

Discrete-Time RL Mean-Variance Problem

3.1 Introduction

The continuous-time exploratory mean-variance (EMV) model proposed in Wang and Zhou (2020) provides a novel approach to solve the mean-variance optimization problem, via reinforcement learning and stochastic control. The model is implemented by discretization and achieves better performance than the MLE-based model and deep deterministic policy gradient (DDPG) model. In this chapter, we study the EMV problem, directly under the discrete-time framework. In a daily trading market setting, both EMV models achieve comparable performance. However, our discrete-time model outperforms the continuous-time model with less frequent trading e.g. weekly or monthly trading.

3.2 Discrete-Time MV Optimization

3.2.1 Problem Setup and Classical Solution

Consider a time interval with discrete time points $t = 0, \Delta t, \dots, T$, where $T = N\Delta t$, $N \in \mathbb{Z}^+$. An investor trades in a financial market at $t = 0, \Delta t, \dots, T - \Delta t$. The market consists of one stock and one risk-free bond. The rate of return of the bond is r. The rate of return of the stock follows a normal distribution $N(\mu, \sigma^2)$. We consider the discounted market by the bond. Hence, the discounted rate of return of the bond from time t to $t+\Delta t$ is 0. The discounted rate of return of the stock from time t to $t+\Delta t$ is

$$R_t \sim N((\mu - r)\Delta t, \sigma^2 \Delta t), \quad t = 0, \Delta t, \ldots, T - \Delta t.$$

The investor's discounted wealth is x_t, $t = 0, \Delta t, \ldots, T$. At each time t, u_t is the discounted wealth invested in the stock and $x_t - u_t$ is the discounted wealth invested in the bond. A trading strategy f is a series of mappings from the wealth (x) to the allocation in the stock (u):

$$f = \{f_0, f_1, \ldots, f_{T-\Delta t}\}, \quad f_t : \mathbb{R} \to \mathbb{R}, \quad u_t = f_t(x_t).$$

The investor aims to achieve the maximal mean-variance trade-off of his terminal wealth

$$\begin{aligned} P(\gamma): \qquad & \max_f \mathbb{E}[x_T] - \gamma Var(x_T) \\ & s.t. \ x_{t+1} = x_t + R_t f_t(x_t) \\ & \qquad t = 0, \Delta t, \ldots, T - \Delta t \end{aligned}$$

According to Li and Ng (2000), if a strategy f^* solves problem $P(\gamma)$, then it solves the following auxiliary problem

$$\begin{aligned} A(w): \qquad & \max_f \mathbb{E}[-(w - x_T)^2] \\ & s.t. \ x_{t+1} = x_t + R_t f_t(x_t) \\ & \qquad t = 0, \Delta t, \ldots, T - \Delta t \end{aligned} \tag{3.1}$$

with $w = \frac{1}{2\gamma} + \mathbb{E}[x_T | f^*]$. We focus on this auxiliary problem in the rest of the chapter.

In Li et al. (1998), the auxiliary problem $A(w)$ is solved analytically using dynamic programming. At time $t = 0, \Delta t, \ldots, T - \Delta t$, the optimal allocation is

$$u_t^* = \frac{\rho}{\sigma(1 + \rho^2 \Delta t)} (w - x_t) \tag{3.2}$$

where $\rho = \frac{\mu - r}{\sigma}$ is the Sharpe ratio of the stock. The efficient frontier (EF) under the classical framework is

$$Var^C(x_T) = \frac{1}{(1 + \rho^2 \Delta t)^{\frac{T}{\Delta t}} - 1} (\mathbb{E}^C[x_T] - x_0)^2. \tag{3.3}$$

The subscript "C" signifies that the mean and variance of the terminal wealth are computed under the solution from the classical setting where the model parameters are assumed known to the investor.

3.2.2 Exploratory Solution

The entropy regularization approach in Wang and Zhou (2020) incorporates the exploitation versus exploration philosophy in reinforcement learning. Similar to Chapter 2, we consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_0, \ldots, \mathcal{F}_{T-\Delta t}\}, \mathbb{P})$. Under the exploratory framework, we consider a random draw for each u_t and study their optimal sampling distributions. We denote these random variables as U_t and their densities as $f_t(u_t)$, $t = 0, \Delta t, \ldots, T-\Delta t$, which are independent of the financial market but may depend on the wealth state process x_t. $f_t : \Omega \to \mathbb{R}_+$ is $\mathcal{F}_t$-measurable and

$$\int_{\mathbb{R}} f_t(u)du = 1.$$

The entropy of the distribution is

$$-\int_{\mathbb{R}} \log\left(f_t(u)\right) f_t(u)du.$$

Modified based on the same philosophy as in Chapter 2, the auxiliary problem $A(w)$ leads to the exploratory version of problem:

$$\begin{aligned} A^E(w): \qquad & \max_f \mathbb{E}[-(w-x_T)^2] - \sum_{t=0}^{T-\Delta t} \lambda\Delta t \int_{\mathbb{R}} \log\left(f_t(u)\right) f_t(u)du \\ & s.t. \; x_{t+1} = x_t + R_t u_t \\ & \qquad u_t \sim f_t \\ & \qquad t = 0, \Delta t, \ldots, T-\Delta t \end{aligned} \tag{3.4}$$

where $u_t \sim f_t$ means that u_t is a random variable whose probability density is f_t, and the extra term is the cumulative entropy and $\lambda > 0$ is the temperature parameter. The subscript "E" indicates that it is a problem under the exploratory setting.

At time $T-\Delta t$, given $x_{T-\Delta t}$, the optimization problem is

$$\begin{aligned} & \max_{f_{T-\Delta t}} J_{T-\Delta t}(f_{T-\Delta t}|x_{T-\Delta t}) \\ := & \max_{f_{T-\Delta t}} \left\{ \mathbb{E}[-(w-x_T)^2] - \lambda\Delta t \int_{\mathbb{R}} \log\left(f_{T-\Delta t}(u)\right) f_{T-\Delta t}(u)du \right\} \end{aligned} \tag{3.5}$$

The corresponding density of the optimal sampling distribution at time $T-\Delta t$ is (see Appendix B for the derivation):

$$f^*_{T-\Delta t}(u) = \frac{1}{c_{T-\Delta t}} \exp\left\{ -\frac{\mathbb{E}[R^2_{T-\Delta t}]}{\lambda\Delta t} \left(u - \frac{(w-x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]}{\mathbb{E}[R^2_{T-\Delta t}]} \right)^2 \right\}$$

where $c_{T-\Delta t} > 0$ is a constant such that $\int_{\mathbb{R}} f_{T-\Delta t}(u)du = 1$. The distribution is a normal distribution with mean $\mu_{T-\Delta t}$ and variance $\sigma^2_{T-\Delta t}$, respectively, defined as follows:

$$\mu_{T-\Delta t} = \frac{(w - x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]}{\mathbb{E}[R^2_{T-\Delta t}]} = \frac{\rho}{\sigma(1+\rho^2\Delta t)}\left(w - x_{T-\Delta t}\right),$$

$$\sigma^2_{T-\Delta t} = \frac{\lambda \Delta t}{2\mathbb{E}[R^2_{T-\Delta t}]} = \frac{\lambda}{2\sigma^2(1+\rho^2\Delta t)}.$$

Note that $\mu_{T-\Delta t}$ is equivalent to the optimal allocation (3.2) in the classical framework.

Substituting the optimal distribution $f^*_{T-\Delta t}$ back into the expression for $J_{T-\Delta t}$ in equation (3.5) yields the optimal *cost-to-go* given $x_{T-\Delta t}$:

$$\begin{aligned}
&J^*_{T-\Delta t}(x_{T-\Delta t}) \\
=& -\mathbb{E}[R^2_{T-\Delta t}](\mu^2_{T-\Delta t} + \sigma^2_{T-\Delta t}) + 2(w - x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]\mu_{T-\Delta t} \\
&- \lambda \Delta t \left(-\frac{1}{2}\log\left(2\pi e \sigma^2_{T-\Delta t}\right)\right) - (w - x_{T-\Delta t})^2 \\
=& -\frac{\mathbb{E}[R^2_{T-\Delta t}] - (\mathbb{E}[R_{T-\Delta t}])^2}{\mathbb{E}[R^2_{T-\Delta t}]}(w - x_{T-\Delta t})^2 \\
=& -\frac{(w - x_{T-\Delta t})^2}{1+\rho^2\Delta t} + a_{T-\Delta t}
\end{aligned}$$

where $a_{T-\Delta t} = \frac{\lambda\Delta t}{2}\log\left(2\pi\sigma^2_{T-\Delta t}\right) = \frac{\lambda \Delta t}{2}\log\left(\frac{\pi\lambda}{\sigma^2(1+\rho^2\Delta t)}\right)$ is a constant.

Using dynamic programming, we can get that, at time t, the optimal allocation u_t follows a normal distribution with mean μ_t and variance σ^2_t defined as follows

$$\mu_t = \frac{\rho}{\sigma(1+\rho^2\Delta t)}\left(w - x_t\right) \qquad \text{and} \qquad \sigma^2_t = \frac{\lambda(1+\rho^2\Delta t)^{\frac{T-t}{\Delta t}-2}}{2\sigma^2}. \tag{3.6}$$

The optimal value function is given by

$$
\begin{aligned}
V(t,x) =& \max_{f}\ \mathbb{E}\left[-(w-x_T)^2 - \lambda\Delta t \sum_{i=t}^{T-\Delta t} \int_{\mathbb{R}} f_i(u)\log f_i(u)du \,\middle|\, x_t = x\right] \\
=& \max_{f}\ \mathbb{E}\left[-\frac{(w-x_{T-\Delta t})^2}{1+\rho^2\Delta t} - \sum_{i=t}^{T-2\Delta t} \lambda\Delta t \int_{\mathbb{R}} \log\left(f_i(u)\right) f_i(u)du + a_{T-\Delta t} \,\middle|\, x_t = x\right] \\
=& \max_{f}\ \mathbb{E}\left[-\frac{(w-x_{T-2\Delta t})^2}{(1+\rho^2\Delta t)^2} - \sum_{i=t}^{T-3\Delta t} \lambda\Delta t \int_{\mathbb{R}} \log\left(f_i(u)\right) f_i(u)du + \sum_{i=T-2\Delta t}^{T-\Delta t} a_i \,\middle|\, x_t = x\right] \\
& \dots \\
=& \mathbb{E}\left[-(1+\rho^2\Delta t)^{-\frac{T-t}{\Delta t}}(w-x_t)^2 + \sum_{i=t}^{T-\Delta t} a_i \,\middle|\, x_t = x\right] \\
=& -(1+\rho^2\Delta t)^{-\frac{T-t}{\Delta t}}(w-x_t)^2 + \sum_{i=t}^{T-\Delta t} \frac{\lambda\Delta t}{2}\log(2\pi\sigma_i^2) \\
=& -(1+\rho^2\Delta t)^{-\frac{T-t}{\Delta t}}(w-x)^2 + \frac{\lambda}{2}(\log\frac{\pi\lambda}{\sigma^2})(T-t) \\
& + \frac{\lambda}{2}\log(1+\rho^2\Delta t)\frac{(T-t)(T-t-3\Delta t)}{2\Delta t}
\end{aligned}
\tag{3.7}
$$

where $a_i = \frac{\lambda\Delta t}{2}\log\left(2\pi\sigma_i^2\right)$.

The exploratory EF under the exploratory framework is

$$
Var^E(x_T) = \frac{1}{(1+\rho^2\Delta t)^{\frac{T}{\Delta t}} - 1}(\mathbb{E}^E[x_T] - x_0)^2 + \frac{\lambda T}{2},
$$

where the subscript "E" signifies that the mean and variance of the terminal wealth are computed under the exploratory solution. A comparison between the above equation and (3.3) indicates that the terminal mean returns under the classical and exploratory frameworks are equivalent but the terminal variance for the exploratory version is larger by $\frac{\lambda T}{2}$, which is the cost of exploration.

3.2.3 Comparison with Continuous-time Solution

Wang and Zhou (2020) study the same problem in continuous time. However, they implement their solution with discretization. Our solution is directly derived in a discrete-time setting. Here, we compare our solution with the discretized continuous-time solution in Wang and Zhou (2020). The mean-variance optimization problem in Wang and Zhou (2020) is

$$\begin{aligned}
&\min_{f} \mathbb{E}[(x_T - w)^2] - (w - z)^2 \\
&s.t.\ \mathbb{E}[x_T] = z \\
&\qquad dx_t = \left(\int_{\mathbb{R}} \rho\sigma u f_t(u) du\right) dt + \left(\sqrt{\int_{\mathbb{R}} \sigma^2 u^2 f_t(u) du}\right) dt
\end{aligned}$$

where w is the Lagrange multiplier such that $\mathbb{E}[x_T] = z$. The admissible set of f is defined in Wang and Zhou (2020).

A comparison between our solution derived under the discrete-time setting and the one from Wang and Zhou (2020) under the continuous-time setting is summarized in Table 3.2.3. As Δt goes to zero, our discrete-time solution converges to the continuous-time solution due to the fact that $\lim_{\Delta t \to 0} (1 + \rho^2 \Delta t)^{\frac{T-t}{\Delta t}} = e^{\rho^2(T-t)}$. For $\Delta t > 0$, our optimal allocation U_t^* has a smaller variance, but a larger variance of the terminal wealth,[1] since $(1 + \rho^2\Delta t)^{\frac{T}{\Delta t}} < e^{\rho^2 T}$.

	Discrete-Time	Continuous-Time
U_t^*	$\sim N(\frac{\rho}{\sigma(1+\rho^2\Delta t)}(w - x_t), \frac{\lambda(1+\rho^2\Delta t)^{\frac{T-t}{\Delta t}-2}}{2\sigma^2})$	$\sim N(\frac{\rho}{\sigma}(w - x_t), \frac{\lambda}{2\sigma^2} e^{\rho^2(T-t)})$
w	$\frac{z(1+\rho^2\Delta t)^{\frac{T}{\Delta t}} - x_0}{(1+\rho^2\Delta t)^{\frac{T}{\Delta t}} - 1}$	$\frac{ze^{\rho^2 T} - x_0}{e^{\rho^2 T} - 1}$
EF	$Var(x_T) = \frac{1}{(1+\rho^2\Delta t)^{\frac{T}{\Delta t}} - 1}(\mathbb{E}[x_T] - x_0)^2 + \frac{\lambda T}{2}$	$Var(x_T) = \frac{1}{e^{\rho^2 T} - 1}(\mathbb{E}[x_T] - x_0)^2 + \frac{\lambda T}{2}$

Table 3.1: Comparison with the continuous-time solution in Wang and Zhou (2020)

3.3 Implementation

Based on the exploratory version of optimal allocation, we build a stochastic gradient descent algorithm and implement it with simulations in this section.

[1] This is also true under the classic setting, due to the natural difference (discretization) between the discrete-time and continuous-time settings.

3.3.1 Stochastic Gradient Descent Algorithm

We first parametrize the value function and the optimal distribution following a similar idea in Wang and Zhou (2020), based on equations (3.6) and (3.7). Recalling equation (3.7), the value function is

$$\begin{aligned}V(t,x) =& -(1+\rho^2\Delta t)^{-\frac{T-t}{\Delta t}}(w-x)^2 + \frac{\lambda}{2}(\log\frac{\pi\lambda}{\sigma^2})(T-t)\\ &+\frac{\lambda}{2}\log(1+\rho^2\Delta t)\frac{(T-t)(T-t-3\Delta t)}{2\Delta t}\end{aligned}$$

With new parameter vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$, the value function at time t can be parametrized as

$$V(t,x;\boldsymbol{\alpha}) = -\alpha_1^{-\frac{T-t}{\Delta t}}(w-x)^2 + \alpha_2(T-t)^2 + \alpha_3(T-t), \tag{3.8}$$

where

$$\begin{aligned}\alpha_1 &= 1+\rho^2\Delta t, \\ \alpha_2 &= \frac{\lambda}{4\Delta t}\log(1+\rho^2\Delta t), \\ \alpha_3 &= \frac{\lambda}{2}\log\frac{\pi\lambda}{\sigma^2} - \frac{3\lambda}{4}\log(1+\rho^2\Delta t).\end{aligned} \tag{3.9}$$

Similar to the continuous-time Bellman's equation (2.43) in Chapter 2, we have

$$V(t,x_t) = \mathbb{E}\left[V(t+\Delta t, x_{t+\Delta t}) - \lambda\Delta t\int_{\mathbb{R}} f_t(u)\log f_t(u)du \,\middle|\, x_t\right]. \tag{3.10}$$

The regularization term can be parametrized as

$$\begin{aligned}-\lambda\Delta t\int_{\mathbb{R}} f_t(u)\log f_t(u)du =& -\lambda\Delta t\left(-\frac{1}{2}\log(2\pi e\sigma_t^2)\right)\\ =&\frac{\lambda\Delta t}{2}\log\frac{\pi e\lambda(1+\rho^2\Delta t)^{\frac{T-t}{\Delta t}-2}}{\sigma^2}\\ =&\beta_1(T-t)+\beta_2\end{aligned} \tag{3.11}$$

where

$$\beta_1 = \frac{\lambda}{2}\log(1+\rho^2\Delta t)$$

$$\beta_2 \quad = \quad \frac{\lambda \Delta t}{2} \log \frac{\pi e \lambda}{\sigma^2 (1+\rho^2 \Delta t)^2}. \tag{3.12}$$

Now, we try to reduce the dimension of the problem by finding the relationships among parameters. If we start from the parametrized value function (3.8) at time $t+\Delta t$, the optimal distribution at time t becomes

$$f_t(u) \sim N\left(\frac{\rho}{\sigma(1+\rho^2\Delta t)}(w-x_t),\ \frac{\lambda \alpha_1^{\frac{T-t}{\Delta t}-1}}{2\sigma^2(1+\rho^2\Delta t)}\right). \tag{3.13}$$

This leads to the regularization term at time t as

$$-\lambda \Delta t \int_{\mathbb{R}} f_t(u) \log f_t(u) du = \frac{\lambda \log \alpha_1}{2}(T-t) + \frac{\lambda \Delta t}{2} \log \frac{\pi e \lambda}{\sigma^2(1+\rho^2\Delta t)\alpha_1}. \tag{3.14}$$

This step is called *policy improvement* in reinforcement learning. Comparing the above equation to equation (3.11), we have the relationships:

$$\frac{\lambda}{2} \log \alpha_1 = \beta_1 \tag{3.15}$$

and

$$\sigma^2(1+\rho^2\Delta t) = \frac{\pi \lambda}{\alpha_1} e^{1-\frac{2\beta_2}{\lambda \Delta t}}. \tag{3.16}$$

On the other hand, if we start with the parametrized form of the regularization term (3.14), solving equation (3.10) gives us the value function at time t

$$\begin{aligned} V(t,x) =& -\alpha_1^{-\frac{T-t}{\Delta t}}(w-x^2) + \alpha_2(T-t)^2 + (-2\alpha_2\Delta t + \alpha_3 + \beta_1)(T-t) \\ & - \frac{\sigma^2 \Delta t(1+\rho^2\Delta t)\alpha_1}{2\pi} e^{\frac{2\beta_2}{\lambda \Delta t}-1} + \alpha_2 \Delta t^2 - \alpha_3 \Delta t + \beta_2 \\ =& -\alpha_1^{-\frac{T-t}{\Delta t}}(w-x^2) + \alpha_2(T-t)^2 + (-2\alpha_2\Delta t + \alpha_3 + \beta_1)(T-t) \\ & - \frac{\Delta t \lambda}{2} + \alpha_2 \Delta t^2 - \alpha_3 \Delta t + \beta_2. \end{aligned}$$

The last equality is from the relationship (3.16) between α_1 and β_2.

This step is called *policy evaluation* in reinforcement learning. From this, comparing the above equation with the parametrized value function (3.8), we have the relationship

$$0 \quad = \quad -\frac{\lambda \Delta t}{2} + \alpha_2 \Delta t^2 - \alpha_3 \Delta t + \beta_2$$

$$\Longrightarrow \alpha_3 \quad = \quad \alpha_2 \Delta t - \frac{\beta_2}{\Delta t} + \frac{\lambda}{2}. \tag{3.17}$$

Therefore, we are able to simplify the parametrized value function and obtain a parametrized optimal distribution. Plugging the above equation into the parametrized value function (3.8), the value function at time t can be further simplified as

$$V(t, x; \boldsymbol{\alpha}, \beta_2) = -\alpha_1^{-\frac{T-t}{\Delta t}}(w-x)^2 + \alpha_2(T-t)^2 + \left(\alpha_2 \Delta t - \frac{\beta_2}{\Delta t} + \frac{\lambda}{2}\right)(T-t) \tag{3.18}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ is redefined.

Plugging equations (3.9), (3.12) and (3.16) into equation (3.13), the optimal distribution at time t can be parametrized as, if we assume the Sharpe ratio ρ is positive,

$$f_t(u) \sim N\left(\sqrt{\frac{\alpha_1 - 1}{\pi \lambda \Delta t}} e^{\frac{\beta_2}{\lambda \Delta t} - \frac{1}{2}}(w-x),\ \frac{1}{2\pi}\alpha_1^{\frac{T-t}{\Delta t}} e^{\frac{2\beta_2}{\lambda \Delta t} - 1}\right). \tag{3.19}$$

Based on the parametrized forms (3.14) and (3.18), we build the reinforcement learning algorithm similar to that in Chapter 2. The algorithm updates parameters $(\boldsymbol{\alpha}, \beta_2)$ to minimize the cumulative temporal difference (TD) error, defined by

$$\begin{aligned}
C(\boldsymbol{\alpha}, \beta_2) =& \frac{1}{2}\sum_{t=0}^{T-\Delta t}\left(V(t+\Delta t, x_{t+\Delta t}) - V(t, x_t) - \lambda \Delta t \int_{\mathbb{R}} f_t(u) \log f_t(u) du\right)^2 \\
=& \frac{1}{2}\sum_{t=0}^{T-\Delta t}\Bigg(- \alpha_1^{-\frac{T-t-\Delta t}{\Delta t}}(w - x_{t+\Delta t})^2 + \alpha_2(T-t-\Delta t)^2 + \alpha_3(T-t-\Delta t) \\
& - (-\alpha_1^{-\frac{T-t}{\Delta t}}(w-x_t)^2 + \alpha_2(T-t)^2 + \alpha_3(T-t)) + \beta_1(T-t) + \beta_2\Bigg)^2 \\
=& \frac{1}{2}\sum_{t=0}^{T-\Delta t}\Bigg(- \alpha_1^{-\frac{T-t-\Delta t}{\Delta t}}(w - x_{t+\Delta t})^2 + \alpha_1^{-\frac{T-t}{\Delta t}}(w-x_t)^2 \\
& - \alpha_2(2T - 2t - \Delta t)\Delta t - \alpha_3 \Delta t + \beta_1(T-t) + \beta_2\Bigg)^2 \\
=& \frac{1}{2}\sum_{t=0}^{T-\Delta t}\Bigg(- \alpha_1^{-\frac{T-t-\Delta t}{\Delta t}}(w - x_{t+\Delta t})^2 + \alpha_1^{-\frac{T-t}{\Delta t}}(w-x_t)^2 \\
& - \alpha_2(2T - 2t - \Delta t)\Delta t - \alpha_2 \Delta t^2 + \beta_2 - \frac{\lambda \Delta t}{2} - \frac{\lambda \log \alpha_1}{2}(T-t) + \beta_2\Bigg)^2
\end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{2}\sum_{t=0}^{T-\Delta t}\Big(-\alpha_1^{-\frac{T-t-\Delta t}{\Delta t}}(w-x_{t+\Delta t})^2+\alpha_1^{-\frac{T-t}{\Delta t}}(w-x_t)^2 \\
&-\frac{\lambda\log\alpha_1}{2}(T-t)-\alpha_2(2T-2t)\Delta t+2\beta_2-\frac{\lambda\Delta t}{2}\Big)^2 \\
:=&\frac{1}{2}\sum_{t=0}^{T-\Delta t}C_t^2.
\end{aligned}
$$

This TD error is derived from equation (3.10) and is similar to that in Chapter 2. See equations (2.45) to (2.48).

The stochastic gradients of $C(\boldsymbol{\alpha},\beta_2)$ are

$$
\left(\sum_{t=0}^{T-\Delta t}C_t\frac{\partial C_t}{\partial\alpha_1},\ \sum_{t=0}^{T-\Delta t}C_t\frac{\partial C_t}{\partial\alpha_2},\ \sum_{t=0}^{T-\Delta t}C_t\frac{\partial C_t}{\partial\beta_2}\right)
$$

where

$$
\begin{aligned}
\frac{\partial C_t}{\partial\alpha_1} &= \frac{T-t-\Delta t}{\Delta t}\alpha_1^{-\frac{T-t}{\Delta t}}(w-x_{t+\Delta t})^2 \\
&\quad -\frac{T-t}{\Delta t}\alpha_1^{-\frac{T-t}{\Delta t}-1}(w-x_t)^2-\frac{\lambda}{2\alpha_1}(T-t) \qquad (3.20)\\
\frac{\partial C_t}{\partial\alpha_2} &= -2(T-t)\Delta t \qquad (3.21)\\
\frac{\partial C_t}{\partial\beta_2} &= 2. \qquad (3.22)
\end{aligned}
$$

In the algorithm, α_1, α_2 and β_2 are updated using the gradient descent approach with the learning rate θ. The Lagrange multiplier w is updated based on the constraint $\mathbb{E}[x_T]=z$ with the learning rate θ_w (Wang and Zhou (2020)).

$$
w_{n+1}=w_n-\theta_w\left(\frac{1}{k}\sum_{i=nk+1}^{nk+k}(x_T)_i-z\right),\qquad n=0,1,\dots,\frac{M}{k}-1, \qquad (3.23)
$$

where M is the number of total simulations, k is the number of simulations to update the Lagrange multiplier and θ_w is the learning rate. $(x_T)_i$ is the terminal wealth in the $i-th$ simulation. When the cumulative TD error $C(\boldsymbol{\alpha},\beta_2)$ is less than a particular threshold e.g. 1×10^{-6}, we stop the update and evaluate the rest of the simulations based on the fixed parameters.

Similar to the continuous-time RL Kelly problem in Chapter 2, we also build an RL algorithm to implement our theoretical results. Here the algorithm is offline. That is, the updated parameters from one simulation will be used as the initial values for the next simulation as adopted by Wang and Zhou (2020).

Algorithm 2: RL Algorithm for Discrete-time EMV Problem

```
Input: Market parameters $(\mu, \sigma, r, \rho)$, learning rates $\theta$, $\theta_w$, initial wealth $x_0$,
  investment horizon $T$, discretization $\Delta t$, exploration rate $\lambda$, target mean
  return $z$, number of total simulations $M$, number of simulations to update
  the Lagrange multiplier $k$.
Initialization: $n = 0$, $w_n = 1$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
while $nk \leq M$ do
    Initialization: $i = 1$
    while $i \leq \frac{T}{\Delta t}$ do
        Sample $(t_i, x_i)$ under $f_t(u)$
        Update set of samples $\mathcal{S}_i = \{(t_j, x_j); j = 0, \ldots, i\}$
        Update $(\alpha_1, \alpha_2)'$ as $(\alpha_1, \alpha_2)' - \theta\nabla_\alpha C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (3.20) and (3.21)
        Update $\beta_2$ as $\beta_2 - \theta\nabla_\beta C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (3.22)
        Update $\alpha_3$ using (3.17) and $\beta_1$ using (3.15)
        Update $f_t(u)$ using (3.19)
        $i = i + 1$
    end
    if $n \mod k = 0$ then
        Update $w_n$ using (3.23)
        $n = n + 1$
end
```

3.3.2 Simulation Results

We compare the numerical performance of our algorithm and that in Wang and Zhou (2020). We use the same parameter values as in Wang and Zhou (2020):

$$T = 1,\ \Delta t = \frac{1}{252},\ x_0 = 1,\ z = 1.4, r = 0.02$$
$$\lambda = 2,\ M = 10{,}000,\ k = 10, \theta = 0.0005,\ \theta_w = 0.05$$

$\Delta t = \frac{1}{252}$ means that the trading is on a daily base.

In Figure 3.1, we report the performance under two scenarios, compared to the continuous-time algorithm. Each point is the average terminal wealth over 50 episodes. In total, we have $M = 20{,}000$ episodes. So there are 400 sample means in each line. The black line is the target mean level $z = 1.4$, which is exogenous. From the comparisons, with our discrete-time algorithm (red line), the average wealth converges much faster to the target level (only after 5,000 episodes).

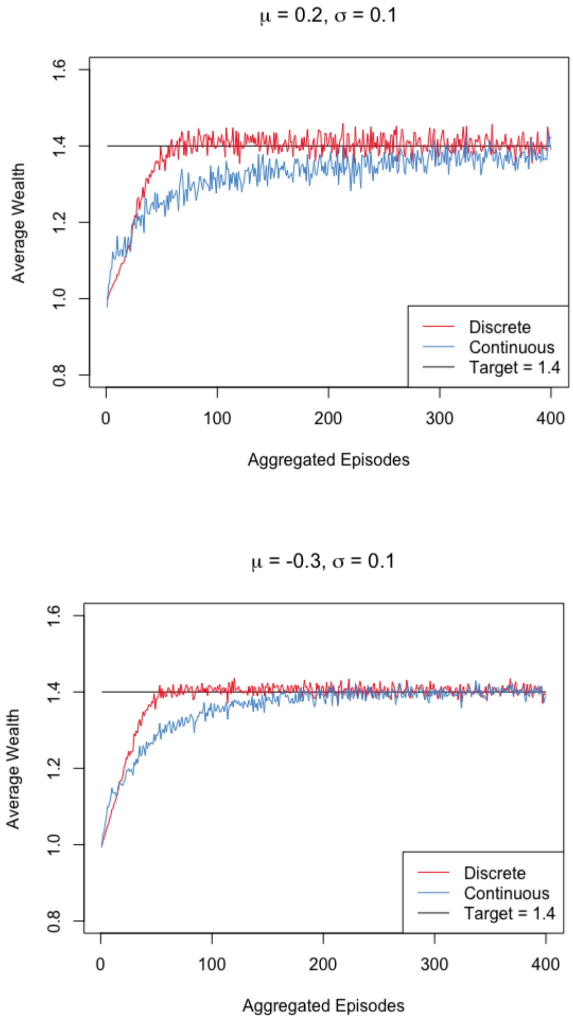

Figure 3.1: Performance comparisons with the continuous-time EMV model

Table 3.2 summarizes the average return, standard deviation and Sharpe ratio of the terminal 2,000 episodes (same as Wang and Zhou (2020)). The results of the continuous-time EMV model (last three columns) are copied from Table 1 in Wang and Zhou (2020).

μ	σ	Discrete			Continuous		
		Avg. Return	SD	Sharpe Ratio	Avg. Return	SD	Sharpe Ratio
-0.5	0.1	0.400	0.074	5.122	0.396	0.078	5.107
-0.3	0.1	0.400	0.120	3.159	0.390	0.127	3.039
-0.1	0.1	0.385	0.312	1.168	0.330	0.272	1.218
0	0.1	0.173	0.825	0.185	0.204	1.130	0.180
0.1	0.1	0.351	0.439	0.753	0.318	0.414	0.769
0.3	0.1	0.400	0.140	2.709	0.385	0.138	2.785
0.5	0.1	0.400	0.082	4.632	0.394	0.084	4.772
-0.5	0.2	0.399	0.147	2.581	0.387	0.148	2.606
-0.3	0.2	0.399	0.239	1.586	0.359	0.226	1.598
-0.1	0.2	0.386	0.622	0.589	0.309	0.495	0.625
0	0.2	0.180	1.648	0.097	0.105	0.853	0.123
0.1	0.2	0.349	0.883	0.373	0.221	0.560	0.395
0.3	0.2	0.400	0.281	1.351	0.345	0.249	1.387
0.5	0.2	0.400	0.164	2.322	0.385	0.164	2.350
-0.5	0.3	0.399	0.220	1.725	0.353	0.210	1.682
-0.3	0.3	0.399	0.357	1.061	0.323	0.326	0.992
-0.1	0.3	0.387	0.929	0.396	0.317	0.834	0.380
0	0.3	0.187	2.462	0.068	0.079	0.852	0.092
0.1	0.3	0.347	1.332	0.246	0.282	0.941	0.300
0.3	0.3	0.400	0.423	0.898	0.334	0.362	0.921
0.5	0.3	0.401	0.246	1.546	0.350	0.221	1.583
-0.5	0.4	0.398	0.292	1.296	0.342	0.247	1.385
-0.3	0.4	0.399	0.475	0.797	0.320	0.382	0.839
-0.1	0.4	0.389	1.234	0.299	0.241	0.841	0.287
0	0.4	0.193	3.271	0.053	0.057	0.820	0.070
0.1	0.4	0.345	1.787	0.182	0.155	0.769	0.202
0.3	0.4	0.400	0.566	0.671	0.320	0.445	0.716
0.5	0.4	0.401	0.329	1.157	0.329	0.279	1.174

Table 3.2: Performance comparison between the discrete-time and continuous-time EMV models

Our algorithm yields generally higher average returns and standard deviations than the continuous-time algorithm. These lead to comparable Sharpe ratios. Higher average returns are consistent with the convergence result. From the above two figures, our algorithm converges to the target level faster while the continuous-time algorithm converges slower. The latter will hence have smaller terminal returns. Higher standard deviations are consistent with the theoretical results in Table 3.2.3. Under the discrete-time setting, the variance of terminal wealth is larger than that under the continuous-time setting.

In Table 3.3.2, we also test the robustness of the two algorithms by using different trading frequencies Δt. We use $\mu = 0.2$ and $\sigma = 0.1$. Our algorithm has a rather stable performance (terminal Sharpe ratio) with different trading frequencies. However, the continuous-time algorithm is very sensitive to the trading frequency, leading to a discretization error. This result particularly shows the advantage of our discrete-time algorithm.

Algorithm	Daily	Weekly	Monthly	Seasonally
Discrete-time	1.819	1.819	1.800	1.785
Continuous-time	1.654	1.587	1.431	1.109

Table 3.3: Performance (terminal Sharpe ratio) comparison between two algorithms with different trading frequencies

3.4 Conclusion

From our analysis, we find that there exists a discretization error in the implementation of the algorithm of Wang and Zhou (2020). By studying the problem directly under the discrete-time framework, we derive the true optimal strategy that can be implemented in a discretized structure. Our results converge to those in the continuous-time setting as the discretization goes to zero. Through several numerical examples, we find that the strategy performance based on the discrete-time solution is generally better than that based on the continuous-time solution, in terms of a faster convergence of the mean return, an improved Sharpe ratio, as well as robustness across different trading frequencies.

Chapter 4

Valuation of Large Variable Annuity Portfolios

4.1 Introduction

An important investment-related task for insurance companies is to manage the assets and liabilities of their products. To do so, companies need to know the present values of products. In this chapter, we focus on the valuation problem of investment-linked insurance products, particularly, variable annuities.

A variable annuity (VA) is a deferred annuity contract that allows an annuitant to invest in the financial market via mutual funds and provides downside protection in the form of a minimum guarantee. There are different types of guarantee riders: guaranteed minimum death benefit (GMDB), guaranteed minimum accumulation benefit (GMAB), guaranteed minimum withdrawal benefit (GMWB), and guaranteed minimum income benefit (GMIB) (Hardy, 2003). In a VA contract, the insurance company is obligated to pay the annuitant the guaranteed amount upon events (death, withdrawal, or survival). VAs have rapidly grown in popularity, and it has become vital for life insurers to develop and maintain effective hedging programs for their variable annuity portfolios. An effective hedging program routinely requires frequent valuation of a very large number of individual VA contracts.

There is a large literature studying the valuation of VA contracts. Early research adopted option pricing techniques, e.g., Brennan and Schwartz (1976); Boyle and Schwartz (1977), and Bacinello and Ortu (1993). Boyle and Hardy (1997) compared stochastic simulation and option pricing methods for their performance in pricing and reserving for VA contracts. Hardy (2000) compared three methods for reserving for VA contracts with maturity guarantees: the simulation and value-at-risk (VaR) method, dynamic hedging, and static hedging. Bauer et al. (2008) provided a comprehensive simulation framework for pricing different types of guarantees. Bacinello et al. (2011) proposed a unifying framework of valuation using ordinary and least squares Monte Carlo (LSMC) simulation techniques. Huang and Kwok (2016) applied the LSMC method to price and hedge VA with GLWB riders. Shen and Weng (2019, 2020) developed a backward simulation method that, employing shape-preserving sieve estimation, provides a computationally efficient LSMC simulation algorithm for the valuation of variable annuities with complex withdrawal options. Yang and Dai (2013) introduced a tree model for valuing withdrawal guarantees. Doyle and Groendyke (2018) found that the efficiency in pricing and hedging VA guarantees is improved by using neural networks. Huang et al. (2022) used a change of numéraire approach in valuing an accumulating guarantee. See Gan (2013) for a review of methods for pricing individual VA contracts. See Gan and Valdez (2019) for a review of methods for pricing VA portfolios using the two-step framework.

Despite the extensive research on the valuation of individual VA contracts, valuing VA portfolios remains a challenging problem because they typically contain a large number of VA contracts that have different attributes such as gender, account value, maturity and guarantee type. The complexity of the guaranteed payoff limits the availability of analytical formulae for valuation. A popular non-parametric method is nested simulation (Reynolds and Man, 2008; Bauer et al., 2012), which uses Monte Carlo simulation procedures in both outer and inner loops to determine prices and sensitivities of interest. However, due to the complex structure of VA portfolios, a huge number of simulations is required to ensure accurate valuation. This raises another issue in practice, i.e., computational cost. As a VA portfolio may contain more than 100,000 individual contracts (Gan and Lin, 2015), valuing the entire portfolio using simulation could be extremely time-consuming, not to mention that a large insurance company usually maintains several VA portfolios.

A practically viable solution is the metamodelling framework where a two-step procedure is applied. The first step is to select a reasonable set of representative contracts to be evaluated and the second step obtains valuation of the remaining contracts through a certain interpolation procedure. The majority of the literature on the valuation of large VA portfolios adopts this two-step framework by proposing different methods for selecting representative contracts and/or interpolation. Gan (2013) proposed a framework using clustering analysis for the selection of representative contracts and the ordinary kriging method to estimate the values and Greeks of the remaining contracts. Gan and Lin (2015) modified the framework by incorporating nested simulation to evaluate the representative contracts and the universal kriging method to estimate values for the remaining contracts. Hejazi and Jackson (2016) adopted a spatial interpolation framework to estimate Greeks of VA portfolios for enhanced efficiency. Hejazi et al. (2017) then improved the method by using neural networks to search for an effective distance function in spatial interpolation. Gan and Valdez (2016, 2018) used GB2 (generalized beta of the second kind) regression models for the interpolation in the second step of the framework. Gan and Valdez (2016) compared five methods in the first step of selecting representatives: the random sampling method, the low-discrepancy sequence method, the data clustering method, the Latin hypercube sampling method, and the conditional Latin hypercube sampling method. Gan and Lin (2017) proposed a two-level metamodelling approach for efficiently calculating the Greeks in VA portfolio hedging. Xu et al. (2018) used moment matching Monte Carlo methods to calculate quantities (dollar Deltas, VaRs and CVaRs) for the representative contracts and then machine learning methods such as regression trees and neural networks to estimate the quantities for the portfolio. Gan and Valdez (2020) compared the hierarchical k-means algorithm to the truncated fuzzy c-means (TFCM) algorithm in the selection and found the hierarchical k-means algorithm is more efficient. Feng et al. (2020) proposed a simple random sampling and clustering method to select representative contracts. Integrating with two metamodels, they found that the proposed method improves the estimation accuracy compared to the state-of-the-art method by Gan (2013). Liu and Tan (2021) introduced a green mesh method for valuation and demonstrated its efficiency in a real-time valuation application.

More recently, Lin and Yang (2020b) developed a fast and efficient nested simulation procedure for the valuation of large VA portfolios using a surrogate modelling approach. They proposed using a spline regression model to reduce the number of outer loops and a model-assisted finite population estimation framework to reduce the number of policies in use for the nested simulation. Their method has the merit of being theoretically justifiable under certain modelling assumptions. Furthermore, Lin and Yang (2020a) extended the surrogate modelling approach into a multi-period setting for dynamic hedging problems.

In this chapter, we focus on the first step of the metamodelling framework and propose a two-phase selection method for the determination of representative contracts. In the second step, we use the universal kriging method as in Gan and Lin (2015); Gan and Valdez (2019, 2020). The general idea of our method is to split the selection of representative contracts into two phases: an initial selection phase, and a conditional selection phase. In the initial selection phase, we select a subset of representatives based on the known attributes of all VA contracts and obtain their values by a valuation method (for example nested simulation). While any conventional selection method could in principle be employed in the first phase, we focus on the k-prototypes and the hierarchical k-means clustering methods, which are unsupervised learning methods. In the conditional selection phase, we make use of the obtained contract values of these representatives from the initial selection phase in addition to the attribute data of the VA contracts. Specifically, we model the contract values as Gaussian random variables. This is also the assumption underlying the universal kriging method. From this assumption, we have a prior distribution for the contract values. Based on the information from the representatives selected in the first phase, we can derive a posterior distribution for the contract values. Then, we use a statistical distance to measure the similarity between two contracts (as Gaussian random variables) and propose a prudently designed conditional k-means algorithm to select the rest of the representative contracts. The conditional k-means algorithm takes those representative contracts from the initial selection phase as fixed centroids and searches for the remaining representative contracts as the other centroids in the k-means clustering procedure to minimize an overall loss. It is called so because the search for the other centroids is conditional on these fixed ones from the initial selection phase, and this conditional search guarantees that there would not be overlapping representations between the set of contracts from the first phase and those from the second phase. Furthermore, we establish the local convergence of the conditional k-means algorithm.

There are several reasons for us to use the posterior distribution for the determination of representative contracts in the second phase. First, when we model the contract values as Gaussian random variables under the kriging regression model, if the distributions of the two contract values are different from each other, then their corresponding attributes must differ significantly. Hence, selecting based on the posterior distribution of the contract values also captures the heterogeneity in the VA contract attributes, which is usually the target to achieve in various representative selection methods in the literature. Second, selecting based on the posterior distribution yields more similarity in the distribution of contract values between the representative set and the whole VA portfolio. Based on the universal kriging method, contracts with different attributes could also have similar contract values. Hence, the conventional selection method tends to yield a more concentrated distribution of contract values in the representative set. The numerical study results in Table 4.4.3 confirm that, compared with the conventional method, the quantiles of the contract values of representatives from our method are closer to those of the whole VA portfolio.

As described above, our two-phase selection method is distinguished from conventional methods in the fact that the selection of representative contracts in our method uses information from both the contract attributes and the contract values. In contrast, most methods previously proposed in the literature (for example, the random sampling method, the clustering methods, the (conditional) Latin hypercube sampling methods, and the population sampling methods (e.g., Gan and Lin (2015); Gan and Valdez (2016, 2019, 2020)) only use the contract attributes. Lin and Yang (2020a,b) are two exceptions in the literature. They also propose a two-phase sampling procedure for the selection of representative contracts, but the motivation in their method for the first phase sampling differs from ours. In their method, the first phase sample is used to develop an estimate for the error standard deviation in a linear surrogate model, as this parameter is needed for the determination of desirable representative contracts in their nested simulation framework.

Furthermore, our two-phase selection method can be applied naturally when an insurance company has already evaluated some representative VA contracts in their previous hedging program and would like to increase the valuation accuracy by selecting more representative contracts. In this case, these contracts with known values can be used as the initial selection ones and the posterior distribution can be established based on these known values for the conditional k-means procedure in the second phase to select a set of extra representative contracts.

The remainder of the chapter is structured as follows. Section 4.2 describes the kriging model based two-step valuation procedure used. Section 4.3 explains the details of our two-phase selection method. Section 4.4 compares the performance of our method with conventional selection methods, based on two numerical studies. Section 4.5 concludes the chapter.

4.2 The Two-step Valuation Paradigm

Consider a VA contract with $p+q$ attributes $\boldsymbol{x} := (x_1, \cdots, x_p, x_{p+1}, \cdots, x_{p+q})'$, where the first p variables are numerical and the remaining q are categorical. Assume, for each contract i, the contract value y_i can be written as $y_i = v(\boldsymbol{x}_i)$ for an unknown function v. The VA portfolio valuation problem is to develop a good estimate of the value $\sum_{i=1}^{N} v(\boldsymbol{x}_i)$ for a large number (e.g. more than 100,000) of contracts $\{\boldsymbol{x}_i, i = 1, \cdots, N\}$. In principle we could evaluate $v(\boldsymbol{x}_i)$ for every $\boldsymbol{x}_i$ through nested simulations (Reynolds and Man (2008); Bauer et al. (2012)). However, this would be computationally prohibitive for a desirable precision.

As mentioned in the introduction section, a typical alternative valuation approach is to resort to a metamodelling framework that follows a two-step procedure to develop an estimate for the sum $\sum_{i=1}^{N} v(\boldsymbol{x}_i)$. In the first step, a selection method is applied to find "representative contracts." For example, the clustering method classifies the pool $\{\boldsymbol{x}_i, i = 1, \cdots, N\}$ into a number of clusters. The centroids of the resulting clusters are taken as representative contracts, which of course constitute a subset of the pool $\{\boldsymbol{x}_i, i = 1, \cdots, N\}$. In the second step, the value of $v(\boldsymbol{x}_i)$ for each selected representative contract $\boldsymbol{x}_i$ is numerically computed. Then a predictive model is established, usually by combining the information from the attribute data of all the contracts and the computed values of these representative contracts. Methods presented in the literature include kriging methods (Gan and Lin (2015); Gan and Valdez (2020)) and regression models (Gan and Valdez (2016, 2018); Gan (2018)). The established predictive model is finally applied to each remaining contract to eventually obtain an estimate for $\sum_{i=1}^{N} v(\boldsymbol{x}_i)$.

Below we describe the specific setting of the two-step metamodelling procedure that was adopted by Gan and Lin (2015); Gan and Valdez (2019, 2020). This is also the setting we will adopt in this chapter except that we will use a two-phase procedure for the selection of representative contracts.

- **Step 1 (Selection of representative contracts).** Select k representative contracts with attributes $\{\boldsymbol{x}_{\text{rep},1}, \boldsymbol{x}_{\text{rep},2}, \cdots, \boldsymbol{x}_{\text{rep},k}\}$ by a given method, e.g. the k-prototypes clustering method (Huang (1997)) or the hierarchical k-means method (Nister and Stewenius (2006)). Then, compute the values of the selected contracts:

$$\boldsymbol{y}_{\text{rep}} = (y_{\text{rep},1}, y_{\text{rep},2}, \cdots, y_{\text{rep},k})' = (v(\boldsymbol{x}_{\text{rep},1}), v(\boldsymbol{x}_{\text{rep},2}), \cdots, v(\boldsymbol{x}_{\text{rep},k}))'$$

  where $y_{\text{rep},i} = v(\boldsymbol{x}_{\text{rep},i})$ is the value of the ith representative contract, $i = 1, \ldots, k$.

- **Step 2 (Estimation of portfolio value using kriging)**. After the selection and valuation of the representative contracts, the universal kriging method is applied to estimate the values of the remaining contracts. This method views the values of the VA contracts as a realization of a multivariate normal random vector:

$$\boldsymbol{Y} := (Y_1, Y_2, \cdots, Y_N)' \sim \mathcal{N}\left((g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \cdots, g(\boldsymbol{x}_N))', \{\text{Cov}(Y_i, Y_j)\}_{N\times N}\right). \tag{4.1}$$

The mean vector, modelled through an attribution function, and the covariance matrix, are specified below.

We call the remaining $(N-k)$ contracts after the selection of the representative contracts in the VA portfolio *regular contracts* to distinguish them from the k evaluated representative contracts in Step 1, and label their attributes and values by $\boldsymbol{x}_{\text{reg},i}$ and $y_{\text{reg},i}$, respectively, $i = 1, \ldots, N-k$. When the value of a contract is treated as a random variable, we use the notation $Y_{\text{rep},i}$ for a representative contract and $Y_{\text{reg},i}$ for a regular contract. For each regular contract, the universal kriging method estimates its value through a linear combination of representatives' values:

$$\hat{Y}_{\text{reg},i} = \sum_{j=1}^{k} \lambda_{ij} Y_{\text{rep},j}$$

where $\sum\limits_{j=1}^{k} \lambda_{ij} = 1$. The weight λ_{ij} are obtained by minimizing the estimation error:

$$\begin{aligned} \min \ & Var(\hat{Y}_{\text{reg},i} - Y_{\text{reg},i}) \\ s.t. \ & \mathrm{E}[\hat{Y}_{\text{reg},i} - Y_{\text{reg},i}] = 0 \end{aligned} \tag{4.2}$$

To solve the above optimization problem, we need to first model the Gaussian distribution (i.e. mean and covariance) of Y. The detailed estimation procedure is as follows:

(2.1) Calculate the distance between $\boldsymbol{x}_{\text{reg},i}$ and each representative contract as well as the distances among representative contracts. For two contracts $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the distance is defined as (Huang et al., 2005):

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{l=1}^{p} w_l (x_{il} - x_{jl})^2 + \sum_{l=p+1}^{p+q} w_l \delta(x_{il}, x_{jl})} \tag{4.3}$$

where $\delta(x_{il}, x_{jl}) = \mathbb{1}_{\{x_{il} \neq x_{jl}\}}$ and w_l is the weight assigned to the l-th attribute. The above distance is a generalization of the Euclidean distance to incorporate categorical components. There are various choices of the weights w_l. For the numerical attributes, common choices are constant weights ($w_l = 1$) (Gan and Valdez (2020)) and variation-adjusted weights, i.e., $w_l = \frac{1}{R_l^2}$ where $R_l^2 = \frac{1}{N-1} \sum\limits_{i=1}^{N} \left(x_{il} - \frac{1}{N} \sum\limits_{i=1}^{N} x_{il} \right)^2$ (Gan and Lin (2015)).

For the categorical attributes, constant weights are usually used (Gan and Lin (2015); Gan and Valdez (2020)).

(2.2) Choose a *semivariogram function* to model the covariance among contracts:

$$\mathrm{Cov}(Y_i, Y_j) = \sigma^2 - \gamma(D(\boldsymbol{x}_i, \boldsymbol{x}_j)). \tag{4.4}$$

The following are three common semivariogram functions (Stanford and Vardeman (1994)):

(a) Exponential semivariogram function

$$\gamma(h) = \begin{cases} 0, & h = 0 \\ a + (\sigma^2 - a)\left(1 - e^{-\frac{3h}{r}}\right), & h > 0 \end{cases} \tag{4.5}$$

(b) Spherical semivariogram function

$$\gamma(h) = \begin{cases} 0, & h = 0 \\ a + (\sigma^2 - a)\left(\frac{3h}{2r} - \frac{h^3}{2r^3}\right), & 0 < h \le r \\ \sigma^2, & \text{otherwise} \end{cases} \tag{4.6}$$

(c) Gaussian semivariogram function

$$\gamma(h) = \begin{cases} 0, & h = 0 \\ a + (\sigma^2 - a)\left(1 - e^{-\frac{3h^2}{r^2}}\right), & h > 0 \end{cases} \tag{4.7}$$

For a valid semivariogram function, we need $\sigma^2 \ge 0$, $a \ge 0$ and $r \ge 0$. The calibration of these parameter values is obtained based on the computed values of the selected representative contracts; see Appendix C.1 for details.

(2.3) Model the mean vector through an attribution function (Stanford and Vardeman (1994); Gan and Lin (2015)):

$$g(\boldsymbol{x}) = f(\boldsymbol{x})'\boldsymbol{\beta} = (1, x_1, \cdots, x_p, f_1(x_{p+1}), \cdots, f_q(x_{p+q}))\boldsymbol{\beta} \tag{4.8}$$

where $f(\boldsymbol{x}) = (1, x_1, \cdots, x_p, f_1(x_{p+1})', \cdots, f_q(x_{p+q})')'$ is the attribution function and $\boldsymbol{\beta}$ is a parameter vector. Recall that $x_{p+1}, \cdots, x_{p+q}$ are categorical. For $1 \le i \le q$, suppose x_{p+i} has m levels: $L_1, \cdots, L_m$, then $f_i(x_{p+i})$ is an $(m-1)$-dimensional vector

$$(\mathbb{1}_{\{x_{p+i}=L_2\}}, \cdots, \mathbb{1}_{\{x_{p+i}=L_m\}})'.$$

In this case, the mean $g(\boldsymbol{x})$ is the linear combination of all attributes (and categorical levels).

(2.4) Let $\mathbf{1}$ denote a vector of ones of length k. Using Lagrange multipliers to solve the minimization problem (4.2),[1] we will end up with solving the following linear system for weights $\boldsymbol{\lambda}_i$ and Lagrange multipliers $\boldsymbol{\eta}_i$ under the constraint $\mathbf{1}'\boldsymbol{\lambda}_i = 1$:

$$\begin{pmatrix} \boldsymbol{B}_{\text{rep}} & \boldsymbol{A}_{\text{rep}} \\ \boldsymbol{A}'_{\text{rep}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\eta}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{B}_{\text{reg},i} \\ \boldsymbol{A}_{\text{reg},i} \end{pmatrix},$$

where $\boldsymbol{A}_{\text{rep}} = (f(\boldsymbol{x}_{\text{rep},1}), f(\boldsymbol{x}_{\text{rep},2}), \cdots, f(\boldsymbol{x}_{\text{rep},k}))'$ is the attribution matrix of the representative contracts, and $\boldsymbol{A}_{\text{reg},i} = f(\boldsymbol{x}_{\text{reg},i})$ is the attribution vector of the contract $\boldsymbol{x}_{\text{reg},i}$. $\boldsymbol{B}_{\text{rep}} = \{\gamma(D(\boldsymbol{x}_{\text{rep},j}, \boldsymbol{x}_{\text{rep},l}))\}_{k\times k}$ is the semivariogram matrix of the representative contracts, and $\boldsymbol{B}_{\text{reg},i} = \{\gamma(D(\boldsymbol{x}_{\text{reg},i}, \boldsymbol{x}_{\text{rep},j}))\}_{k\times 1}$ is the semivariogram vector between the contract $\boldsymbol{x}_{\text{reg},i}$ and the representative contracts.

(2.5) The estimated value of the contract $\boldsymbol{x}_i$ is given by

$$\hat{v}(\boldsymbol{x}_{\text{reg},i}) = \boldsymbol{\lambda}_i \cdot \boldsymbol{y}_{\text{rep}}.$$

4.3 Two-Phase Selection

In this section, we introduce our two-phase selection method for determining representative contracts to be used in the metamodelling procedure described in the preceding section.

4.3.1 General Procedure of the Two-Phase Selection Method

The two-phase selection method breaks the first step of the metamodelling procedure (i.e., selection of the representative contracts) into two phases, which we label as Phases 1.1 and 1.2, as described below:

- **Phase 1.1 (Initial selection):** Apply the k-prototypes clustering method (Huang (1997)) or the hierarchical k-means method (Nister and Stewenius (2006)) to the attributes data of all the VA contracts and identify k contracts

[1] For details, see Stanford and Vardeman (1994).

that are closest to these k centroids, respectively. We then further apply a k_1-clustering method to the attribute data of these obtained k contracts to identify k_1 contracts that are closest to the k_1 centroids, where $k_1 < k$. The resulting k_1 contracts constitute a subset of the k contracts from the clustering procedure. Then, we compute the values of these identified k_1 contracts.

- **Phase 1.2 (Conditional selection):** In this step, we use the clustering method again to select $(k - k_1)$ representatives, out of the remaining $(N - k_1)$ candidate contracts. Since we now have the values of the k_1 contracts, which contain useful information about the underlying model $v(\cdot)$, we should not neglect their values when selecting new representative contracts. Hence, in this conditional selection phase, we make use of the information from the computed values of the first k_1 representative contracts in addition to the attribute data of all the VA contracts. While there might be alternative ways to incorporate the computed values of the representatives from the first phase, the specific conditional selection procedure in our method is described in Section 4.3.2. It is based on the posterior distribution of the working assumption (4.1) conditional on these computed values of k_1 representative contracts.

Remark 2. *The above two-phase selection procedure is also appealing in practice, where an insurance company has valuation results for some VA contracts (which may or may not be included in the currently prevailing VA portfolio) and decides to select more representative contracts to develop a more reliable valuation of its VA portfolio. In such a situation, these VA contracts with values available can be viewed as the k_1 contracts in Phase 1.1, and Phase 1.2 is focused on selecting additional representative contracts that are most conducive to improving the VA portfolio valuation, based on the known k_1 contract prices.*

4.3.2 The Posterior Distribution and a Distance Measure

The conditional selection phase is conducted under the working assumption that the value vector $\boldsymbol{Y}$ of the VA portfolio follows a Gaussian process as shown in (4.1). The mean is described through an attribution function and the covariance is described through a semivariogram function. We write

$$\boldsymbol{Y} = (\boldsymbol{Y}_{\text{pre}}, \boldsymbol{Y}_{\text{can}})' \sim \mathcal{N}\left((\boldsymbol{\mu}'_{\text{pre}}, \boldsymbol{\mu}'_{\text{can}})', \begin{pmatrix} \Sigma_{\text{pre}} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{\text{can}} \end{pmatrix}\right), \tag{4.9}$$

where $\boldsymbol{Y}_{\text{pre}}$ denotes the vector that contains the values of the k_1 representative contracts selected in Phase 1.1, and $\boldsymbol{Y}_{\text{can}}$ is the vector containing the values of all the remaining $(N-k_1)$ contracts. In the above, the mean vector and the covariance matrix are also partitioned accordingly. Σ_{12} and Σ_{21} are the covariance matrices between the two random vectors.

Under the working assumption (4.1), we can compute the posterior joint Gaussian distribution of the N contract values conditional on the computed values of the k_1 representatives selected in Phase 1.1 as follows:

$$(\boldsymbol{Y}_{\text{pre}}, \boldsymbol{Y}_{\text{can}})'\Big| (\boldsymbol{Y}_{\text{pre}} = \boldsymbol{y}_{\text{pre}}) \sim \mathcal{N}\left((\boldsymbol{y}'_{\text{pre}}, (\boldsymbol{\mu}^*_{\text{can}})')', \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma^*_{\text{can}} \end{pmatrix} \right), \tag{4.10}$$

where $\mathbf{0}$ denotes a matrix of all zeros in the appropriate dimension,

$$\boldsymbol{\mu}^*_{\text{can}} = \boldsymbol{\mu}_{\text{can}} + \Sigma_{21}\Sigma^{-1}_{\text{pre}}(\boldsymbol{y}_{\text{pre}} - \boldsymbol{\mu}_{\text{pre}}), \tag{4.11}$$

and

$$\Sigma^*_{\text{can}} = \Sigma_{\text{can}} - \Sigma_{21}\Sigma^{-1}_{\text{pre}}\Sigma_{12}. \tag{4.12}$$

For convenience, we call the distribution in (4.10) the posterior distribution and the one in (4.9) the prior distribution.

From the kriging modelling procedure described in Section 4.2, one can see that two contracts with similar attributes have similar mean and variance in their values. Moreover, in this case, their values have a strong positive correlation. Furthermore, if the distributions of two contract values are different enough from each other, then their corresponding attributes must differ from each other significantly. This means that the heterogeneity and homogeneity among the VA contracts in their attributes can be captured by those in the distributions of their values. We choose the remaining $(k-k_1)$ representative contracts by a clustering analysis based on the posterior distribution in (4.10). A natural statistical distance for two contracts $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (with value random variables Y_i and Y_j, respectively) to be used in the clustering analysis is as follows:

$$\begin{aligned} \mathrm{E}_p[(Y_i - Y_j)^2] &= \mathrm{Var}(Y_i - Y_j) + (\mathrm{E}[Y_i] - \mathrm{E}[Y_j])^2 \\ &= (\sigma^*_i)^2 + (\sigma^*_j)^2 - 2\sigma^*_i\sigma^*_j\rho^*_{ij} + (\mu^*_i - \mu^*_j)^2, \end{aligned} \tag{4.13}$$

where E_p denotes the expectation operator under the posterior distribution (4.10), μ_i and σ_i denote the mean and the standard deviation of the value variable Y_i, respectively, and ρ_{ij} is the correlation coefficient between Y_i and Y_j under the posterior distribution. When one of the two contracts is among those preselected, its variance is zero and thus the term involving the correlation coefficient in the above equation is also zero.

In the implementation, we instead use the following distance in the clustering analysis for the conditional selection of the remaining $(k-k_1)$ representative contracts:

$$W(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j)=||\boldsymbol{\theta}_i-\boldsymbol{\theta}_j||^2=(\mu_i^*-\mu_j^*)^2+(\sigma_i^*-\sigma_j^*)^2, \tag{4.14}$$

where $\boldsymbol{\theta}_i=(\mu_i^*,\sigma_i^*)$ collects the posterior distributional attributes. Such a distance can also be viewed as the Wasserstein distance for Gaussian distributions (Kantorovich (1960)). The reason for us to consider such a distance is as follows. This distance comes from (4.13) by assuming $\rho_{ij}=1$. In the application of large VA portfolio valuation, the value variables turn out to have a strong positive correlation under the kriging model. Furthermore, with this simplified distance, we only need to compute the mean and variance of each contract value under the posterior distribution and do not need to compute the covariance. Avoiding computing the covariance can save a considerable amount of computation time and reduce memory usage in the computational implementation of the clustering analysis.

To implement a clustering analysis based on the distance (4.14), we need to develop an estimate for the posterior mean and the posterior variance of the remaining $(N-k_1)$ candidate contracts. The estimation is carried out using the following steps:

- First, we follow the parametric model in (4.8) and apply an Ordinary Least Squares (OLS) regression with the data of the k_1 initially selected contracts (including both the computed contract values and the attribute data). The OLS regression gives us an estimate of the coefficient vector $\boldsymbol{\beta}$, and substituting its value into (4.8) leads to an estimate for the mean for each contract. This provides an estimate for the prior mean vector $\boldsymbol{\mu}_{\text{can}}$.

- Second, noticing that each element of the prior covariance matrix is modelled by equation (4.4), we follow the procedure described in Appendix C.1 using the data of the k_1 preselected representative contracts for the calibration.

- Finally, we substitute the estimates of the prior mean and the prior covariance matrix into equations (4.11) and (4.12) to get the estimates for the posterior mean vector $\boldsymbol{\mu}^*_{\text{can}}$ and the posterior covariance matrix Σ^*_{can}.

4.3.3 A Conditional k-means Algorithm

To enhance the representativeness of the selected contracts, we consider applying a conditional k-means algorithm for the determination of the $(k-k_1)$ representative contracts in Phase 1.2. We call it a conditional k-means algorithm because we take the k_1 initially selected contracts as k_1 centroids in the clustering analysis and search for the remaining $(k-k_1)$ centroids to minimize the total loss defined through the distance in equation (4.14).

Recall that $\boldsymbol{\theta}_i = (\mu_i^*, \sigma_i^*)$ collects the posterior distributional attributes of contract i, $i = 1, 2, \ldots, N$. Denote $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$, and let $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{k_1}\}$ be the k_1 known centroids. We choose the remaining $(k-k_1)$ centroids $\{\boldsymbol{c}_{k_1+1}, \boldsymbol{c}_{k_1+2}, \ldots, \boldsymbol{c}_k\}$ by minimizing the overall loss:

$$L(\boldsymbol{z}, \boldsymbol{c}) = \sum_{j=1}^{k} \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j))^2,$$

where $\boldsymbol{c} := \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$, $\boldsymbol{z} = (z_1, z_2, \ldots, z_N)'$, $z_i \in \{1, 2, \ldots, k\}$, are the clusters assigned to the N points so that $z_i = j$ when the point $\boldsymbol{\theta}_i$ is assigned to the jth cluster.

We follow a similar algorithm to the one described in Gan and Lin (2015) for searching for the remaining $(k-k_1)$ centroids:

1. We first randomly select $(k-k_1)$ centroids from $\boldsymbol{\theta} \setminus \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{k_1}\}$. Denote the initial centroids as $\boldsymbol{c}^{(0)} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{k_1}, \boldsymbol{c}_{k_1+1}^{(0)}, \ldots, \boldsymbol{c}_k^{(0)}\}$.

2. Then, each point $\boldsymbol{\theta}_i$ is assigned to a cluster by

   $$z_i^{(0)} = \underset{j=1,\cdots,k}{\arg\min}\, W(\boldsymbol{\theta}_i, \boldsymbol{c}_j^{(0)}), \quad i = 1, \cdots, N.$$

   If $\boldsymbol{\theta}_i$ has the same distance to more than one centroid, it will be randomly assigned to one of those clusters.

3. For clusters with centroids $\boldsymbol{c}^{(0)}$, we keep the first k_1 centroids and update the last $(k-k_1)$ centroids by

   $$c_{jl}^{(1)} = \frac{1}{\#A_j} \sum_{z_i=j} \theta_{il}, \quad l = 1, 2, \quad \text{and} \quad j = k_1 + 1, \cdots, k, \tag{4.15}$$

   where $\#A_j$ denotes the number of elements in $A_j := \{i = 1, \ldots N | z_i = j\}$, i.e., the number among the N points that are assigned to cluster j.

4. We repeat the above two steps until the centroids do not change or until we reach a preset upper limit of the iteration number.

Note that the above clustering algorithm does not guarantee convergence to a global optimum. Therefore, the output depends on the initialization of the centroids. A common solution as for the conventional k-means clustering procedure is to try a number of randomized initializations and take the best one.

The following results guarantee the convergence of the above clustering algorithm with fixed centroids. That is, the algorithm finds a unique set of centroids after finitely many iterations.

Lemma 4.3.1. *Let centroids $\boldsymbol{c} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ be fixed. Define the cluster assignment for point $\boldsymbol{\theta}_i$ as*

$$\beta_i(\boldsymbol{c}) = \mathop{\arg\min}_{j \in \{1,2,\ldots,k\}} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j))^2, \;\; i = 1, 2, \ldots, N.$$

Let $\beta(\boldsymbol{c}) = (\beta_1(\boldsymbol{c}), \beta_2(\boldsymbol{c}), \ldots, \beta_N(\boldsymbol{c}))$. Then, for any assignment $\boldsymbol{z}$, it holds that

$$L(\boldsymbol{z}, \boldsymbol{c}) \geq L(\beta(\boldsymbol{c}), \boldsymbol{c}).$$

Proof. The optimality of $\beta(\boldsymbol{c})$ immediately implies

$$L(\boldsymbol{z}, \boldsymbol{c}) = \sum_{j=1}^{k} \sum_{z_i = j} (W(\boldsymbol{\theta}_i - \boldsymbol{c}_j))^2 \geq \sum_{j=1}^{k} \sum_{\beta_i(\boldsymbol{c}) = j} (W(\boldsymbol{\theta}_i - \boldsymbol{c}_j))^2 = L(\beta(\boldsymbol{c}), \boldsymbol{c}).$$

□

Lemma 4.3.2. *Let $\boldsymbol{z}$ be any fixed cluster assignment. Given the first k_1 cluster centroids fixed, define the centroid of cluster j, $j = k_1 + 1, k_1 + 2, \ldots, k$, as*

$$\alpha_j(\boldsymbol{z}) = \frac{1}{\#A_j} \sum_{z_i = j} \boldsymbol{\theta}_i.$$

Let $\alpha(\boldsymbol{z}) = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{k_1}, \alpha_{k_1+1}(\boldsymbol{z}), \ldots, \alpha_k(\boldsymbol{z}))$. Then, for any $\boldsymbol{c}^* = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{k_1}, \boldsymbol{c}^*_{k_1+1} \ldots, \boldsymbol{c}^*_k\}$ with $\boldsymbol{c}^*_i \in \mathbb{R}^2$, it holds that

$$L(\boldsymbol{z}, \boldsymbol{c}^*) \geq L(\boldsymbol{z}, \alpha(\boldsymbol{z})).$$

Proof. See Appendix C.2. □

Theorem 4.3.3. *The loss function of the conditional k-means algorithm, with the first k_1 centroids $(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{k_1})$ known and fixed, converges.*

Proof. Let $l_{n-1} = L(\boldsymbol{z}^{(n-1)}, \boldsymbol{c}^{(n-1)})$ be the loss function at iteration $(n-1)$, for given centroids $\boldsymbol{c}^{(n-1)}$ and cluster assignments $\boldsymbol{z}^{(n-1)}$. We also carry the same meanings of all the notation from Lemmas 4.3.1 and 4.3.2.

The nth iteration updates $\boldsymbol{z}^{(n-1)}$, the assignment of each point $\boldsymbol{\theta}_i$, to $\boldsymbol{z}^{(n)} := (z_1^{(n)}, \cdots, z_N^{(n)})$ with

$$z_i^{(n)} = \operatorname*{arg\,min}_{j \in \{1,2,\ldots,k\}} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j^{(n-1)}))^2, \ \ i = 1, 2, \ldots, N.$$

Thus, by Lemma 4.3.1, we have $L(\boldsymbol{z}^{(n-1)}, \boldsymbol{c}^{(n-1)}) \geq L(\boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n-1)})$. Furthermore, the nth iteration also updates the centroids to $\boldsymbol{c}^{(n)} := (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{k_1}, \boldsymbol{c}_{k_1+1}^{(n)}, \ldots, \boldsymbol{c}_k^{(n)})$ with

$$\boldsymbol{c}_j^{(n)} = \frac{1}{\#A_j^{(n)}} \sum_{z_i^{(n)}=j} \boldsymbol{\theta}_i, \ \ j = k_1 + 1, \ldots, k, \tag{4.16}$$

where $\#A_j^{(n)}$ denotes the number of elements in $A_j^{(n)} := \{i = 1, \ldots N | z_i^{(n)} = j\}$. Therefore, by Lemma 4.3.2, we have $L(\boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n-1)}) \geq L(\boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)})$.

Combining the above, we obtain

$$l_{n-1} = L(\boldsymbol{z}^{(n-1)}, \boldsymbol{c}^{(n-1)}) \geq L(\boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n-1)}) \geq L(\boldsymbol{z}^{(n)}, \boldsymbol{c}^{(n)}) = l_n,$$

which means that the loss sequence $\{l_n\}$ is non-increasing. Furthermore, the sequence is obviously bounded from below by zero. Therefore, the sequences, $\{l_n\}$ converge.

□

Remark 3. *When implementing the conditional or unconditional k-means algorithm, although unlikely, cycling can be an issue, i.e. centroids change but the overall loss does not reduce. In this case, the algorithm stops when the loss is no longer reduced (after a certain number of iterations). According to Selim and Ismail (1984), this issue does not contradict the local convergence in the above theorem.*

Remark 4. *This conditional k-means algorithm can be applied to higher dimensional data. We can also derive similar conditional algorithms for other conventional algorithms such as the k-prototype algorithm and mini-batch k-means algorithm. For the conditional k-prototype algorithm, the distance is defined by equation (4.3). In the* *updating step 3, the numerical attributes are still updated by equation* (4.15)*. Each categorical attribute is updated by the mode i.e. the categorical level that appears most in the cluster. For the conditional mini-batch k-means algorithm, the convergence proof will be similar to that for the conventional mini-batch k-means algorithm (Bottou and Bengio (1994); Sculley (2010)).*

4.4 Numerical Examples

In this section, we implement two numerical studies based on a dataset of synthetic variable annuities from Gan and Valdez (2017) to illustrate the performance of our two-phase selection method for the valuation of VA portfolios.

4.4.1 Variable Annuity Contracts

Gan and Valdez (2017) build a synthetic dataset that contains 10,000 VA contracts for each of 19 products. They use Monte Carlo simulations to obtain the fair market values and relevant Greeks (Delta and Rho) for each contract.[2] The fair market values will be used as the "true" contract values to measure the performance of different valuation methods in our studies. Instead of using their original attribute data, we process the data following Gan and Valdez (2020):

(1) We convert dates in their dataset into age, maturity, and time-to-maturity for each contract.

(2) We drop the base fee and underlying investment fund fees as they are set to be universal for all contracts.

(3) We drop rider fees, withdrawal rates, and roll-up rates as they are determined by the guarantee types.

Tables 4.1 and 4.2 summarize the categorical and numerical attributes of the VA contracts. The explanations of the guarantee types are in Appendix C.3. For more details about the design of the VA contracts, see Gan and Valdez (2017).

We normalize all numeric attributes to be distributed in $[0, 1]$ in our analysis as in Gan and Lin (2015); Gan and Valdez (2019, 2020). We test some benchmark selection procedures and our two-phase selection procedure below. We use

[2] These public datasets are accessible at https://www2.math.uconn.edu/~gan/software.html.

Attribute	Values	Distributions
Guarantee type	DBRP, DBRU, DBSU, ABRP, ABRU, ABSU	10,000 each
	IBRP, IBRU, IBSU, MBRP, MBRU, MBSU	
	WBRP, WBRU, WBSU, DBAB, DBIB, DBMB, DBWB	
Gender	Male, Female	60%, 40%

Table 4.1: Categorical attributes of VA contracts

Attribute	Min.	1st Q.	Median	Mean	3rd Q.	Max.
Age	34.52	42.03	49.45	49.49	56.96	64.46
Time-to-maturity	0.59	10.34	14.51	14.54	18.76	28.52
Guaranteed benefit	50001.72	179758.97	303524.62	313507.22	427544.13	989204.53
GMWB balance	0.00	0.00	0.00	36140.74	0.00	499708.73
Withdrawn amount	0.00	0.00	0.00	21927.80	0.00	499585.73
Fund 1 value	0.00	0.00	8299.21	26611.38	39208.90	921548.70
Fund 2 value	0.00	0.00	8394.07	26044.48	38463.42	844322.70
Fund 3 value	0.00	0.00	4941.89	17391.42	24251.44	580753.42
Fund 4 value	0.00	0.00	4225.33	14507.35	20755.88	483936.90
Fund 5 value	0.00	0.00	7247.69	21041.04	32111.72	494381.61
Fund 6 value	0.00	0.00	8555.75	26569.62	39241.12	872706.64
Fund 7 value	0.00	0.00	6602.11	21505.81	31087.78	634819.08
Fund 8 value	0.00	0.00	6254.55	19990.40	29404.16	562485.37
Fund 9 value	0.00	0.00	5943.19	19646.92	28100.22	663196.22
Fund 10 value	0.00	0.00	6738.22	21002.82	31255.73	599675.34

Table 4.2: Statistical summary of numerical attributes of VA contracts

two conventional clustering methods: the k-prototypes method ("k-prototypes" for short, Section 4.4.2) and hierarchical k-means method ("hkmeans" for short, Section 4.4.3). For both methods, we apply them in our two-phase selection procedure ("TP. SD"[3] for short). For the conventional clustering procedure, we choose k initial representative contracts. For our procedure, we use the conventional clustering procedure to choose $k_1 = k/2$ initial representatives and select the remaining representatives through our conditional clustering method based on the statistical distance.

4.4.2 Numerical Study 1: *k*-Prototypes

In this first numerical study, we use the conventional k-prototypes method as the benchmark method. We also use the k-prototypes method in Phase 1.1 for the initial selection of representative contracts in our two-phase procedure. The k-prototypes method is a conventional clustering method, feasible for categorical data. Similarly to the k-means clustering method, the distance between two contracts (data points) is employed to divide the dataset into clusters. For the k-prototypes method, the distance is defined in equation (4.3), where the contribution of the categorical component to the total distance is measured by the weight w_l. We use the variation-adjusted weights $\left(w_l = \frac{1}{R_l^2}\right)$ for the numerical attributes and constant weights ($w_l = 1$) for the categorical attributes, following (Gan and Lin (2015)).

However, the use of distance (4.3) makes the k-prototypes method computationally expensive because it involves the search for the mode as we have previously pointed out in Remark 4. Hence, for this study, we use a subset of the synthetic data from Gan and Valdez (2017).[4] In particular, we use the data for VA contracts with DBSU (GMDB with annual ratchet), WBSU (GMWB with annual ratchet) and DBWB (GMDB + GMWB with annual ratchet) guarantee types as examples. In total, we have 30,000 contracts. Also, we reduce the number of attribute variables by collecting the 10 fund values into an aggregate fund value with a statistical summary in Table 4.3. Note that each VA contract usually can only have one non-zero value in the 10 funds in the raw data, and this conversion of data does not lose much information.

For the two valuation methods, we test different values of k: 150, 300, and 450, representing 0.5%, 1%, and 1.5% of the entire dataset, respectively.[5] We use

[3] SD stands for the statistical distance - in particular the Wasserstein distance in this chapter.

[4] Note that the expensive computational cost is not from our framework, but from the k-prototypes clustering method.

[5] Similar proportions are used in Gan and Lin (2015).

Attribute	Min.	1st Q.	Median	Mean	3rd Q.	Max.
Aggregate fund value	0.00	63317.03	133318.13	172387.05	265302.99	865452.36

Table 4.3: Aggregate Fund Value of VA Contracts

the Gaussian semivariogram function (4.7) to model the covariance matrix of the Gaussian distribution (4.1). In our two-phase selection procedure, we take $k_1 = 75, 150$, and 225, respectively, which comes from setting $k_1 = k/2$.

Both valuation procedures give us the estimated values of the entire portfolio. We then compare the estimated values to the "true" values provided in the synthetic dataset, measured by the absolute percentage error (APE), at the portfolio level,

$$\mathrm{APE} = \frac{\left| \sum_{i=1}^{n} \hat{y}_i - \sum_{i=1}^{n} y_i \right|}{\sum_{i=1}^{n} y_i}.$$

We record the computational cost of each procedure, as our aim is to reduce estimation error while controlling the computational cost. The computational cost has two parts: the time to select representatives, and the time to estimate candidate contracts' values (similar for all methods). The biggest difference is the first part, which originates from the different selection designs.

In the clustering methods such as k-means and k-prototypes, the initialization of the cluster centroids is chosen randomly. This randomness could affect the final clusters since a single run of the methods only finds a local optimum. Hence, to mitigate the impact of the random initialization on the clustering results, we repeat the procedure under 100 random seeds.

Table 4.4 summarizes the average estimation errors and average computational costs of both the k-proptotype and our two-phase selection methods when applied to the fair market value, with different values of k. The two columns "Difference" record the percentage deduction by our two-phase selection method compared with the k-proptotype method, in terms of average APE and average computational time over the 100 random seeds, respectively. A negative value in the column "Difference" means a smaller value from our two-phase method compared with the conventional benchmark, and a positive value suggests a bigger value from our method. The detailed estimation errors and computational costs under each random seed are reported in Tables C.2-C.4, in Appendix C.4.

On average, our two-phase selection method achieves lower relative errors for each k, compared to the conventional k-prototypes method. This indicates that we should not neglect the information provided by the existing representatives. In particular, our two-phase selection method based on the Wasserstein distance reduces the estimation error 22.36%, 38.81%, and 53.12% when k is equal to 150, 300, and 450, respectively. The standard errors are also smaller compared to those of the conventional procedure. At the same time, our procedure costs 19.47%, 14.57%, and 8.79% more computational time, respectively. The improvement in the estimation error compensates for the extra computational time. Moreover, when the value of k increases from 150 to 450, the increase in the computational cost becomes less, while the reduction in the estimation error becomes more significant. From Table 4.4, we have another finding when comparing the two selection methods with different k's. Specifically, our procedure achieves 0.0184 estimation error when k is equal to 150, which is less than that of the conventional result when k is equal to 300 (0.0201). A similar result holds when k for our procedure is 300 and for the conventional procedure is 450.

	Average APE			Average Time		
k	k-prototypes	TP.SD	Difference	k-prototypes	TP.SD	Difference
150	0.0237 (0.0019)	0.0184 (0.0014)	-22.36%	2717.05s	3246.08s	+19.47%
300	0.0201 (0.0016)	0.0123 (0.0010)	-38.81%	3501.16s	4011.43s	+14.57%
450	0.0192 (0.0013)	0.0090 (0.0007)	-53.12%	5657.59s	6155.04s	+8.79%

Table 4.4: Performance of methods: Average APE and computational time with different k's. The values in brackets are standard errors.

We could also look at the performance under each random seed in Tables C.2-C.4. We conduct a paired t test to test whether our procedure produces statistically significantly smaller estimation errors than the conventional procedure. Our hypothesis test is

$$H_0 : \text{APE}_{k-\text{prototypes}} = \text{APE}_{\text{TP.SD}} \qquad H_a : \text{APE}_{k-\text{prototypes}} > \text{APE}_{\text{TP.SD}}.$$

The results are summarized in Table 4.5. From the results, the estimation errors from our procedure are significantly smaller, especially when k is large. These results are consistent with those on the average level, giving evidence of the benefits of the conditional valuation procedure.

k	Mean Difference	t Statistics	p-value	Confidence Interval
150	0.0053	2.3922	0.0093	$(0.0016, \infty)$
300	0.0077	4.5898	< 0.0001	$(0.0049, \infty)$
450	0.0102	7.1934	< 0.0001	$(0.0078, \infty)$

Table 4.5: Summary of the paired t tests for the APE when using the k-prototypes method.

In addition to the APE, we also consider the percentage absolute error (PAE) of the estimates (Liu and Tan (2021))

$$\text{PAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} y_i}.$$

A lower PAE means the estimation is more accurate. In contrast to the APE, the PAE considers both the portfolio level deviation and individual level deviation, from the true contract values. The average estimation errors for the two models and their differences are reported in Table 4.6. The conclusions based on the PAE are similar to those based on the APE. Our two-phase selection method produces smaller estimation errors which decrease when k increases. When k is equal to 150, the percentage reduction in the estimation error does not compensate for the percentage increase in the computational cost. However, there is a promising trend that when k is larger, the percentage reduction in the estimation error becomes larger and exceeds the percentage increase in the computational cost.

k	k-prototypes	TP.SD	Difference
150	0.2133 (0.0328)	0.1874 (0.0320)	-12.12%
300	0.1834 (0.0224)	0.1476 (0.0178)	-19.51%
450	0.1662 (0.0216)	0.1283 (0.0153)	-22.81%

Table 4.6: Performance of methods: Average PAE with different k's. The values in brackets are standard errors.

4.4.3 Numerical Study 2: Hierarchical k-Means

In the second numerical study, we implement another clustering method, the hierarchical k-means (Nister and Stewenius (2006)) method, to select representatives from the entire synthetic VA dataset as proposed by Gan and Valdez (2020). Because the conventional k-prototypes method spends a very large amount of time to calculate the distances among contracts (as can be seen from the first study), it is inefficient to deal with 190,000 contracts. The hierarchical k-means method, however, is more efficient. It uses the k-means method to repeatedly divide the largest cluster into two clusters until the number of clusters is equal to k. The dividing process is similar to creating a binary tree. Hence, tree-based algorithms can be used to improve the computational efficiency of the hierarchical k-means method.

However, as the name suggests, the hierarchical k-means method cannot handle categorical data. It can only process dummy binary variables. Hence, following Gan and Valdez (2020) we convert the categorical variable, Guarantee type, into 19 dummy binary variables, each of which represents a particular guarantee type. The distance we use is still the one given by equation (4.3). In this study, we assume all weights w_l are equal to 1 and use a binary tree algorithm to implement the hierarchical k-means method as in Gan and Valdez (2020). The binary tree algorithm does not accommodate the conditional clustering feature as required in our conditional selection phase (i.e., Phase 1.2). Moreover, the exact version of the conditional k-means algorithm we proposed in Section 4.3.3 has the same limitation as the conventional k-means algorithm that it is computationally expensive for a large dataset. Therefore, in this study, we use the conditional mini-batch k-means algorithm which is an adaptation of the mini-batch k-means algorithm (Sculley (2010)) in the same spirit as we modify the conventional k-means algorithm to the conditional one. We use a mini-batch size of 1,000 and 10 iterations. This needs a similar computational time as the hierarchical k-means algorithm.

Due to the computational efficiency, we conduct more numerical experiments for this study. We implement the conventional selection procedure and our two-phase selection procedure with three semivariogram functions: exponential (4.5), spherical (4.6) and Gaussian semivariograms (4.7).

In this study, we use another approach to adjust for the randomness from the initialization in the k-means method. For each value of k, we run the hierarchical k-means clustering algorithm to select k representatives out of 190,000 contracts under 100 different random seeds. Under each seed, we obtain k representatives and the cluster assignments of all contracts. Then, we calculate the within-cluster distances and sum them up. Suppose $\{\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{in_i}\}$ are assigned to the i-th cluster whose centroid is $\boldsymbol{c}_i$, $i = 1, \ldots, k$. The sum of within-cluster distances[6] is defined as

$$\sum_{i=1}^{k} \sum_{j=1}^{n_i} D(\boldsymbol{x}_{ij}, \boldsymbol{c}_i).$$

The seed that yields the smallest sum of within-cluster distances is chosen as the best seed, in terms of clustering. The semivariogram function is not yet used in the initial selection, so the choice of the semivariogram function will not affect the seed selection. We only report the results under the selected seed. Note that this approach does not benefit our two-phase selection procedure, because we would use the distributions of the contract values to select the second batch of representatives.

Table 4.4.3 summarizes the quantiles of the representative contract values from the conventional selection method and our two-phase selection method, as well as the quantiles of the contract values of the whole VA portfolio when the number of representative contracts, k is set equal to 340 and the Gaussian semivariogram function is used in our conditional selection phase. Representatives are selected under the best seed for both methods described above. The table indicates that the quantiles of the representative contract values from our two-phase method are closer to those of the whole VA portfolio. This comes from the fact that selecting based on the posterior distribution in our method yields more similarity in the distribution of contract values between the representative set and the whole VA portfolio, which we have also mentioned in the introduction section.

	0.05%	0.5%	1%	99%	99.5%	99.95%
hkmeans	-21811.13	-19856.94	-16433.13	754018.93	852935.05	944307.75
TP.SD	-31021.91	-22067.18	-20690.93	757172.07	853480.68	1187245.85
VA portfolio	-36208.41	-26482.91	-22832.66	839914.48	976883.50	1384513.94

Table 4.7: Quantiles of contract values of representatives from the two selection methods and of all the VA contracts in the portfolio

In terms of the performance measurement, we use the APE and PAE as well as a new metric, R^2 (Gan and Valdez (2019, 2020)):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

[6] This is also the loss function in the conventional k-means and k-prototypes methods.

A higher R^2 means the estimation is more accurate. Similar to the PAE, R^2 also considers both the portfolio level deviation and individual level deviation, from the true contract values. We also record the computational costs of the two selection methods. The computational time still consists of two parts: selection and estimation. However, in this study, the selection time includes the time to repeat the initial selection 100 times to select the best seed. Hence, if we would like to select the final seed based on more experiments, the associated time is expected to increase.

Table 4.8 reports the valuation performance for different values of k and semivariogram functions. Every three rows are under a particular seed, the determination of which is discussed above. We do not report the relative differences in R^2 because they are of less interest.

The results indicate that the APEs are more volatile than the other two measurements, across different values of k and semivariogram functions, for our particular dataset. Especially for the conventional procedure, a larger k may not reduce the APE. This is because of the cancellation of errors. Note that this measurement will eventually decrease to zero when k increases to a large number. However, for our method, there is a generally decreasing trend in APE in Table 4.8. The PAE and R^2 do not have the cancellation of errors issue. In terms of these two measurements, the performance of the two procedures is improved when k is larger. For our particular dataset, the Gaussian semivariogram function seems to have the best estimation performance using our two-phase selection method, compared to the other semivariogram functions. The computational cost is greatly reduced, compared to the first study, which suggests the advantage of the hierarchical k-means method. When comparing the two valuation procedures, our procedure outperforms the conventional procedure in terms of all three measurements. The outperformance is consistent across different values of k and semivariogram functions. Moreover, the percentage reduction of the estimation error is larger than the percentage increase of the computational time.

In Table 4.8, we could also compare the PAE and R^2 across panels. We do not consider the APE because it does not have a clear trend across k's for the conventional procedure. In terms of the PAE, we can compare the two methods while keeping the computational costs at the same level. For example, we compare our method when k is equal to 220, to the conventional method when k is equal to 280. Both have comparable computational time, around 2,000 seconds. But the PAEs of our procedure are consistently better than those of the conventional procedure. In terms of the R^2, when k is equal to 170 and the semivariogram function is exponential, our method yields an R^2 of 0.8890. This value is even higher than that of the conventional

k	Semivariogram	APE			PAE		
		hkmeans	TP.SD	Difference	hkmeans	TP.SD	Difference
170	Exponential	0.0124	0.0083	-32.85%	0.5510	0.3828	-30.52%
	Spherical	0.0124	0.0078	-37.14%	0.5509	0.3755	-31.84%
	Gaussian	0.1761	0.0408	-76.82%	0.5178	0.3288	-36.50%
220	Exponential	0.0076	0.0044	-42.03%	0.5090	0.3911	-23.18%
	Spherical	0.0076	0.0007	-91.05%	0.5090	0.3734	-26.63%
	Gaussian	0.0452	0.0155	-65.74%	0.3890	0.2713	-30.25%
280	Exponential	0.0021	0.0017	-17.27%	0.5086	0.3842	-24.46%
	Spherical	0.0021	0.0016	-24.89%	0.5085	0.3878	-23.74%
	Gaussian	0.0077	0.0049	-36.49%	0.4163	0.2193	-47.32%
340	Exponential	0.0066	0.0056	-14.92%	0.4183	0.2735	-34.62%
	Spherical	0.0066	0.0054	-18.35%	0.4182	0.2604	-37.73%
	Gaussian	0.0015	0.0013	-17.48%	0.2682	0.2011	-25.02%
680	Exponential	0.0138	0.0042	-69.30%	0.2265	0.1959	-13.51%
	Spherical	0.0138	0.0021	-85.09%	0.2265	0.1939	-14.40%
	Gaussian	0.0102	0.0020	-80.53%	0.1910	0.1335	-30.08%
k	Semivariogram	R^2			Time		
		hkmeans	TP.SD	Difference	hkmeans	TP.SD	Difference
170	Exponential	0.8069	0.8890	-	1551.67s	1641.51s	+5.79%
	Spherical	0.8070	0.8927	-	1552.53s	1645.89s	+6.01%
	Gaussian	0.7921	0.9242	-	1571.11s	1653.43s	+5.24%
220	Exponential	0.7963	0.8879	-	1799.44s	1950.66s	+8.40%
	Spherical	0.7964	0.8947	-	1798.06s	1921.82s	+6.88%
	Gaussian	0.8625	0.9467	-	1783.55s	1902.97s	+6.70%
280	Exponential	0.7802	0.8920	-	2084.86s	2241.78s	+7.53%
	Spherical	0.7803	0.8875	-	2089.79s	2240.87s	+7.23%
	Gaussian	0.8257	0.9659	-	2095.36s	2255.56s	+7.65%
340	Exponential	0.8471	0.9398	-	2574.87s	2743.82s	+6.56%
	Spherical	0.8471	0.9438	-	2579.59s	2750.33s	+6.62%
	Gaussian	0.9324	0.9699	-	2541.81s	2783.18s	+9.50%
680	Exponential	0.9594	0.9650	-	6129.49s	6560.80s	+7.04%
	Spherical	0.9595	0.9664	-	6109.91s	6501.85s	+6.41%
	Gaussian	0.9707	0.9832	-	6137.96s	6541.23s	+6.57%

Table 4.8: Valuation performance based on the selected seed

method when k is equal to 340. From this perspective, our method not only achieves higher estimation accuracy in return for more computational cost but also improves the estimation at the same level of computational cost.

Figures 4.1-4.3 show the distributions of the estimated contract values from the two representative selection methods, compared to that of the contract values of the whole VA portfolio. The x-axis is truncated at -500 from the left and 1,000 from the right because the data beyond these limits are rather sparse. We use the Gaussian semivariogram function and k equal to 170, 340, and 680 as examples. From the graphs, our two-phase selection procedure yields much closer distributions to the true one, especially at the tails. This is consistent with the measurements in Table 4.8. Similar results for the quantiles are summarized in Table 4.4.3. These results would also explain why the APE from the conventional method does not decrease when k increases. Comparing the distribution of the conventional procedure when k is equal to 340 and 680 in Figure 4.1, the estimated values when k is equal to 340 deviate from the true ones at both the left and right tails, allowing for the cancellation of errors. However, when k is equal to 680, the left tail becomes thinner which reduces the error from the negative values, but it turns out to increase the error for the aggregated value. Note that when k is small, there is still a noticeable discrepancy between the distribution from our estimates and that from the true values, around zero. In the two-step valuation framework, representative contracts are selected to reflect heterogeneity in the whole portfolio. Hence, with a certain number of representatives (k), if we select contracts in the tails of the distribution, then there will be fewer contracts around zero. This leads to the discrepancy after interpolation. When k increases, more representatives around zero will be selected, compared to those in the tails. Therefore, the discrepancy becomes less significant for larger k. This is a general problem using the two-step valuation framework. But our method mitigates this issue compared to the conventional method, as attested by Figures 4.1-4.3.

k	Method	0.05%	0.5%	1%	99%	99.5%	99.95%
170	hkmeans	-343510.46	-188650.90	-151991.03	759447.62	838610.10	1025983.20
	TP.SD	-214848.33	-123143.22	-97526.29	825569.06	941026.28	1260804.40
340	hkmeans	-525002.40	-262815.49	-177997.82	788792.59	881756.17	1086587.33
	TP.SD	-184191.88	-99305.97	-77151.27	843470.98	972740.81	1371481.07
680	hkmeans	-173347.99	-99478.87	-77934.94	790973.37	899144.84	1133066.60
	TP.SD	-83186.35	-47369.67	-37525.00	852815.71	986951.59	1388607.68
True Values		-36208.41	-26482.91	-22832.66	839914.48	976883.50	1384513.94

Table 4.9: Quantiles of the estimated values based on two procedures and all contracts' true values.

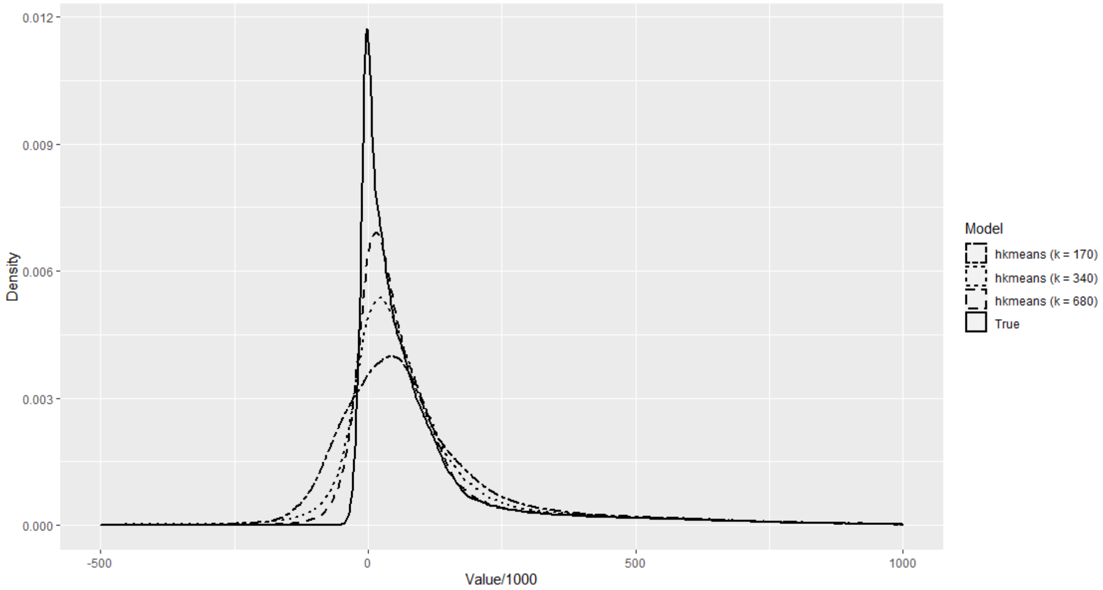

Figure 4.1: Distributions of the estimated values from the conventional procedure and the true contract values.

4.5 Conclusion

In this chapter, we study the valuation of a large variable annuity portfolio. The majority of the literature uses a two-step metamodelling framework to address this problem: selection of representative contracts followed by estimation based on the representatives. This framework relies on spatial models of the contract attributes such as the guarantee type, policyholder's age, and underlying fund values, to select representatives. We propose a two-phase selection method to model the contract values and select additional representatives based on this model. In particular, we use the Gaussian distribution to model the contract values and use the Wasserstein distance in selection. Our valuation procedure works with different clustering methods such as the k-prototypes method and hierarchical k-means method. Based on numerical studies, our method selects representatives that have closer distribution quantiles to the true ones. Two extensive numerical studies confirm that our method outperforms the conventional one in terms of estimation errors. While keeping the computational cost at the same level, our method still achieves higher estimation accuracy with a smaller number of representatives.

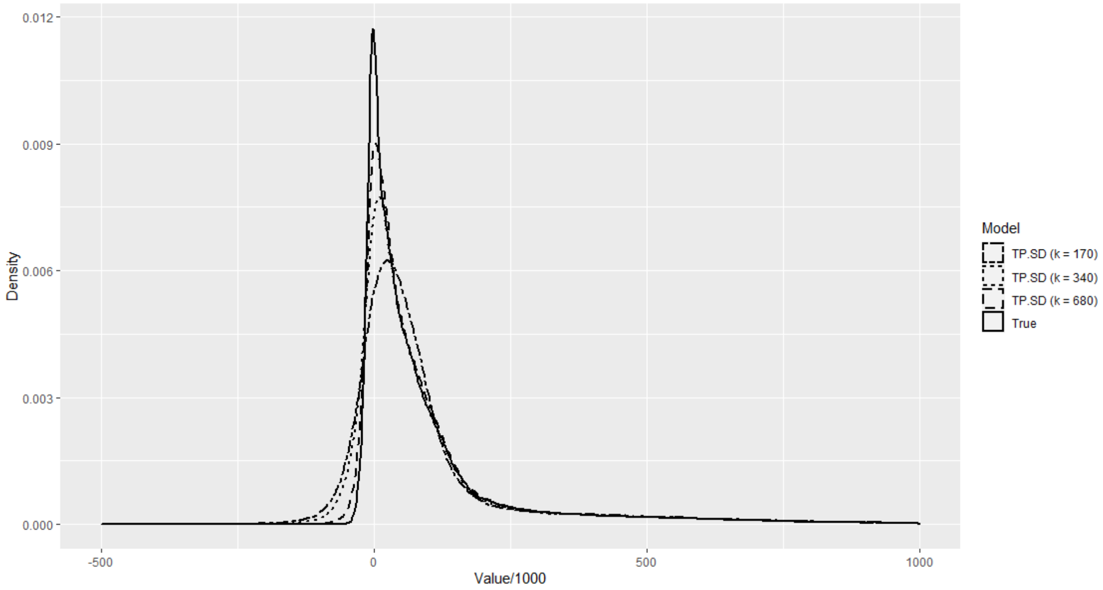

Figure 4.2: Distributions of the estimated values from the two-phase selection procedure and the true contract values.

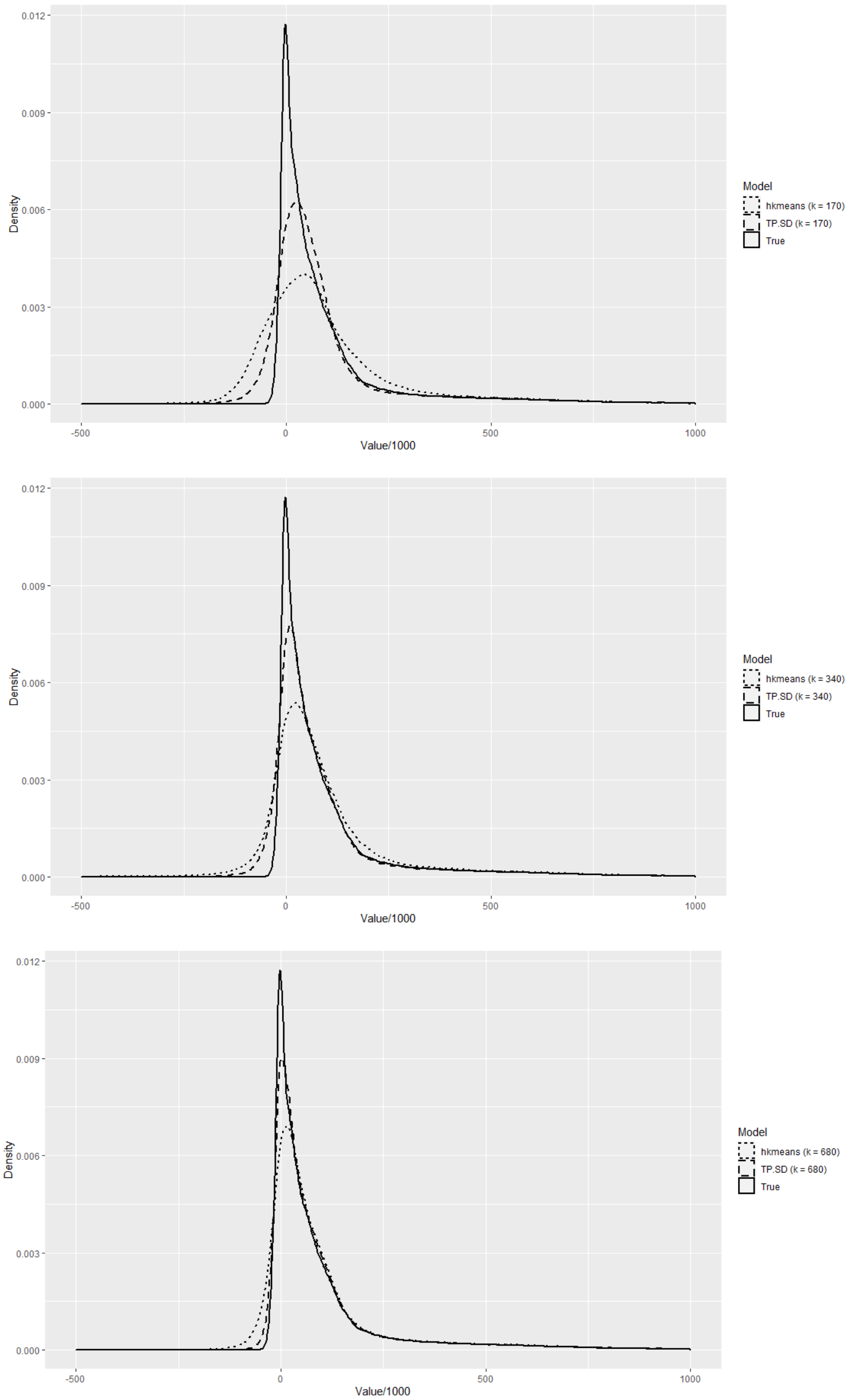

Figure 4.3: Distributions of the estimated values from the conventional procedure and the two-phase selection procedure and the true contract values. Each subfigure is for a particular k.

Chapter 5

Capture Ratio in Fund Management

5.1 Introduction

In this chapter, we investigate a performance measure for investment funds. Different from stocks, a fund consists of several underlying assets. A fund manager chooses the assets and their contributions to the fund. This diversity in investment provides better protection against the potential loss of an individual asset, to the investor. Because funds are more conservative investment tools than stocks, investors care about not only whether they yield positive returns, but also their relative performance compared to some benchmarks.

The capture ratio is commonly used to measure fund managers' performance against a market benchmark, for example, the S&P 500 index. It serves as an important criterion employed by Morningstar (Haslem (2014)) in selecting funds. Unlike other performance metrics such as the Sharpe ratio, Treynor ratio, and Sortino ratio, the upside and downside capture ratios aim to measure the relative performance of the fund when the benchmark experiences positive and negative returns, respectively. An (overall) capture ratio can be calculated as the difference or ratio between the upside and downside capture ratios.

Even though the capture ratio is popular in practice (Cox and Goff (2013)), we are not aware of any research regarding its statistical estimation. Investors extensively use the capture ratio to measure and compare fund performance. The use of a naive calculation/estimation of the capture ratio based on empirical data could be misleading if the statistical properties of the estimator are unknown. A positive or higher capture ratio estimate does not necessarily mean a statistically better underlying distribution. Indeed, the literature finds little evidence that the capture ratio predicts future performance, especially over a short time period (Coy and Robbins (2021); Gottesman and Morey (2021)). Hence, understanding the distribution of the estimated capture ratio is important for practical applications.

To our knowledge, Cox and Goff (2013) were the first researchers to define the capture ratio analytically. The bulk of the existing literature uses the capture ratio as a measure in comparing the performance of different funds (e.g. Chang and Krueger (2013); Chang et al. (2013); Chen (2014); Bello (2014); Kuhle and Lin (2018); Chang et al. (2019)). Some authors have employed the capture ratio in other financial models. For example, Jacobsen (2009) uses an arbitrage pricing model to analyze the “fair value” of a fund manager. At the fair value, the manager’s upside and downside capture ratio lie on an “efficient” line. Cline and Gilstrap (2021) apply a factor model to distinguish skilled fund managers with high future fund returns. They use the capture ratio as a factor that represents a manager’s skill.

This chapter aims to study the statistical properties of capture ratio estimates by deriving the asymptotic joint distribution of capture ratios of two funds. Based on the asymptotic distribution, we test hypotheses regarding the relative performance of two funds. The asymptotic distribution for a single capture ratio can be obtained as the marginal of the joint one. We also compare the asymptotic method with the non-parametric bootstrap method, which requires fewer distributional assumptions, in their performance of estimation and hypothesis testing. We find that the capture ratio estimates based on different return frequencies can significantly differ in magnitude. They can also be substantially biased when the underlying model is misspecified or the sample size is not large enough. For hypothesis tests based on the asymptotic distribution, our simulation study shows that a large sample size is required to achieve a 5% (which is the test size we set in simulation) empirical size of the test. Furthermore, only when the true capture ratio is far from the hypothesized value can one limit the empirical false acceptance rate to 20% at a reasonable sample size (e.g. 120 for monthly data or 250 for daily data), for particular models. In general, sample sizes of 12 and 60 for monthly data are not sufficient for statistically reliable estimation and hypothesis tests. However, these are used in practice and the research literature (see Cox and Goff (2013); Chang and Krueger (2013); Chen (2014); Haslem (2014); Kuhle and Lin (2018); Cline and Gilstrap (2021)).

The remainder of the chapter is structured as follows. Section 5.2 presents the mathematical results on the asymptotic distribution of the capture ratio estimates, including both the cases where returns are independently and identically distributed, and when there is a serial correlation among the returns. Section 5.3 presents numerical results, including a simulation study on finite sample properties of the estimators, and the application of capture ratio estimates to HFRX hedge fund index data. Section 5.4 presents conclusions and discusses directions for future work.

5.2 Asymptotic Distribution

In this section, we investigate the asymptotic joint distribution of two capture ratio estimates. We begin with analytical definitions of the upside, downside, and overall capture ratios. The asymptotic distributions are then obtained under two cases in which the underlying asset returns follow different data generating processes. The necessary statistical techniques are similar to those found in Lo (2002), where the asymptotic distribution of the Sharpe ratio is studied.

5.2.1 Analytical Definition

We assume the fund returns X_n and benchmark returns B_n, $n = 1, 2, \cdots, N$, have finite second-order moments. $X_n, n = 1, \ldots, N$, are identically distributed and B_n is also identically distributed with

$$p^U = \mathbb{P}(B_n > 0) \in (0,1) \text{ and } p^D = \mathbb{P}(B_n < 0) \in (0,1). \tag{5.1}$$

For technical convenience, we assume X_n and B_n are continuous variables. The definitions of capture ratios depend on the following indicator variables of "up markets" and "down markets":

$$U_n = \mathbb{1}_{\{B_n > 0\}} \text{ and } D_n = \mathbb{1}_{\{B_n < 0\}}, \tag{5.2}$$

where $\mathbb{1}_A$ is an indicator function, returning a value of 1 if the event A occurs and 0 otherwise. Accordingly, $U_n = 1$ indicates the benchmark has a positive return over the nth period, and $D_n = 1$ means the benchmark has a negative return over the nth period. The two variables are related by

$$U_n + D_n = 1 - \mathbb{1}_{\{B_n = 0\}},$$

where the second item on the right-hand side of the above equation can be ignored since we assume B_n is a continuous variable and thus the term is equal to zero with probability one.

Cox and Goff (2013) defines the upside capture ratio as

$$R^U_{X,N} := \frac{\left(\displaystyle\prod_{n=1}^{N}(1+X_nU_n)\right)^{\kappa/M^U_N} - 1}{\left(\displaystyle\prod_{n=1}^{N}(1+B_nU_n)\right)^{\kappa/M^U_N} - 1}, \tag{5.3}$$

and the downside capture ratio as

$$R^D_{X,N} := \frac{\left(\displaystyle\prod_{n=1}^{N}(1+X_nD_n)\right)^{\kappa/M^D_N} - 1}{\left(\displaystyle\prod_{n=1}^{N}(1+B_nD_n)\right)^{\kappa/M^D_N} - 1}, \tag{5.4}$$

where κ is the number of periods in a year, $M^U_N = \sum_{n=1}^N U_n$ is the number of periods in which the benchmark return is positive, and $M^D_N = \sum_{n=1}^N D_n$ is the number of periods in which the benchmark return is negative. Note that $M^U_N + M^D_N = N + \sum_{n=1}^N \mathbb{1}_{\{B_n=0\}}$, where $\sum_{n=1}^N \mathbb{1}_{\{B_n=0\}} = 0$ with probability one since we assume the benchmark returns B_n are continuous random variables.

Rigorously speaking, the above definition for the upside/downside capture ratio is problematic, because M^U_N and M^D_N could be zero. When $M^U_N > 0$, there exists at least one period, say j, such that $U_j = 1$ or equivalently $B_n > 0$, which yields $\prod_{n=1}^N(1 + B_nU_n) > 1$ and thus the denominator in (5.3) is strictly positive. So, $R^U_{X,N}$ is well defined if and only if $M^U_N > 0$ (or equivalently $M^D_N < N$ if we ignore a null probability set), which means there is at least one out of the N periods that has a positive return for the benchmark. Similarly, $R^D_{X,N}$ is well defined if and only if $M^D_N > 0$ (or equivalently $M^U_N < N$ if we ignore a null probability set), which means there is at least one out of the N periods that has a negative return for the benchmark. When the number of periods is reasonably large, the undefinedness issue with the above capture ratios rarely happens with stock return data because one can always observe at least one period with a positive return and another period with a negative return for any well-known benchmark (e.g., the S&P 500 index).

The overall capture ratio is defined as the difference between the upside capture ratio and the downside capture ratio:

$$R_{X,N} := R^U_{X,N} - R^D_{X,N}. \tag{5.5}$$

It is also called the capture spread in the literature, e.g., Cline and Gilstrap (2021). Furthermore, there is also literature that uses the ratio of the upside and downside ratios as the overall capture ratio e.g. Cox and Goff (2013) and Venugopal and Sophia (2020).

In this chapter, we are interested in the statistical uncertainty of the quantity $R_{X,N}$ as a performance metric for a fund. We will study its asymptotic distribution under some reasonable technical assumptions. For theoretical rigour, we extend the analytical definition of the capture ratio in (5.5) to cover the case where either M_N^U or M_N^D is zero and define

$$\tilde{R}_{X,N} := R_{X,N} \cdot \mathbb{1}_{S_N^C} + \xi \mathbb{1}_{S_N} = \left[R_{X,N}^U - R_{X,N}^D\right] \cdot \mathbb{1}_{S_N^C} + \xi \mathbb{1}_{S_N} \tag{5.6}$$

where ξ is a constant, S_N denotes the event $\{M_N^U = 0 \text{ or } M_N^D = 0\}$, and S_N^C is the complementary set of S_N, i.e., $S_N^C = \{\text{neither } M_N^U = 0 \text{ nor } M_N^D = 0\}$. The inclusion of the constant ξ in the definition is to make the capture ratio well-defined for the case when either M_N^U or M_N^D is zero. As we previously pointed out, when the number of periods is reasonably large, the event S_N will be extremely rare, and in probabilistic terminology, the event has a probability of zero to occur. Though the specification for $\tilde{R}_{X,N}$ in (5.6) depends on ξ, as we will see shortly, the asymptotic distribution of $R_{X,N}$ is independent of the choice of ξ.

To facilitate the discussion in the sequel, we introduce some additional notation:

$$\begin{aligned} \tilde{X}_n^U &= \log(1 + X_n U_n) \text{ and } \tilde{B}_n^U = \log(1 + B_n U_n); \\ \tilde{X}_n^D &= \log(1 + X_n D_n) \text{ and } \tilde{B}_n^D = \log(1 + B_n D_n). \end{aligned}$$

Then, on the event S_N^C,

$$\begin{aligned} \tilde{R}_{X,N} = R_{X,N} &= \frac{\exp\left\{\frac{\kappa}{M_N^U}\sum_{n=1}^{N}\log(1+X_nU_n)\right\}-1}{\exp\left\{\frac{\kappa}{M_N^U}\sum_{n=1}^{N}\log(1+B_nU_n)\right\}-1} - \frac{\exp\left\{\frac{\kappa}{M_N^D}\sum_{n=1}^{N}\log(1+X_nD_n)\right\}-1}{\exp\left\{\frac{\kappa}{M_N^D}\sum_{n=1}^{N}\log(1+B_nD_n)\right\}-1} \\ &= \frac{\exp\left\{\frac{\kappa N}{M_N^U}\frac{1}{N}\sum_{n=1}^{N}\tilde{X}_n^U\right\}-1}{\exp\left\{\frac{\kappa N}{M_N^U}\frac{1}{N}\sum_{n=1}^{N}\tilde{B}_n^U\right\}-1} - \frac{\exp\left\{\frac{\kappa N}{M_N^D}\frac{1}{N}\sum_{n=1}^{N}\tilde{X}_n^D\right\}-1}{\exp\left\{\frac{\kappa N}{M_N^D}\frac{1}{N}\sum_{n=1}^{N}\tilde{B}_n^D\right\}-1}. \end{aligned} \tag{5.7}$$

We will study hypothesis tests on the capture ratios of two funds based on an asymptotic inference framework. We suppose the second fund has returns Y_n, $n =$

$1, 2, \cdots, N$, and capture ratio $R_{Y,N}$, computed using the same benchmark as $R_{X,N}$. Further, we define $\tilde{R}_{Y,N}$ as the extended version of $R_{Y,N}$ in the same fashion as we defined $\tilde{R}_{X,N}$ in (5.6). Putting both capture ratios together, we consider the bivariate capture ratio:

$$\tilde{R}_N = (\tilde{R}_{X,N}, \tilde{R}_{Y,N})^{'}.$$

For the derivation of the asymptotic distribution of the above bivariate capture ratio, it is helpful to further define some additional notation. We let

$$Z_n = (\tilde{X}_n^U, \tilde{Y}_n^U, \tilde{B}_n^U, \tilde{X}_n^D, \tilde{Y}_n^D, \tilde{B}_n^D)^{'}, \;\; n = 1, 2, \cdots, N,$$

and put

$$\bar{Z}_N = \frac{1}{N}\sum_{n=1}^{N} Z_n.$$

Then, the bivariate capture ratio can be written as

$$\tilde{R}_N = \phi\left(C_N \bar{Z}_N\right) \cdot \mathbb{1}_{S_N^C} + \xi \mathbb{1}_{S_N} \tag{5.8}$$

with matrix

$$C_N = \text{diag}\left\{\frac{\kappa N}{M_N^U}, \frac{\kappa N}{M_N^U}, \frac{\kappa N}{M_N^U}, \frac{\kappa N}{M_N^D}, \frac{\kappa N}{M_N^D}, \frac{\kappa N}{M_N^D}\right\}.$$

and real-valued function

$$\phi((x_1, x_2, x_3, x_4, x_5, x_6)^{'}) = \left(\frac{e^{x_1}-1}{e^{x_3}-1} - \frac{e^{x_4}-1}{e^{x_6}-1},\; \frac{e^{x_2}-1}{e^{x_3}-1} - \frac{e^{x_5}-1}{e^{x_6}-1}\right)^{'}.$$

5.2.2 Independence Case

For the simplest case, we assume that the sequence of triplets $\{(X_n,\, Y_n, B_n), n = 1, \ldots, N\}$ are independent and identically distributed (i.i.d.). The Multivariate Central Limit Theorem implies the asymptotic distribution of the mean estimator $\bar{Z}_N$:

$$\sqrt{N}(\bar{Z}_N - \mu) \xrightarrow{d} N(\mathbf{0}, \Sigma),$$

where

$$\begin{aligned} \mu &= \mathbb{E}[Z_n] = \left(\mathbb{E}[\tilde{X}_n^U], \mathbb{E}[\tilde{Y}_n^U], \mathbb{E}[\tilde{B}_n^U], \mathbb{E}[\tilde{X}_n^D], \mathbb{E}[\tilde{Y}_n^D], \mathbb{E}[\tilde{B}_n^D]\right)^{'}, \\ \Sigma &= \mathbb{E}[(Z_1 - \mu)(Z_1 - \mu)^{'}] \end{aligned}$$

and "$\xrightarrow{d}$" means convergence in distribution. Furthermore, $\{(U_n, D_n), n = 1, \ldots, N\}$ are i.i.d. since both U_n and D_n are measurable functions of B_n; see (5.2). Hence, by the Weak Law of Large Numbers,

$$\frac{M_N^U}{N} \xrightarrow{P} p^U \quad \text{and} \quad \frac{M_N^D}{N} \xrightarrow{P} p^D,$$

where "$\xrightarrow{P}$" means convergence in probability. Then, according to Slutsky's Theorem, we have

$$\sqrt{N}\left(C_N \bar{Z}_N - C\mu\right) \xrightarrow{d} N\left(\mathbf{0}, C\Sigma C'\right),$$

where

$$C = \operatorname{diag}\left\{\frac{\kappa}{p^U}, \frac{\kappa}{p^U}, \frac{\kappa}{p^U}, \frac{\kappa}{p^D}, \frac{\kappa}{p^D}, \frac{\kappa}{p^D}\right\}.$$

For the derivation of the asymptotic distribution for the bivariate capture ratio $\tilde{R}_N$, we utilize its representation in (5.8), where the Jacobian matrix of ϕ is

$$\nabla\phi = \begin{pmatrix} \frac{e^{x_1}}{e^{x_3}-1} & 0 & -\frac{(e^{x_1}-1)e^{x_3}}{(e^{x_3}-1)^2} & \frac{-e^{x_4}}{e^{x_6}-1} & 0 & \frac{(e^{x_4}-1)e^{x_6}}{(e^{x_6}-1)^2} \\ 0 & \frac{e^{x_2}}{e^{x_3}-1} & -\frac{(e^{x_2}-1)e^{x_3}}{(e^{x_3}-1)^2} & 0 & \frac{-e^{x_5}}{e^{x_6}-1} & \frac{(e^{x_5}-1)e^{x_6}}{(e^{x_6}-1)^2} \end{pmatrix}'.$$

As proved in Appendix D.1, the asymptotic distribution of $\tilde{R}_N$ is given by:

$$\sqrt{N}(\tilde{R}_N - R) \xrightarrow{d} N\left(\mathbf{0}, \left(\nabla\phi\left(C\mu\right)\right)' C\Sigma C' \left(\nabla\phi\left(C\mu\right)\right)\right),$$

where $\mathbf{0} = (0, 0)'$, and

$$R = (R_X, R_Y)' := \begin{pmatrix} \frac{\exp\left\{\frac{\kappa}{p^U}\mathbb{E}[\tilde{X}_1^U]\right\}-1}{\exp\left\{\frac{\kappa}{p^U}\mathbb{E}[\tilde{B}_1^U]\right\}-1} - \frac{\exp\left\{\frac{\kappa}{p^D}\mathbb{E}[\tilde{X}_1^D]\right\}-1}{\exp\left\{\frac{\kappa}{p^D}\mathbb{E}[\tilde{B}_1^D]\right\}-1} \\ \frac{\exp\left\{\frac{\kappa}{p^U}\mathbb{E}[\tilde{Y}_1^U]\right\}-1}{\exp\left\{\frac{\kappa}{p^U}\mathbb{E}[\tilde{B}_1^U]\right\}-1} - \frac{\exp\left\{\frac{\kappa}{p^D}\mathbb{E}[\tilde{Y}_1^D]\right\}-1}{\exp\left\{\frac{\kappa}{p^D}\mathbb{E}[\tilde{B}_1^D]\right\}-1} \end{pmatrix}. \tag{5.9}$$

R can be viewed as the vector of theoretically true capture ratios for X_n and Y_n. By the assumptions in equation (5.1), we have $\mathbb{E}[\tilde{B}_1^U] > 0$ and $\mathbb{E}[\tilde{B}_1^D] > 0$, and thus, R is well defined.

5.2.3 Serially Correlated Returns

In real financial markets, asset returns are unlikely to be i.i.d. Here we consider the case where there is a serial correlation in the asset returns. Specifically, we assume

$\{(X_n, Y_n, B_n), n = 1, 2, \ldots\}$ are (strictly) stationary and ergodic. By definition,

$$\{(\tilde{X}_n^U, \tilde{Y}_n^U, \tilde{B}_n^U, \tilde{X}_n^D, \tilde{Y}_n^D, \tilde{B}_n^D), n = 1, 2, \ldots\}$$

are also stationary and ergodic since they are measurable functions of (X_n, Y_n, B_n). Hence, $\{Z_n, n = 1, 2 \ldots\}$ is stationary and ergodic.

In this case, the asymptotic distribution of $\bar{Z}_N$, or equivalently $\sqrt{N}(\bar{Z}_N - \mu)$, can be obtained via the generalized method of moments (GMM) estimator of Hansen (1982). Lo (2002) uses the GMM estimator to obtain the asymptotic distribution of the Sharpe ratio under the assumption of serially correlated returns. In our case, $\bar{Z}_N$ is the GMM estimator of μ. According to Theorem 3.1 in Hansen (1982), its asymptotic distribution is given by

$$\sqrt{N}(\bar{Z}_N - \mu) \xrightarrow{d} N\left(\mathbf{0}, H^{-1}\Sigma_g(H^{-1})'\right) = N\left(\mathbf{0}, \Sigma_g\right),$$

where

$$H = \lim_{N\to\infty} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \nabla_\beta f(Z_i, \mu)\right] = -\boldsymbol{I}_6,$$

In the above equation, $f(x, \beta) = x - \beta$ where x and β are six-dimensional column vectors. $\boldsymbol{I}_6$ is the six by six identity matrix and

$$\Sigma_g = \lim_{N\to\infty} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} f(Z_i, \mu)(f(Z_j, \mu))'\right].$$

Following a similar procedure to the independence case, the asymptotic distribution of R_N in the serially correlated case is

$$\sqrt{N}(R_N - R) \xrightarrow{d} N\left(\mathbf{0}, \left(\nabla\phi\left(C\mu\right)\right)' C\Sigma_g C' \left(\nabla\phi\left(C\mu\right)\right)\right). \tag{5.10}$$

The estimator of Σ_g can be obtained by the procedure of Newey and West (1987) (see Lo (2002)):

$$\begin{aligned} \hat{\Sigma}_g &= \hat{A}_0 + \sum_{j=1}^{m} a(j, m)(\hat{A}_j + \hat{A}_j'), \quad m \ll N, \\ \hat{A}_j &= \frac{1}{N}\sum_{n=j+1}^{N} f(Z_n, \bar{Z}_N)(f(Z_{n-j}, \bar{Z}_N))', \\ a(j, m) &= 1 - \frac{j}{m+1}. \end{aligned} \tag{5.11}$$

A rule of thumb for choosing an appropriate m is $m = \lceil 0.75 \times N^{1/3} \rceil$, where $\lceil x \rceil$ denotes the ceiling of a real number x.

5.2.4 Hypothesis Tests and Confidence Intervals

In this section, we present the analytical results for some hypothesis tests and confidence interval estimation based on the asymptotic distributions in the two cases studied above.

Univariate Test

The first type of hypothesis test is to test whether a capture ratio is positive (or zero):

$$H_0 : R_X = 0, \qquad H_a : R_X > 0 \text{ (or } R_X \neq 0).$$

Based on the asymptotic joint distributions, we have

$$\sqrt{N}(R_{X,N} - R_X) \xrightarrow{d} N(0, \sigma_X^2),$$

where

$$\sigma_X^2 = \begin{cases} (1,0)\left(\nabla\phi\left(C\mu\right)\right)' C_N \hat{\Sigma} C_N' \left(\nabla\phi\left(C\mu\right)\right)(1,0)', & \text{Independence case,} \\ (1,0)\left(\nabla\phi\left(C\mu\right)\right)' C_N \hat{\Sigma}_g C_N' \left(\nabla\phi\left(C\mu\right)\right)(1,0)', & \text{Serial correlation case.} \end{cases}$$

The test statistic is

$$T_X = \frac{R_{X,N}}{\hat{\sigma}_X/\sqrt{N}} \xrightarrow{d} N(0,1), \text{ as } N \to \infty, \text{ under } H_0,$$

where

$$\hat{\sigma}_X^2 = \begin{cases} (1,0)\left(\nabla\phi\left(C_N \bar{Z}_N\right)\right)' C\Sigma C' \left(\nabla\phi\left(C_N \bar{Z}_N\right)\right)(1,0)', & \text{Independence case,} \\ (1,0)\left(\nabla\phi\left(C_N \bar{Z}_N\right)\right)' C\Sigma_g C' \left(\nabla\phi\left(C_N \bar{Z}_N\right)\right)(1,0)', & \text{Serial correlation case.} \end{cases}$$

Then, the rejection interval at a significance level of 5% is given by:

$(R_{X,N} + 1.645\hat{\sigma}_X/\sqrt{N}, +\infty)$ when H_a is $R_X > 0$, and
$(-\infty, R_{X,N} - 1.96\hat{\sigma}_X/\sqrt{N}) \cup (R_{X,N} + 1.96\hat{\sigma}_X/\sqrt{N}, \infty)$ when H_a is $R_X \neq 0$.

A 95% confidence interval for R_X is given by:

$$\left(R_{X,N} - 1.96\hat{\sigma}_X/\sqrt{N},\ R_{X,N} + 1.96\hat{\sigma}_X/\sqrt{N}\right).$$

Bivariate Test

Suppose we are interested in comparing the capture ratios of two funds, e.g. R_X and R_Y. The null and alternative hypotheses are:

$$H_0 : R_X = R_Y, \qquad H_a : R_X > R_Y \text{ (or } R_X \neq R_Y).$$

According to the asymptotic joint distributions, we have the test statistic

$$T = \frac{R_{X,N} - R_{Y,N}}{\hat{\sigma}_T/\sqrt{N}} \xrightarrow{d} N(0,1), \text{ as } N \to \infty, \text{ under } H_0,$$

where

$$\hat{\sigma}_T^2 = \begin{cases} (1,-1)\left(\nabla\phi\left(C_N\bar{Z}_N\right)\right)' C_N\hat{\Sigma}C_N\left(\nabla\phi\left(C_N\bar{Z}_N\right)\right)(1,-1)', & \text{Independence case,} \\ (1,-1)\left(\nabla\phi\left(C_N\bar{Z}_N\right)\right)' C_N\hat{\Sigma}_gC_N\left(\nabla\phi\left(C_N\bar{Z}_N\right)\right)(1,-1)', & \text{Serial correlation case.} \end{cases}$$

The rejection interval at a significant level of 5% is given by:

$$\left((R_{X,N} - R_{Y,N}) + 1.645\hat{\sigma}_T/\sqrt{N}, +\infty\right)$$

when H_a is $R_X > 0$, and

$$\left(-\infty, R_{X,N} - R_{Y,N} - 1.96\hat{\sigma}_T/\sqrt{N}\right) \cup \left(R_{X,N} - R_{Y,N} + 1.96\hat{\sigma}_T/\sqrt{N}, \infty\right)$$

when H_a is $R_X \neq 0$.

A 95% confidence interval for $R_X - R_Y$ is:

$$\left(R_{X,N} - R_{Y,N} - 1.96\hat{\sigma}_T/\sqrt{N}, R_{X,N} - R_{Y,N} + 1.96\hat{\sigma}_T/\sqrt{N}\right).$$

5.3 Numerical Examples

In this section, we use numerical simulation and empirical data to study the performance of the estimation of the capture ratio. In addition to applying the asymptotic results of the previous section, we will also apply the bootstrap method for the estimation of the capture ratio, and conduct a comparative analysis of the results from both methods.

5.3.1 Simulation Study

For the simulation study, we fit two joint distributions for the returns of two hedge funds based on data from January 2011 to December 2020. There are in total 120 months (2517 days) over the period. The data we use is the daily and monthly return data of HFRX[1] hedge fund indexes: HFRXCA and HFRXEHG, and the S&P 500 value-weighted return including dividends. These two hedge funds are chosen because they have the highest and lowest 10-year capture ratios based on the monthly data, respectively (see Table 5.8). During the period, HFRXCA has capture ratios of 0.0450 and 0.0189, based on the daily and monthly data respectively. HFRX-EHG has capture ratios of -0.0395 and -0.3778, based on the daily and monthly data respectively. The S&P 500 index serves as the benchmark in calculating the capture ratio. We consider two capture ratios (R_X and R_Y, respectively, for HFRXCA and HFRXEHG), and the difference between them. For the independence returns assumption, we consider two data generating models: multivariate normal returns (MVN) and normal marginal distributions with a t-copula (MNT). For the serial correlation case, we apply the multivariate autoregressive model MAR(1). We fit the empirical data with each of the three models to get three data generating models used in the simulation as follows:

(1) MVN-Daily: $N\left(\begin{pmatrix}0.000154\\0.000036\\0.000495\end{pmatrix},\begin{pmatrix}0.000008 & 0.000002 & 0.000002\\0.000002 & 0.000027 & 0.000022\\0.000002 & 0.000022 & 0.000121\end{pmatrix}\right)$

(2) MVN-Monthly: $N\left(\begin{pmatrix}0.0030\\0.0010\\0.0099\end{pmatrix},\begin{pmatrix}0.0002 & 0.0003 & 0.0003\\0.0003 & 0.0010 & 0.0009\\0.0003 & 0.0009 & 0.0015\end{pmatrix}\right)$

(3) MNT-Daily: Normal marginal distributions are the same as in MVN-Daily model. Degree of freedom is 5.2770. The correlation structure is

$$\begin{pmatrix}1 & 0.0821 & -0.0499\\0.0821 & 1 & 0.4984\\-0.0499 & 0.4984 & 1\end{pmatrix}.$$

[1] The HFRX database is one of the public databases provided by https://www.hfr.com. Compared to other HFR databases, the HFRX database provides more data over a longer time period. The hedge fund data is collected from the HFR website. The S&P 500 data is from the WRDS database.

(4) MNT-Monthly: Normal marginal distributions are the same as in the MVN-Monthly model. The degree of freedom is 3.8555. The correlation structure is

$$\begin{pmatrix} 1 & 0.5668 & 0.5866 \\ 0.5668 & 1 & 0.6416 \\ 0.5866 & 0.6416 & 1 \end{pmatrix}.$$

(5) MAR-Daily:

$$X_t - \begin{pmatrix} 0.000154 \\ 0.000036 \\ 0.000495 \end{pmatrix} = \begin{pmatrix} -0.1432 & 0.3340 & 0.9297 \\ 0.0856 & 0.1640 & 0.7413 \\ -0.0074 & -0.0077 & -0.3210 \end{pmatrix} \left(X_{t-1} - \begin{pmatrix} 0.000154 \\ 0.000036 \\ 0.000495 \end{pmatrix} \right) + \epsilon_t$$

$$\text{where } \epsilon_t \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.000008 & 0.000002 & 0.000002 \\ 0.000002 & 0.000026 & 0.000017 \\ 0.000002 & 0.000017 & 0.000096 \end{pmatrix} \right).$$

(6) MAR-Monthly:

$$X_t - \begin{pmatrix} 0.0030 \\ 0.0010 \\ 0.0099 \end{pmatrix} = \begin{pmatrix} 0.3845 & 0.1175 & 0.1957 \\ 0.0133 & 0.3042 & 0.4885 \\ -0.0911 & -0.3183 & 0.4329 \end{pmatrix} \left(X_{t-1} - \begin{pmatrix} 0.0030 \\ 0.0010 \\ 0.0099 \end{pmatrix} \right) + \epsilon_t$$

$$\text{where } \epsilon_t \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.0002 & 0.0003 & 0.0003 \\ 0.0003 & 0.0001 & 0.0008 \\ 0.0003 & 0.0008 & 0.0014 \end{pmatrix} \right).$$

The following table reports the theoretically "true"[2] capture ratio for each model and frequency. For daily and monthly frequencies, we use $\kappa = 250$ and $\kappa = 12$, respectively.

	MVN		MNT		MAR	
	R_X	R_Y	R_X	R_Y	R_X	R_Y
Daily	0.0060	-0.3205	0.0748	-0.3761	1.1852	0.3133
Monthly	0.0106	-0.4028	0.0059	-0.3530	0.4489	0.2100

Table 5.1: "True" capture ratios for each model and frequency

[2] The theoretically true capture ratio is difficult to calculate since the formula involves $\mathbb{E}[\tilde{X}_n^U]$, $\mathbb{E}[\tilde{X}_n^D]$, $\mathbb{E}[\tilde{B}_n^U]$ and $\mathbb{E}[\tilde{B}_n^D]$. We estimate the true ratio based on 10^7 simulations. That is, based on the estimated parameters from empirical data, we simulate 10^7 pairs (X_n, B_n). Then we use all the data points to estimate $\mathbb{E}[\tilde{X}_n^U]$, $\mathbb{E}[\tilde{X}_n^D]$, $\mathbb{E}[\tilde{B}_n^U]$ and $\mathbb{E}[\tilde{B}_n^D]$.

From the above table, it is easy to tell that the capture ratio can be significantly different in magnitude under different models and different frequencies. Hence, applying an appropriate model is important in practice. However, these issues are beyond the scope of this chapter. We are more interested in the differences between the estimated (asymptotic or bootstrapped) results and the "true" ones.

Estimation of the Capture Ratios

For each model (i.e. MVN, MNT, and MAR) and data frequency (i.e. daily and monthly), we simulate N (see Tables 5.2 and 5.3 for the specific values of N) data points and then use formula (5.7) to compute the capture ratios, $R_{X,N}$ and $R_{Y,N}$, and their difference. We independently replicate the simulation 500 times and compute the mean squared error (MSE) and the mean relative error (MRE) of the capture ratio as well as the capture ratio difference estimates based on the 500 simulations. The results are reported in Tables 5.2 and 5.3, respectively for MSE and MRE. The "Empirical" estimates are computed using formula (5.7) based on N simulated data points $\{X_1, \cdots, X_N\}$ for each model and data frequency. We also adopt bias-correction methods: bootstrap bias-corrected estimator and jackknife estimator (Efron and Tibshirani (1994)). Let $\hat{R}_N$ denote the empirical estimate based on the N simulated data points. The bootstrap bias-corrected estimator is computed as $2\hat{R}_N - \hat{R}^*_N$, where $\hat{R}^*_N$ is the bootstrapped estimate with a bootstrap sample size 2,000. The jackknife estimate is computed as

$$N\hat{R}_N - (N-1)\hat{R}_{N,jack} = N\hat{R}_N - \frac{N-1}{N}\sum_{i=1}^{N}\hat{R}_{N-1,i}$$

where $\hat{R}_{N,jack} = \frac{1}{N}\sum\limits_{i=1}^{N}\hat{R}_{N-1,i}$ and $\hat{R}_{N-1,i}$ is the estimate based on the sample excluding the i-th data point, i.e., $\{X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_N\}$.

From the results, the error decreases when the sample size N increases. But estimates may not be reliable when the model is not appropriate, e.g. MVN model in terms of MRE. The bias-correction methods improve the estimates in most cases. However, the small sample size is an issue in both the empirical estimation and the estimation with bias correction (Steck and Jaakkola (2003)). Hence, in practice, to have a less biased estimation, one needs a large sample size, e.g. larger than or equal to 120, even when adopting a bias-corrected method. The one-year capture ratio based on monthly returns is provided by Morningstar© [3] and also used in the literature (see Cox and Goff (2013); Cline and Gilstrap (2021)). However, our simulation results in Tables 5.2 and 5.3 indicate that an estimation calculated from 12 monthly returns could be misleading.

Type I and Type II Error

We also examine the type I and type II errors of the hypothesis tests using simulation studies. For a hypothesis test, the type I error is the probability that we incorrectly reject the true null hypothesis. The type II error, on the other hand, is the probability that we incorrectly accept the false null hypothesis.

Suppose we want to examine the univariate hypothesis test that one capture ratio is positive, at a significance level of 5%. We consider the null hypothesis $R_X = 0$ and alternative hypothesis $R_X > 0$. For each of the three data generating models (i.e., MVN, MNT, and MAR) and different data frequencies (i.e., daily and monthly), we independently generate 2,000 simulated capture ratios and compute the type I error as the portion of the obtained capture ratio values that are greater than the 95% quantile based on the asymptotic distribution, i.e., $1.645\sigma_X/\sqrt{N}$, where σ_X is estimated as explained in section 5.2.4. The type II error is computed as the portion of simulated capture ratios less than the 95% quantile. In addition to the tests based on the asymptotic normality, we also assess the performance of hypothesis testing using a bootstrap method (Efron and Tibshirani (1994)) for data generated from the MVN model. The bootstrap sample size is 2,000. The resample size for calculating the capture ratio is N. According to Cohen (2013), an efficient hypothesis test should achieve a type I error of no more than 5% and a type II error of no more than 20%.

In our simulations, the model parameters are calibrated from the same set of real return data as we explained at the beginning of the section, but we tune the average returns to get different theoretical capture ratios as indicated in Tables 5.4 - 5.7, which correspond to true or false null hypotheses. We use different sample sizes (as indicated in Tables 5.4 - 5.7) to calculate the sample capture ratios. We set $\kappa = 12$ and $\kappa = 250$ for monthly data frequency and daily frequency, respectively.

Table 5.4 summarizes the type I and type II errors of the univariate test. For each model, the null hypothesis is that R_X is zero. When the true value is zero (first row in each model), the table reports the type I error of the test. When the true

[3] See https://www.morningstar.ca/ca/news/185421/what-are-upside-and-downside-capture-ratios.aspx, and Kuhle and Lin (2018).

Frequency	Period	Model	Empirical			Bootstrap Bias-Corrected			Jackknife		
			R_X	R_Y	$R_X - R_Y$	R_X	R_Y	$R_X - R_Y$	R_X	R_Y	$R_X - R_Y$
Daily ($\kappa = 250$)	5-Year ($N = 1250$)	MVN	0.0010	0.0015	0.0022	0.0010	0.0015	0.0022	0.0010	0.0015	0.0022
		MNT	0.0012	0.0012	0.0021	0.0012	0.0012	0.0021	0.0012	0.0012	0.0021
		MAR	0.0695	0.0240	0.0236	0.0684	0.0238	0.0231	0.0681	0.0238	0.0230
	3-Year ($N = 750$)	MVN	0.0017	0.0023	0.0035	0.0017	0.0023	0.0035	0.0017	0.0023	0.0035
		MNT	0.0020	0.0018	0.0033	0.0020	0.0018	0.0033	0.0020	0.0018	0.0033
		MAR	0.107	0.0395	0.0339	0.1040	0.0386	0.0330	0.1042	0.0387	0.0330
	2-Year ($N = 500$)	MVN	0.0025	0.0036	0.0051	0.0025	0.0035	0.0050	0.0025	0.0035	0.0050
		MNT	0.0030	0.0025	0.0048	0.0030	0.0024	0.0048	0.0030	0.0024	0.0048
		MAR	0.2101	0.0706	0.0692	0.2013	0.0684	0.0662	0.2007	0.0681	0.0661
	1-Year ($N = 250$)	MVN	0.0054	0.0074	0.0108	0.0053	0.0073	0.0106	0.0053	0.0073	0.0106
		MNT	0.0060	0.0064	0.0107	0.0059	0.0063	0.0106	0.0059	0.0063	0.0106
		MAR	0.3799	0.1427	0.1190	0.3493	0.1322	0.1106	0.3499	0.1325	0.1106
	6-Month ($N = 125$)	MVN	0.0119	0.0156	0.0234	0.0117	0.0148	0.0227	0.0116	0.0148	0.0227
		MNT	0.0121	0.0119	0.0226	0.0119	0.0113	0.0220	0.0119	0.0113	0.0220
		MAR	0.8659	0.3101	0.2930	0.7265	0.2677	0.2493	0.7279	0.2678	0.2501
Monthly ($\kappa = 12$)	30-Year ($N = 360$)	MVN	0.0020	0.0049	0.0043	0.0019	0.0049	0.0042	0.0019	0.0049	0.0042
		MNT	0.0021	0.0083	0.0079	0.0021	0.0083	0.0079	0.0021	0.0082	0.0079
		MAR	0.0129	0.0251	0.0079	0.0129	0.0251	0.0079	0.0129	0.0250	0.0078
	20-Year ($N = 240$)	MVN	0.0032	0.0080	0.0066	0.0032	0.0079	0.0065	0.0032	0.0079	0.0065
		MNT	0.0031	0.0114	0.0107	0.0031	0.0113	0.0106	0.0031	0.0113	0.0105
		MAR	0.0173	0.0378	0.0126	0.0171	0.0374	0.0125	0.0170	0.0373	0.0124
	10-Year ($N = 120$)	MVN	0.0068	0.0177	0.0148	0.0066	0.0173	0.0145	0.0066	0.0173	0.0145
		MNT	0.0067	0.0237	0.0219	0.0065	0.0232	0.0214	0.0065	0.0231	0.0214
		MAR	0.0365	0.0801	0.0251	0.0356	0.0778	0.0245	0.0356	0.0778	0.0245
	5-Year ($N = 60$)	MVN	0.0135	0.0366	0.0302	0.0130	0.0350	0.0287	0.0129	0.0348	0.0285
		MNT	0.0126	0.0488	0.0411	0.0122	0.0468	0.0394	0.0121	0.0468	0.0394
		MAR	0.0734	0.1626	0.0487	0.0708	0.1564	0.0468	0.0705	0.1557	0.0466
	1-Year ($N = 12$)	MVN	0.2856	0.579	0.4414	1.3295	1.8127	1.7298	0.4354	2.1485	1.8481
		MNT	0.0849	0.3542	0.3267	0.7207	1.4932	2.4857	7.3524	3.1053	14.7346
		MAR	0.4432	1.0808	0.3784	0.5644	1.6464	0.9915	0.2945	0.7528	0.3031

Table 5.2: Estimation MSEs in different models

Frequency	Period	Model	Empirical			Bootstrap Bias-Corrected			Jackknife		
			R_X	R_Y	$R_X - R_Y$	R_X	R_Y	$R_X - R_Y$	R_X	R_Y	$R_X - R_Y$
Daily ($\kappa = 250$)	5-Year ($N = 1250$)	MVN	2.2302	2.5321	3.2424	2.2261	2.5386	3.2441	2.2262	2.5302	3.2406
		MNT	0.2232	0.2103	0.2822	0.2234	0.2102	0.2820	0.2231	0.2102	0.2818
		MAR	0.4394	0.2574	0.2483	0.4362	0.2566	0.2466	0.4354	0.2563	0.2462
	3-Year ($N = 750$)	MVN	2.7226	3.5102	4.0539	2.7136	3.4955	4.0376	2.7185	3.4927	4.0424
		MNT	0.2866	0.2656	0.3625	0.2857	0.2636	0.3624	0.2857	0.2638	0.3617
		MAR	0.5238	0.3202	0.2935	0.5202	0.3181	0.2922	0.5214	0.3190	0.2923
	2-Year ($N = 500$)	MVN	3.5240	3.8454	4.6560	3.5188	3.8318	4.6383	3.5219	3.8424	4.6502
		MNT	0.3523	0.3190	0.4444	0.3518	0.3185	0.4434	0.3513	0.3181	0.4434
		MAR	0.6975	0.3927	0.4051	0.6875	0.3895	0.4010	0.6871	0.3881	0.4003
	1-Year ($N = 250$)	MVN	5.0799	6.0224	7.5185	5.0474	6.0047	7.4784	5.0548	6.0037	7.4737
		MNT	0.4775	0.4842	0.6462	0.4768	0.4806	0.6430	0.4757	0.4810	0.6429
		MAR	0.9490	0.5640	0.5320	0.9190	0.5510	0.5199	0.9181	0.5514	0.5194
	6-Month ($N = 125$)	MVN	7.2274	7.9914	9.7402	7.2391	7.8527	9.6253	7.1938	7.8412	9.6104
		MNT	0.7401	0.6803	0.9608	0.7371	0.6751	0.9517	0.7350	0.6753	0.9506
		MAR	1.3931	0.8294	0.8168	1.2957	0.7837	0.7714	1.2974	0.7851	0.7714
Monthly ($\kappa = 12$)	30-Year ($N = 360$)	MVN	1.7257	2.7247	2.4088	1.7165	2.7131	2.4019	1.7206	2.7132	2.3990
		MNT	2.9076	6.4924	6.1563	2.9027	6.4842	6.1409	2.8985	6.4724	6.1355
		MAR	0.3109	0.4392	0.2462	0.3103	0.4389	0.2462	0.3102	0.4388	0.2457
	20-Year ($N = 240$)	MVN	2.1265	3.3107	3.0285	2.1084	3.2814	3.0067	2.1136	3.2897	3.0101
		MNT	3.5907	7.3112	6.9689	3.5723	7.2732	6.9305	3.5732	7.2749	6.9344
		MAR	0.3729	0.5318	0.3158	0.3706	0.5290	0.3142	0.3704	0.5277	0.3136
	10-Year ($N = 120$)	MVN	3.1036	5.0764	4.6154	3.0626	5.0192	4.5644	3.0675	5.0189	4.5792
		MNT	5.6363	10.7627	10.207	5.5999	10.6491	10.1077	5.5916	10.6394	10.0971
		MAR	0.5259	0.7791	0.4455	0.5198	0.7720	0.4397	0.5201	0.7720	0.4403
	5-Year ($N = 60$)	MVN	4.6325	7.3630	6.7292	4.5388	7.1918	6.5972	4.5351	7.1795	6.5706
		MNT	7.5908	14.617	13.1417	7.4939	14.3424	12.9257	7.4799	14.3306	12.9282
		MAR	0.7383	1.1370	0.6252	0.7266	1.1205	0.6145	0.7254	1.1178	0.6130
	1-Year ($N = 12$)	MVN	15.4658	24.9436	21.4583	17.8421	31.3252	26.847	15.8484	29.3383	26.1269
		MNT	18.4263	37.5744	35.9127	27.1296	47.7336	56.3698	46.4481	56.4678	81.4573
		MAR	1.7760	2.8056	1.6572	1.6098	2.5371	1.6660	1.4939	2.3734	1.4726

Table 5.3: Estimation MREs in different models

value is greater than zero (other rows in each model), the values are type II errors. From the results, to keep the type I error less than 5%, a sample size greater than or equal to 120 is required, except for the MAR model.[4] We have the same conclusion for the type II error when the true R_X is close to zero. However, as the true R_X deviates more from zero, fewer samples are required to achieve power (i.e., one minus the type II error) of 80% or higher. However, this still excludes the one-year scenario where only 12 data points are used.

Table 5.5 reports the results of the bivariate test. All models achieve a type I error close to the target of 5% when the sample size is sufficiently large. For the type II error, it is still large when the true value is close to zero or the sample size is not large enough.

For the daily case, Tables 5.6 and 5.7 summarize the type I and type II errors for different models and sample sizes. A similar pattern of how the type I and type II errors respond to the sample size and deviation of the true capture ratio value from the hypothetical one in testing can be observed as we do from those simulations based on monthly data frequency.

These simulation results indicate that the hypothesis testing outcomes from both the asymptotic limiting distribution based method and the bootstrap based method can be misleading. When the true value is close to the value from the null hypothesis, a large sample is required for a reliable statistical inference conclusion. The needed sample size could be unattainable in practice. For $R_X = 0$ under the monthly MVN model, the corresponding asymptotic variance is around 0.7. Indeed, for a normal distribution $N(0, 0.7)$, when the true mean is 0.05, one needs more than 1,700 data points to obtain power (i.e., one minus the type II error) of 80% for testing at the 5% significance level. Furthermore, even when the true value deviates from zero, a commonly used 1-year monthly dataset is still not enough to reach a reliable conclusion via a hypothesis test.

5.3.2 Empirical Illustration

In this section, we calculate the empirical capture ratios of HFRX hedge fund indices. Throughout the section, there are thirty HFRX hedge fund indexes, which are the only HFRX indexes that have full daily and monthly data during the test period from January 2011 to December 2020. The benchmark asset is the S&P 500 index.

[4] Indeed, the type I error of the MAR model only reduces to 5% when the sample size is extremely large, more than 60,000.

Model	True R_X	$N = 12$	$N = 60$	$N = 120$	$N = 240$	$N = 360$
MVN	0	0.1040	0.0655	0.0570	0.0595	0.0425
	0.05	0.8628	0.8605	0.8190	0.7720	0.7005
	0.1	0.8226	0.7475	0.6225	0.4240	0.2905
	0.2	0.7382	0.4120	0.1790	0.0295	0.0025
	0.3	0.6156	0.1530	0.0180	0.0000	0.0000
	0.4	0.4784	0.0330	0.0000	0.0000	0.0000
MNT	0	0.1063	0.0770	0.0585	0.0545	0.0540
	0.05	0.8591	0.8510	0.8225	0.7590	0.6915
	0.1	0.8195	0.7400	0.6055	0.4195	0.2615
	0.2	0.7213	0.4000	0.1650	0.0190	0.0030
	0.3	0.5990	0.1490	0.0160	0.0000	0.0000
	0.4	0.4757	0.0290	0.0000	0.0000	0.0000
MAR	0	0.1520	0.1270	0.1250	0.1210	0.1150
	0.05	0.8180	0.8210	0.7910	0.7570	0.7370
	0.1	0.8020	0.7645	0.6875	0.6010	0.5205
	0.2	0.7510	0.5920	0.4435	0.2525	0.1440
	0.3	0.6935	0.4055	0.2145	0.0640	0.0145
	0.4	0.6145	0.2370	0.0830	0.0105	0.0005
Boot-strapped MVN	0	0.1076	0.0730	0.0595	0.0625	0.0570
	0.05	0.8624	0.8535	0.8225	0.7485	0.6750
	0.1	0.8249	0.7460	0.6345	0.4175	0.2605
	0.2	0.7359	0.4250	0.1755	0.0265	0.0020
	0.3	0.6088	0.1535	0.0150	0.0000	0.0000
	0.4	0.4642	0.0340	0.0000	0.0000	0.0000

Table 5.4: Type I and type II errors of the univariate test in the monthly case: The first row in each panel displays type I errors and the remaining rows show type II errors under different levels of true capture ratio.

Model	True $R_X - R_Y$	$N = 12$	$N = 60$	$N = 120$	$N = 240$	$N = 360$
MVN	0	0.0692	0.0500	0.0440	0.0425	0.0495
	0.05	0.9082	0.9075	0.8885	0.8525	0.8265
	0.1	0.8867	0.8450	0.7755	0.6490	0.5390
	0.2	0.8270	0.6510	0.4460	0.1860	0.0750
	0.3	0.7497	0.4170	0.1690	0.0205	0.0030
	0.4	0.6670	0.2080	0.0330	0.0005	0.0000
MNT	0	0.0691	0.0520	0.0460	0.0420	0.0455
	0.05	0.9194	0.9165	0.9050	0.8755	0.8570
	0.1	0.8988	0.8765	0.8160	0.7445	0.6690
	0.2	0.8673	0.7470	0.5820	0.3590	0.2035
	0.3	0.8037	0.5570	0.2945	0.0920	0.0215
	0.4	0.7371	0.3605	0.1065	0.0080	0.0005
MAR	0	0.0425	0.0325	0.0380	0.0450	0.0405
	0.05	0.9460	0.9435	0.9210	0.8935	0.8825
	0.1	0.9280	0.9085	0.8615	0.7955	0.7230
	0.2	0.8950	0.7970	0.6585	0.4650	0.2965
	0.3	0.8525	0.6305	0.3950	0.1605	0.0475
	0.4	0.8000	0.4460	0.1820	0.0260	0.0035
Boot -strapped MVN	0	0.0797	0.0530	0.0590	0.0480	0.0450
	0.05	0.9042	0.9015	0.8895	0.8395	0.8030
	0.1	0.8721	0.8375	0.7685	0.6460	0.5400
	0.2	0.8044	0.6370	0.4355	0.1930	0.0755
	0.3	0.7337	0.3920	0.1555	0.0160	0.0015
	0.4	0.6379	0.1920	0.0270	0.0015	0.0000

Table 5.5: Type I and type II errors of the bivariate test in the monthly case: The first row in each panel displays type I errors and the remaining rows show type II errors under different levels of true capture ratio.

Model	True R_X	$N = 21$	$N = 63$	$N = 125$	$N = 250$	$N = 500$
MVN	0	0.0915	0.0715	0.0640	0.0495	0.0455
	0.05	0.8765	0.8650	0.8600	0.8385	0.7755
	0.1	0.8445	0.7905	0.7385	0.6350	0.4500
	0.2	0.7070	0.5495	0.3465	0.1275	0.0125
	0.3	0.5925	0.3360	0.1220	0.0095	0.0000
	0.4	0.4765	0.1710	0.0295	0.0005	0.0000
MNT	0	0.0905	0.0650	0.0510	0.0600	0.0440
	0.05	0.8775	0.8755	0.8785	0.8350	0.7515
	0.1	0.8355	0.7915	0.7555	0.6265	0.4370
	0.2	0.7520	0.5830	0.4440	0.1855	0.0270
	0.3	0.6105	0.3280	0.1225	0.0100	0.0000
	0.4	0.4840	0.1790	0.0350	0.0000	0.0000
MAR	0	0.1515	0.1165	0.0970	0.0755	0.0650
	0.05	0.8390	0.8675	0.8895	0.8990	0.8950
	0.1	0.8265	0.8490	0.8645	0.8680	0.8485
	0.2	0.8010	0.8160	0.8085	0.7815	0.7090
	0.3	0.7805	0.7775	0.7575	0.6900	0.5700
	0.4	0.7540	0.7300	0.6870	0.5635	0.3890
Boot -strapped MVN	0	0.0985	0.0785	0.0650	0.0645	0.0530
	0.05	0.8660	0.8690	0.8565	0.8155	0.7550
	0.1	0.8250	0.7975	0.7480	0.6160	0.4225
	0.2	0.7020	0.5505	0.3445	0.1150	0.0125
	0.3	0.5885	0.3270	0.1145	0.0120	0.0000
	0.4	0.4640	0.1650	0.0255	0.0000	0.0000

Table 5.6: Type I and type II errors of the univariate test in the daily case: The first row in each panel displays type I errors and the remaining rows show type II errors under different levels of true capture ratio.

Model	True $R_X - R_Y$	$N = 21$	$N = 63$	$N = 125$	$N = 250$	$N = 500$
MVN	0	0.0375	0.0325	0.0365	0.0375	0.0500
	0.05	0.9450	0.9330	0.9100	0.8760	0.8195
	0.1	0.9250	0.8825	0.8365	0.7285	0.5665
	0.2	0.8630	0.7330	0.5825	0.3520	0.1180
	0.3	0.7760	0.5540	0.3240	0.1035	0.0095
	0.4	0.6650	0.3530	0.1345	0.0140	0.0010
MNT	0	0.0365	0.0270	0.0375	0.0375	0.0315
	0.05	0.9400	0.9340	0.8985	0.8740	0.8085
	0.1	0.9090	0.8705	0.8015	0.7185	0.5245
	0.2	0.8315	0.6650	0.4855	0.2510	0.0495
	0.3	0.7395	0.4745	0.2610	0.0625	0.0020
	0.4	0.6400	0.2915	0.0895	0.0050	0.0000
MAR	0	0.1275	0.0985	0.0750	0.0545	0.0505
	0.05	0.8585	0.8820	0.8915	0.9080	0.8930
	0.1	0.8455	0.8585	0.8650	0.8570	0.8055
	0.2	0.8140	0.8045	0.7885	0.7300	0.5790
	0.3	0.7770	0.7355	0.6740	0.5265	0.2880
	0.4	0.7505	0.6635	0.5630	0.3655	0.1140
Boot-strapped MVN	0	0.0395	0.0415	0.0455	0.0405	0.0485
	0.05	0.9425	0.9180	0.9025	0.8755	0.8135
	0.1	0.9195	0.8710	0.8160	0.7250	0.5670
	0.2	0.8520	0.7335	0.5625	0.3455	0.1315
	0.3	0.7585	0.5370	0.3145	0.1110	0.0090
	0.4	0.6550	0.3455	0.1255	0.0145	0.0000

Table 5.7: Type I and type II errors of the bivariate test in the daily case: The first row in each panel displays type I errors and the remaining rows show type II errors under different levels of true capture ratio.

10-Year Capture Ratios

Table 5.8 summarizes the capture ratios of the thirty indexes, calculated based on the monthly data for the period January 2011 to December 2020. The rows are ordered decreasingly in terms of the 10-year estimated capture ratio (the first column). The second and third columns are the 95% confidence intervals of the estimation, under the independence and serial correlation assumptions, respectively. The fourth column is the bootstrapped 95% confidence interval.

The results suggest that only two indexes (HFRXCA and HFRXMA) have positive 10-year capture ratios.[5] Even in these cases, the confidence intervals indicate that their deviations from zero are not statistically significant. For some funds with negative capture ratios, their confidence intervals exclude zero. However, in view of the simulation results,[6] the statistical power may not be significant when the estimate is not far from zero. Comparing the confidence intervals under the independence and serial correlation cases, both lead to the same conclusion when the test's statistical power is large, though values of the lower and upper bounds are different.

Table 5.9 summarizes the 10-year capture ratios calculated from the daily data. All 95% confidence intervals exclude zero. However, the exclusion may also lack statistical power when the estimate is close to zero.

Comparing Tables 5.8 and 5.9, we have the following observations. First, different data frequencies yield different rankings for funds when we use the capture ratio to measure the performance of funds. Second, only when the sample size is large and the estimated capture ratio is far from zero could we reject that the capture ratio is not zero (or non-negative) with statistical significance. Third, the confidence intervals of the bootstrap method are similar to those of the asymptotic method when the sample size is large.

Rolling Window Estimates

We now calculate the capture ratios based on a rolling window. For each fund, we calculate the 1-year and 5-year rolling capture ratios based on the monthly and daily data. The results are shown in Tables 5.10-5.13. In each column, the bold value is

[5] It should be noted that many hedge fund strategies do not aim for relative performance against a benchmark e.g. the HFRXAR (HFRX Absolute Return Index) and HFRXEHG (HFRX EH: Fundamental Growth Index). We select all available hedge fund indices for a complete analysis.

[6] Even though the simulation tests have different designs, similar results can be obtained for the two-sided tests.

Ticker	R_X	CI.ind	CI.sc	CI.boots
HFRXCA	0.0189	(-0.1125, 0.1503)	(-0.1293, 0.1670)	(-0.1026, 0.1509)
HFRXMA	0.0132	(-0.1351, 0.1614)	(-0.1365, 0.1628)	(-0.1301, 0.1510)
HFRXAR	-0.0256	(-0.1256, 0.0744)	(-0.1190, 0.0678)	(-0.1256, 0.0649)
HFRXSDV	-0.0830	(-0.3847, 0.2186)	(-0.3453, 0.1792)	(-0.3857, 0.2145)
HFRXRVMS	-0.0961	(-0.2172, 0.0250)	(-0.2102, 0.0180)	(-0.2257, 0.0270)
HFRXM	-0.1103	(-0.2789, 0.0583)	(-0.2674, 0.0469)	(-0.2912, 0.0682)
HFRXEMN	-0.1207	(-0.2698, 0.0283)	(-0.2897, 0.0482)	(-0.2607, 0.0355)
HFRXRVA	-0.1253	(-0.2437, -0.0069)	(-0.2474, -0.0033)	(-0.2383, -0.0082)
HFRXEW	-0.1351	(-0.2267, -0.0435)	(-0.2367, -0.0335)	(-0.2255, -0.0432)
HFRXMREG	-0.1510	(-0.3122, 0.0102)	(-0.3588, 0.0568)	(-0.3208, 0.0187)
HFRXED	-0.1539	(-0.3081, 0.0004)	(-0.3491, 0.0413)	(-0.3130, 0.0053)
HFRXSS	-0.1594	(-0.3391, 0.0204)	(-0.3714, 0.0526)	(-0.3449, 0.0239)
HFRXME	-0.1674	(-0.3377, 0.0029)	(-0.3301, -0.0047)	(-0.3509, 0.0006)
HFRXEWG	-0.1700	(-0.2633, -0.0767)	(-0.2710, -0.0690)	(-0.2685, -0.0768)
HFRXEWJ	-0.1820	(-0.2711, -0.0928)	(-0.2800, -0.0839)	(-0.2687, -0.0946)
HFRXGL	-0.1850	(-0.2798, -0.0902)	(-0.3015, -0.0684)	(-0.2833, -0.0927)
HFRXRVAE	-0.1860	(-0.3017, -0.0704)	(-0.3013, -0.0707)	(-0.2954, -0.0670)
HFRXGLCD	-0.1880	(-0.2844, -0.0916)	(-0.3032, -0.0727)	(-0.2798, -0.0853)
HFRXEWE	-0.2005	(-0.2901, -0.1109)	(-0.2969, -0.1041)	(-0.2931, -0.1104)
HFRXEDE	-0.2198	(-0.3743, -0.0653)	(-0.4138, -0.0257)	(-0.3720, -0.0563)
HFRXEWC	-0.2320	(-0.3216, -0.1424)	(-0.3330, -0.1309)	(-0.3185, -0.1425)
HFRXGLJ	-0.2355	(-0.3273, -0.1437)	(-0.3466, -0.1243)	(-0.3291, -0.1433)
HFRXGLG	-0.2448	(-0.3390, -0.1506)	(-0.3505, -0.1391)	(-0.3381, -0.1489)
HFRXEHV	-0.2743	(-0.4420, -0.1065)	(-0.5097, -0.0389)	(-0.4480, -0.1213)
HFRXGLE	-0.2770	(-0.3681, -0.1859)	(-0.3808, -0.1732)	(-0.3681, -0.1823)
HFRXGLC	-0.3086	(-0.4008, -0.2164)	(-0.4180, -0.1992)	(-0.4023, -0.2204)
HFRXEH	-0.3096	(-0.4419, -0.1773)	(-0.4933, -0.1258)	(-0.4401, -0.1641)
HFRXMD	-0.3656	(-0.5565, -0.1747)	(-0.6172, -0.1139)	(-0.5725, -0.1727)
HFRXEHE	-0.3730	(-0.5004, -0.2456)	(-0.5465, -0.1994)	(-0.4987, -0.2407)
HFRXEHG	-0.3778	(-0.6813, -0.0744)	(-0.6867, -0.0690)	(-0.6565, -0.0548)

Table 5.8: Summary of HFRX hedge fund capture ratios from 2011 to 2020 based on monthly data ($\kappa = 12$)

Ticker	R_X	CI.ind	CI.sc	CI.boots
HFRXCA	0.0948	(0.0384, 0.1512)	(0.0268, 0.1628)	(0.0369, 0.1526)
HFRXRVMS	-0.0617	(-0.0898, -0.0336)	(-0.1032, -0.0203)	(-0.0914, -0.0341)
HFRXRVA	-0.0690	(-0.0972, -0.0408)	(-0.1110, -0.0269)	(-0.0966, -0.0423)
HFRXAR	-0.0731	(-0.0992, -0.0469)	(-0.1112, -0.0349)	(-0.0996, -0.0465)
HFRXRVAE	-0.0842	(-0.1126, -0.0558)	(-0.1256, -0.0428)	(-0.1118, -0.0556)
HFRXEMN	-0.1068	(-0.1461, -0.0674)	(-0.1466, -0.0669)	(-0.1463, -0.0659)
HFRXMA	-0.1233	(-0.1738, -0.0727)	(-0.2020, -0.0446)	(-0.1755, -0.0745)
HFRXSDV	-0.1341	(-0.2150, -0.0532)	(-0.2297, -0.0385)	(-0.2100, -0.0527)
HFRXEW	-0.1696	(-0.1926, -0.1466)	(-0.2087, -0.1304)	(-0.1929, -0.1472)
HFRXM	-0.1704	(-0.2201, -0.1208)	(-0.2241, -0.1168)	(-0.2177, -0.1207)
HFRXMREG	-0.1763	(-0.2082, -0.1445)	(-0.2266, -0.1260)	(-0.2088, -0.1440)
HFRXEWG	-0.1815	(-0.2060, -0.1570)	(-0.2219, -0.1411)	(-0.2056, -0.1573)
HFRXEWJ	-0.1828	(-0.2060, -0.1595)	(-0.2200, -0.1455)	(-0.2059, -0.1599)
HFRXME	-0.1831	(-0.2326, -0.1337)	(-0.2370, -0.1293)	(-0.2327, -0.1350)
HFRXEWE	-0.1851	(-0.2079, -0.1623)	(-0.2230, -0.1472)	(-0.2078, -0.1632)
HFRXEWC	-0.1958	(-0.2194, -0.1722)	(-0.2330, -0.1586)	(-0.2197, -0.1734)
HFRXGL	-0.2383	(-0.2626, -0.2139)	(-0.2763, -0.2002)	(-0.2619, -0.2133)
HFRXGLCD	-0.2437	(-0.2697, -0.2176)	(-0.2833, -0.2040)	(-0.2694, -0.2183)
HFRXGLJ	-0.2491	(-0.2732, -0.2251)	(-0.2855, -0.2128)	(-0.2716, -0.2261)
HFRXED	-0.2529	(-0.2877, -0.2181)	(-0.3034, -0.2024)	(-0.2867, -0.2180)
HFRXGLG	-0.2533	(-0.2790, -0.2276)	(-0.2925, -0.2142)	(-0.2775, -0.2283)
HFRXGLE	-0.2593	(-0.2833, -0.2352)	(-0.2958, -0.2227)	(-0.2826, -0.2359)
HFRXGLC	-0.2656	(-0.2896, -0.2417)	(-0.3017, -0.2296)	(-0.2891, -0.2415)
HFRXEDE	-0.2677	(-0.3020, -0.2333)	(-0.3170, -0.2183)	(-0.2995, -0.2357)
HFRXSS	-0.2888	(-0.3281, -0.2494)	(-0.3420, -0.2355)	(-0.3271, -0.2464)
HFRXEHG	-0.3340	(-0.3908, -0.2772)	(-0.4101, -0.2579)	(-0.3862, -0.2768)
HFRXEH	-0.3715	(-0.4029, -0.3400)	(-0.4212, -0.3218)	(-0.4023, -0.3412)
HFRXMD	-0.3715	(-0.4149, -0.3280)	(-0.4277, -0.3152)	(-0.4143, -0.3263)
HFRXEHE	-0.3843	(-0.4153, -0.3532)	(-0.4327, -0.3358)	(-0.4141, -0.3514)
HFRXEHV	-0.3898	(-0.4217, -0.3580)	(-0.4375, -0.3422)	(-0.4222, -0.3573)

Table 5.9: Summary of HFRX hedge fund capture ratios from 2011 to 2020 based on daily data ($\kappa = 250$)

the highest capture ratio among all the thirty funds. From the results, the relative performance of funds using the capture ratio as the measure is rather unstable across years, especially when we use only 12 monthly data to compute the measure (see Table 5.10). In Tables 5.11-5.13, even though the top two, HFRXCA and HFRXMA, remain on the top for most periods, overall ranking for all the thirty funds is rather volatile. Considering the fact that the estimation from a small sample is subject to large estimation error, the reliability of the commonly-used 1-year capture ratio is doubtful.

5.4 Conclusion

The capture ratio is a popular measurement of fund performance. We study the asymptotic joint distributions of two capture ratios under the independence assumption and a serial dependence structure, respectively, for the underlying asset return process. The asymptotic multivariate normal distributions provide useful information on the statistical properties of the capture ratio when the sample size is large. For the 1-year monthly sample size (i.e., 12 data points only) which is commonly used in practice and the literature, we find that there are serious concerns regarding the statistical estimates and we caution that using capture ratio for fund ranking could be severely misleading. To properly apply the metric and asymptotic results, a large enough sample is necessary.

Furthermore, our study using a real-world hedge fund return data set indicates that the estimates of the capture ratio from the monthly and daily data can be significantly different. Last but not least, the estimation from a small sample is rather volatile through time and deviates from the long-term capture ratio. These observations raise questions about the practical use of the capture ratio.

	2020	2019	2018	2017	2016	2015	2014	2013	2012	2011
HFRXCA	**0.1098**	0.0524	-0.0660	4.5200	0.3368	-0.0526	-0.5584	0.0867	0.1981	-0.2270
HFRXMA	-0.1057	-0.0282	-0.1137	9.2888	0.3570	**0.4709**	0.1057	0.0581	-0.0934	-0.1816
HFRXAR	-0.1122	0.1661	**-0.0346**	6.6904	-0.0216	0.1494	-0.0401	-0.0593	-0.0341	-0.2436
HFRXSDV	-0.0428	**0.5128**	-0.2532	-47.8952	-0.1439	-0.0718	0.2767	-0.2425	-0.2703	**-0.0397**
HFRXRVMS	-0.0256	0.1778	-0.0766	-7.3995	-0.2091	-0.2006	-0.5252	-0.3413	0.0304	-0.0597
HFRXM	-0.0220	0.2958	-0.1739	-23.5189	-0.2824	-0.1537	**0.5125**	-0.3605	-0.0643	-0.2873
HFRXEMN	-0.2943	-0.1966	-0.1749	19.9096	-0.4309	0.3585	0.3183	-0.4061	-0.3397	-0.2007
HFRXRVA	-0.0157	0.1770	-0.0754	-5.8954	-0.1287	-0.2330	-0.5529	-0.2483	-0.0307	-0.2866
HFRXEW	-0.1044	0.0791	-0.2988	1.5339	0.0490	-0.1356	-0.1893	-0.1749	-0.0788	-0.3982
HFRXMREG	0.0511	0.1636	-0.3289	-5.0656	-0.0040	-0.1215	-0.0746	-0.1460	**0.2723**	-0.8594
HFRXED	-0.0062	0.2475	-0.6135	8.9704	**0.3958**	-0.4577	-0.6170	0.0409	0.0020	-0.3433
HFRXSS	0.0012	0.2254	-0.6772	19.6582	0.2427	-0.4210	-0.7611	**0.1502**	-0.0287	-0.2820
HFRXME	-0.0696	0.2139	-0.3363	-27.5124	-0.4326	-0.1772	0.4970	-0.3766	-0.0744	-0.2671
HFRXEWG	-0.1388	0.0250	-0.4108	-1.8865	-0.0298	-0.1406	-0.1754	-0.1604	-0.0783	-0.3921
HFRXEWJ	-0.1228	0.0079	-0.4189	-2.1438	-0.0454	-0.1688	-0.2458	-0.1795	-0.0931	-0.4061
HFRXGL	-0.0864	0.1708	-0.3653	1.0781	-0.1119	-0.2669	-0.2945	-0.2564	-0.0339	-0.5430
HFRXRVAE	-0.0619	0.0923	-0.2276	-9.9743	-0.2706	-0.2783	-0.5897	-0.2663	-0.0601	-0.2685
HFRXGLCD	-0.1015	0.1100	-0.4117	0.5364	-0.1297	-0.2618	-0.2280	-0.2121	0.0014	-0.5195
HFRXEWE	-0.1484	-0.0105	-0.4657	-3.4508	-0.1007	-0.1775	-0.2280	-0.1937	-0.1052	-0.3841
HFRXEDE	-0.0524	0.1590	-0.7604	4.1065	0.2444	-0.5031	-0.6712	0.0194	-0.0305	-0.3297
HFRXEWC	-0.1545	-0.0176	-0.4923	-4.4841	-0.1607	-0.2309	-0.2576	-0.2056	-0.1240	-0.4543
HFRXGLJ	-0.1036	0.0374	-0.4810	-2.8449	-0.2050	-0.2976	-0.3480	-0.2598	-0.0463	-0.5499
HFRXGLG	-0.1862	-0.0020	-0.4804	-2.3380	-0.2194	-0.2751	-0.2819	-0.2418	-0.0431	-0.5174
HFRXEHV	-0.1802	0.1451	-0.3229	11.1007	-0.1142	-0.2305	-0.3707	-0.0792	0.1560	-1.2815
HFRXGLE	-0.1894	-0.0507	-0.5359	-4.1524	-0.2633	-0.3095	-0.3335	-0.2773	-0.0658	-0.5327
HFRXGLC	-0.1957	-0.0613	-0.5921	-5.1800	-0.3214	-0.3618	-0.3646	-0.2875	-0.0868	-0.5886
HFRXEH	-0.2492	0.0119	-0.5028	18.1077	-0.4044	-0.1979	-0.2470	-0.4102	-0.0308	-1.0810
HFRXMD	-0.1224	-0.0296	-0.6559	5.0976	-0.0289	-0.5546	0.0375	-0.5217	-0.1905	-1.0701
HFRXEHE	-0.2920	-0.0708	-0.6547	13.0676	-0.5473	-0.2407	-0.2847	-0.4332	-0.0743	-1.0726
HFRXEHG	-0.3474	-0.0858	-0.7584	**34.8806**	-1.0388	-0.1955	0.0003	-0.9971	-0.0642	-0.8576

Table 5.10: 1-year rolling capture ratios based on monthly data

	2016-2020	2015-2019	2014-2018	2013-2017	2012-2016	2011-2015
HFRXCA	**0.1287**	0.0855	-0.0394	0.0140	0.0123	-0.1102
HFRXMA	-0.0227	**0.1403**	**0.1814**	**0.2894**	**0.2121**	**0.0534**
HFRXAR	-0.0099	0.0784	0.0485	0.0707	0.0342	-0.0460
HFRXSDV	-0.1016	-0.1120	-0.1300	-0.0544	-0.1008	-0.0658
HFRXRVMS	-0.0164	-0.0811	-0.1764	-0.2927	-0.2380	-0.1891
HFRXM	-0.0964	-0.1547	-0.1240	-0.1096	-0.1002	-0.1315
HFRXEMN	-0.2197	-0.0193	0.0571	0.1167	0.0054	-0.0073
HFRXRVA	-0.0001	-0.0748	-0.1811	-0.2717	-0.2322	-0.2682
HFRXEW	-0.0785	-0.0886	-0.1270	-0.0773	-0.0886	-0.2043
HFRXMREG	-0.0054	-0.0749	-0.1198	-0.0637	-0.0122	-0.3159
HFRXED	-0.0558	-0.1949	-0.3228	-0.1729	-0.1460	-0.2700
HFRXSS	-0.0909	-0.2319	-0.3755	-0.1657	-0.1586	-0.2410
HFRXME	-0.1983	-0.2623	-0.2231	-0.1733	-0.1390	-0.1363
HFRXEWG	-0.1479	-0.1580	-0.1864	-0.1056	-0.0981	-0.1993
HFRXEWJ	-0.1483	-0.1738	-0.2098	-0.1356	-0.1263	-0.2246
HFRXGL	-0.0917	-0.1528	-0.2257	-0.1957	-0.1806	-0.2966
HFRXRVAE	-0.0992	-0.1847	-0.2837	-0.3455	-0.2830	-0.2859
HFRXGLCD	-0.1248	-0.1847	-0.2357	-0.1805	-0.1554	-0.2649
HFRXEWE	-0.1848	-0.2082	-0.2392	-0.1557	-0.1396	-0.2222
HFRXEDE	-0.1591	-0.3098	-0.4312	-0.2538	-0.2021	-0.2930
HFRXEWC	-0.2051	-0.2432	-0.2800	-0.1978	-0.1755	-0.2667
HFRXGLJ	-0.1701	-0.2481	-0.3053	-0.2515	-0.2160	-0.3149
HFRXGLG	-0.2116	-0.2535	-0.2921	-0.2315	-0.1980	-0.2876
HFRXEHV	-0.1031	-0.1175	-0.2110	-0.1507	-0.0956	-0.4701
HFRXGLE	-0.2457	-0.3025	-0.3404	-0.2756	-0.2337	-0.3172
HFRXGLC	-0.2700	-0.3424	-0.3863	-0.3169	-0.2696	-0.3571
HFRXEH	-0.2087	-0.1906	-0.2411	-0.1871	-0.1955	-0.4323
HFRXMD	-0.2333	-0.3835	-0.3521	-0.3502	-0.2801	-0.5247
HFRXEHE	-0.3070	-0.3020	-0.3473	-0.2634	-0.2500	-0.4550
HFRXEHG	-0.3597	-0.2766	-0.2900	-0.2901	-0.3661	-0.3990

Table 5.11: 5-year rolling capture ratios based on monthly data

	2020	2019	2018	2017	2016	2015	2014	2013	2012	2011
HFRXCA	-0.2137	0.0297	**0.1608**	**0.6277**	**0.3803**	**0.1146**	-0.1951	**0.3072**	**0.1234**	0.0279
HFRXMA	-0.3995	-0.1240	-0.1103	-0.0245	0.0018	0.0256	-0.0544	0.0522	-0.1236	-0.3572
HFRXAR	-0.2273	0.0074	-0.0361	-0.0103	-0.0278	-0.0389	-0.1729	0.0534	-0.0109	-0.1946
HFRXSDV	**-0.0725**	0.0392	-0.3287	-0.2706	-0.2684	-0.0486	-0.1608	-0.1805	-0.2051	**0.5245**
HFRXRVMS	-0.0879	0.0555	-0.1143	-0.0408	-0.0303	-0.2006	-0.1877	-0.0261	0.0640	0.0142
HFRXM	-0.1336	-0.0043	-0.2440	-0.2485	-0.2952	-0.1740	-0.1403	-0.1935	-0.1244	-0.1203
HFRXEMN	-0.1853	-0.0757	-0.1431	-0.1460	-0.1023	0.0371	**-0.0301**	-0.0450	-0.1200	-0.2504
HFRXRVA	-0.0992	**0.0556**	-0.0903	0.0056	-0.0044	-0.1872	-0.2076	0.0070	0.0347	-0.1258
HFRXEW	-0.2570	-0.0410	-0.2505	-0.0493	-0.0975	-0.1967	-0.2556	-0.0661	-0.1048	-0.2938
HFRXMREG	-0.1991	0.0078	-0.2853	0.0101	-0.1735	-0.2426	-0.2287	0.0007	0.0208	-0.4431
HFRXED	-0.1943	0.0328	-0.4026	-0.0471	-0.1062	-0.3775	-0.4500	-0.1382	-0.1745	-0.3824
HFRXSS	-0.1800	-0.0038	-0.4653	-0.0496	-0.1650	-0.4169	-0.4889	-0.1283	-0.1865	-0.4190
HFRXME	-0.1417	-0.0317	-0.2824	-0.2930	-0.3201	-0.1803	-0.1462	-0.1951	-0.1271	-0.1109
HFRXEWG	-0.2712	-0.0787	-0.2763	-0.0946	-0.1126	-0.2103	-0.2552	-0.0614	-0.1041	-0.2940
HFRXEWJ	-0.2549	-0.1043	-0.2772	-0.0914	-0.1150	-0.2054	-0.2612	-0.0706	-0.1093	-0.2943
HFRXGL	-0.2360	-0.0345	-0.3597	-0.1185	-0.2047	-0.3133	-0.3251	-0.1571	-0.1411	-0.3622
HFRXRVAE	-0.1093	0.0278	-0.1279	-0.0443	-0.0348	-0.2020	-0.2160	0.0030	0.0291	-0.1192
HFRXGLCD	-0.2441	-0.0541	-0.3813	-0.1370	-0.2149	-0.3044	-0.3120	-0.1539	-0.1390	-0.3751
HFRXEWE	-0.2632	-0.0755	-0.2884	-0.1073	-0.1252	-0.2086	-0.2611	-0.0687	-0.1125	-0.2932
HFRXEDE	-0.2011	-0.0032	-0.4349	-0.1022	-0.1316	-0.3886	-0.4569	-0.1419	-0.1817	-0.3825
HFRXEWC	-0.2644	-0.1237	-0.2959	-0.1217	-0.1358	-0.2282	-0.2658	-0.0744	-0.1158	-0.3012
HFRXGLJ	-0.2343	-0.0835	-0.3823	-0.1604	-0.2190	-0.3195	-0.3302	-0.1605	-0.1451	-0.3626
HFRXGLG	-0.2691	-0.0915	-0.3828	-0.1632	-0.2211	-0.3243	-0.3247	-0.1534	-0.1410	-0.3594
HFRXEHV	-0.5305	-0.1760	-0.4616	-0.2827	-0.3188	-0.4330	-0.3937	-0.2453	-0.1777	-0.6155
HFRXGLE	-0.2593	-0.1140	-0.3958	-0.1780	-0.2301	-0.3226	-0.3340	-0.1595	-0.1505	-0.3627
HFRXGLC	-0.2605	-0.1154	-0.4032	-0.1912	-0.2402	-0.3388	-0.3359	-0.1636	-0.1538	-0.3701
HFRXEH	-0.4283	-0.1690	-0.4919	-0.1517	-0.3340	-0.4100	-0.3558	-0.2413	-0.2401	-0.6097
HFRXMD	-0.3569	-0.1531	-0.4818	-0.3463	-0.1976	-0.5255	-0.2697	-0.2461	-0.2843	-0.5418
HFRXEHE	-0.4327	-0.1893	-0.5200	-0.2009	-0.3553	-0.4182	-0.3629	-0.2444	-0.2478	-0.6129
HFRXEHG	-0.2234	-0.1690	-0.5407	0.1914	-0.3483	-0.3887	-0.2504	-0.2250	-0.2630	-0.5706

Table 5.12: 1-year rolling capture ratios based on daily data

	2016-2020	2015-2019	2014-2018	2013-2017	2012-2016	2011-2015
HFRXCA	**0.1253**	**0.2082**	**0.1556**	**0.1857**	**0.1296**	**0.0635**
HFRXMA	-0.1418	-0.0433	-0.0312	0.0039	-0.0209	-0.1068
HFRXAR	-0.0651	-0.0234	-0.0597	-0.0454	-0.0445	-0.0815
HFRXSDV	-0.2008	-0.1912	-0.2197	-0.1868	-0.1711	-0.0484
HFRXRVMS	-0.0474	-0.0800	-0.1217	-0.1096	-0.0871	-0.0770
HFRXM	-0.1837	-0.1942	-0.2135	-0.2046	-0.1864	-0.1551
HFRXEMN	-0.1252	-0.0786	-0.0696	-0.0463	-0.0495	-0.0853
HFRXRVA	-0.0329	-0.0602	-0.1073	-0.0931	-0.0817	-0.1045
HFRXEW	-0.1485	-0.1407	-0.1788	-0.1415	-0.1494	-0.1907
HFRXMREG	-0.1430	-0.1614	-0.1994	-0.1454	-0.1381	-0.2091
HFRXED	-0.1760	-0.2239	-0.3081	-0.2536	-0.2691	-0.3204
HFRXSS	-0.2145	-0.2715	-0.3539	-0.2865	-0.3028	-0.3512
HFRXME	-0.2075	-0.2201	-0.2353	-0.2181	-0.1949	-0.1564
HFRXEWG	-0.1708	-0.1645	-0.1959	-0.1521	-0.1546	-0.1928
HFRXEWJ	-0.1711	-0.1678	-0.1965	-0.1540	-0.1574	-0.1952
HFRXGL	-0.2050	-0.2265	-0.2725	-0.2328	-0.2358	-0.2704
HFRXRVAE	-0.0585	-0.0893	-0.1324	-0.1114	-0.0947	-0.1098
HFRXGLCD	-0.2197	-0.2374	-0.2769	-0.2304	-0.2313	-0.2673
HFRXEWE	-0.1748	-0.1711	-0.2036	-0.1584	-0.1605	-0.1958
HFRXEDE	-0.2012	-0.2522	-0.3307	-0.2699	-0.2799	-0.3257
HFRXEWC	-0.1874	-0.1888	-0.2147	-0.1696	-0.1700	-0.2050
HFRXGLJ	-0.2237	-0.2493	-0.2880	-0.2435	-0.2421	-0.2739
HFRXGLG	-0.2345	-0.2528	-0.2886	-0.2431	-0.2411	-0.2718
HFRXEHV	-0.3707	-0.3436	-0.3692	-0.3290	-0.3291	-0.4107
HFRXGLE	-0.2421	-0.2628	-0.2969	-0.2495	-0.2469	-0.2761
HFRXGLC	-0.2473	-0.2727	-0.3065	-0.2586	-0.2547	-0.2836
HFRXEH	-0.3473	-0.3399	-0.3597	-0.3056	-0.3223	-0.3950
HFRXMD	-0.3424	-0.3797	-0.3799	-0.3361	-0.3411	-0.4019
HFRXEHE	-0.3675	-0.3635	-0.3803	-0.3198	-0.3319	-0.4005
HFRXEHG	-0.3061	-0.3301	-0.3371	-0.2474	-0.2999	-0.3588

Table 5.13: 5-year rolling capture ratios based on daily data

Chapter 6

Conclusion and Future Work

In the previous chapters, we study four different investment problems. Chapters 2 and 3 revisit two portfolio selection problems: the Kelly criterion problem and the mean-variance optimization problem, under the reinforcement learning framework. Chapter 4 proposes a new selection method in the valuation of large variable annuity portfolios. Chapter 5 studies the statistical properties and estimation of the capture ratio. Following we summarize the main results and directions of future work.

In Chapter 2, we build an entropy-regularization framework to study the Kelly criterion problem, inspired by Wang et al. (2019) and Wang and Zhou (2020). The aim is to improve the empirical performance of the full Kelly strategy and mitigate the impact of the estimation error. In the study, we include a general time-varying temperature parameter in the regularization term to balance the degree of exploration and exploitation. We consider three specific functional forms for the temperature parameter. Different temperature parameters lead to different time-varying patterns of the variance term in the Gaussian distribution of optimal control: increasing, constant, and decreasing over time. We call the resulting optimal strategies the RL Kelly strategies and implement them by RL algorithms. In particular, We use simulations to prove the convergence of the RL algorithms. We compare the RL Kelly strategies to several MLE-based strategies, based on extensive simulation studies. The results show that the RL Kelly strategies yield significantly better and more robust performance than the benchmark strategies under different market settings. Particularly, the RL strategy with a time-decaying temperature parameter is the best in that it not only achieves the highest average terminal log-return but also learns the entire terminal wealth distribution more precisely. The outperformance is also robust under model misspecification. Thus, the RL Kelly strategy provides a practical improvement to those existing Kelly strategies.

In Chapter 3, we study the discrete-time mean-variance optimization problem. The continuous-time problem under the RL framework has been studied by Wang and Zhou (2020). They study the theoretical problem under the continuous-time setting and implement it under the discrete-time setting, with discretized time intervals. This is a realization of a continuous trading strategy in a, say daily, trading frequency. They find the entropy-regularization framework does improve the performance of the optimal strategy. We directly study the problem under the discrete-time setting and derive the optimal strategy. The optimal strategy also follows a Gaussian distribution, with parameters depending on time, wealth and discretization step. We find that our strategy converges to the discretized strategy in Wang and Zhou (2020), as the discretization step tends to zero. The optimal strategy can be implemented using a stochastic gradient descent algorithm. Through simulation examples, we find that the continuous-time strategy is subject to discretization errors, especially when the trading frequency is large. However, our discrete-time strategy performs generally better than the continuous-time strategy, in terms of a faster convergence of the mean return, an improved terminal Sharpe ratio, as well as robustness across different trading frequencies.

For the above RL portfolio selection problems, there are several possible directions of future work. First, a natural extension of Chapter 2 is to study the discrete-time Kelly criterion problem. Based on the results from Chapter 3, the discrete-time RL Kelly strategy could potentially further improve practical performance. Second, both problems assume a basic asset price process: geometric Brownian motion. Even though the resulting algorithms work under model misspecification (Jiang et al. (2022a)), it is interesting to study more realistic underlying asset models such as the local volatility model. Based on our preliminary investigations, with more complicated asset models, solving the optimization problem through the Hamilton-Jacobi-Bellman (HJB) equation is not feasible. Hence, other approaches could be helpful. For example, Dai et al. (2020) adopt the backward stochastic differential equation (BSDE) techniques to solve the exploratory mean-variance problem of the log return. Third, under the RL framework, the optimal strategy follows a time-varying Gaussian distribution. The Gaussian mean and variance are continuous in time. However, when implemented, the trading actions are drawn from the distributions, and hence not continuous in time. This sometimes gives rather volatile actions over time. It will be an interesting topic to study how to smooth the empirical action sequence or impose an action range to reduce the volatility. This direction is related to the improvement in terms of transaction costs and turnover. As discussed in Section 2.5.3, another regularization term could be added to the problem to penalize transaction costs or turnover.

The valuation problem in Chapter 4 has drawn the attention of researchers and practitioners because of the popularity of variable annuity products. A popular valuation procedure to balance the accuracy and computational cost is a two-step framework. In the first step, a reasonable set of representative contracts is selected through clustering methods or sampling methods, based on the contract attributes. Their values are obtained through nested simulation. The second step builds an interpolation model to estimate the values of all remaining contracts. Methods commonly used are spatial interpolation methods such as the kriging method. We study the selection part of the valuation from another perspective. Instead of selecting all contracts based on the attributes, We propose a two-phase selection procedure. We first select a sub-sample of the representative contracts based on the attributes. Using their values from the nested simulation, we build a model for the candidate contracts' contract values. Then, the remaining representatives are selected from the candidate contracts, based on the model of the contract values. In our study, we model the contract values as Gaussian random variables, which is also assumed in the universal kriging method. To diversify the second subset of representative contracts from the first part, we calculate the conditional distributions of the contract values, on the first part of representatives' values. The second part of the selection is then based on the Wasserstein distance among conditional Gaussian distributions, using our proposed conditional clustering algorithm. Using public VA data (attributes and fair market values), we compare our proposed method with the conventional two-step framework. Using either the k-prototype clustering method or the hierarchical k-means clustering method, our procedure produces more accurate estimates of the fair market values of large VA portfolios, with a similar computational cost. A further benefit is that the procedure is robust with respect to model inputs.

The future work on this topic has two possible directions. With the flexibility of the above proposed valuation procedure, we could incorporate more selection methods to improve accuracy, for example, the simple random sampling and clustering method proposed by Feng et al. (2020). On the other hand, we model the candidate contracts' contract values as Gaussian random variables and calculate their conditional distributions. The original and conditional distributions can be viewed as the prior and posterior distributions in Bayes' theorem, respectively. Hence, if more representative contracts are gradually selected, the posterior distribution for the candidate contracts can be updated gradually. In this case, the Bayesian optimization methods could be adopted, with a specific objective to measure the similarities among the contracts. This could be helpful to determine the size of representatives as well as to find better representatives.

Chapter 5 studies the capture ratio. The capture ratio is commonly used to measure fund managers' performance against a market benchmark, for example, the S&P 500 index. It serves as an important criterion employed by Morningstar (Haslem (2014)) in selecting funds. Even though the capture ratio is popular in practice (Cox and Goff (2013)), we are not aware of any previous research regarding its statistical estimation. Investors extensively use the capture ratio to measure and compare fund performance. The use of a naive calculation/estimation of the capture ratio based on empirical data could be misleading if the statistical properties of the estimator are unknown. Hence, in this chapter, we study the asymptotic joint distributions of the ratios and test hypotheses regarding the relative performance of two funds. Using simulations, we find that the capture ratio estimates based on different return frequencies can significantly differ in magnitude. They can also be substantially biased when the underlying model is misspecified or the sample size is not large enough. Using a real-world hedge fund return dataset, we find that the estimates of capture ratios from monthly and daily data can be significantly different. Moreover, the estimation from a small sample is rather volatile through time and deviates from the long-term capture ratio. All our findings raise questions about the practical use of the capture ratio. If the sample size is not reasonably large or the underlying model assumption is not appropriate, the selection or ranking based on the capture ratio would be misleading and hence increase the risk in fund management. Based on our findings, future work could focus on bias-corrected estimations such that an estimation based on a small sample size could still be informative in practice.

References

Bacinello, A. R., Millossovich, P., Olivieri, A., and Pitacco, E. (2011). Variable annuities: A unifying valuation approach. *Insurance: Mathematics and Economics*, 49(3):285–297.

Bacinello, A. R. and Ortu, F. (1993). Pricing equity-linked life insurance with endogenous minimum guarantees. *Insurance: Mathematics and Economics*, 12(3):245–257.

Bauer, D., Kling, A., and Russ, J. (2008). A universal pricing framework for guaranteed minimum benefits in variable annuities. *ASTIN Bulletin: The Journal of the IAA*, 38(2):621–651.

Bauer, D., Reuss, A., and Singer, D. (2012). On the calculation of the solvency capital requirement based on nested simulations. *ASTIN Bulletin: The Journal of the IAA*, 42(2):453–499.

Bello, Z. (2014). The characteristics of alternative mutual funds. *Journal of Finance and Economics*, 2(4):68–77.

Bottou, L. and Bengio, Y. (1994). Convergence properties of the k-means algorithms. *Advances in neural information processing systems*, 7.

Boyle, P. P. and Hardy, M. R. (1997). Reserving for maturity guarantees: Two approaches. *Insurance: Mathematics and Economics*, 21(2):113–127.

Boyle, P. P. and Schwartz, E. S. (1977). Equilibrium prices of guarantees under equity-linked contracts. *Journal of Risk and Insurance*, pages 639–660.

Brennan, M. J. and Schwartz, E. S. (1976). The pricing of equity-linked life insurance policies with an asset value guarantee. *Journal of Financial Economics*, 3(3):195–213.

Chang, C. E. and Krueger, T. M. (2013). The VICEX fund: Recent shortcomings of a long-run success story. *Journal of Management and Sustainability*, 3:131.

Chang, C. E., Krueger, T. M., and Mbanga, C. T. (2019). Intelligent selection of smart beta mutual funds. *The Journal of Wealth Management*, 22(1):10–23.

Chang, C. E., Ragan, K. P., and Witte, H. D. (2013). ETFs versus CEFs: Performance in international equity investing. *International Journal of Economics and Finance*, 5(12):79–85.

Chen, J. M. (2014). Measuring gaps between hypothetical investment returns and actual investor returns. *Available at SSRN 2500079*.

Cline, B. N. and Gilstrap, C. (2021). Active share: A blessing and a curse. *Journal of Financial Research*.

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Academic Press.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley.

Cox, D. R. and Goff, D. C. (2013). Capture ratios: A popular method of measuring portfolio performance in practice. *Journal of Economics and Finance Education*, 2:50–55.

Coy, J. M. and Robbins, E. J. (2021). Are all capture ratios created equal? *The Journal of Index Investing*, 12(2):60–83.

Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

Dai, M., Yuchao, D., and Jia, Y. (2020). Learning equilibrium mean-variance strategy. *Available at SSRN 3770818*.

Davis, M. and Lleo, S. (2013). Fractional Kelly strategies in continuous time: Recent developments. In *Handbook of the Fundamentals of Financial Decision Making: Part II*, pages 753–787. World Scientific.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245.

Doyle, D. and Groendyke, C. (2018). Using neural networks to price and hedge variable annuity guarantees. *Risks*, 7(1):1.

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC press.

Feng, B. M. and Liu, K. (2020). Path generation methods for valuation of large variable annuities portfolio using Quasi-Monte Carlo simulation. In *2020 Winter Simulation Conference (WSC)*, pages 481–491. IEEE.

Feng, B. M., Tan, Z., and Zheng, J. (2020). Efficient simulation designs for valuation of large variable annuity portfolios. *North American Actuarial Journal*, 24(2):275–289.

Firoozi, D. and Jaimungal, S. (2022). Exploratory lqg mean field games with entropy regularization. *Automatica*, 139:110177.

Gan, G. (2013). Application of data clustering and machine learning in variable annuity valuation. *Insurance: Mathematics and Economics*, 53(3):795–801.

Gan, G. (2018). Valuation of large variable annuity portfolios using linear models with interactions. *Risks*, 6(3):71.

Gan, G. and Lin, X. S. (2015). Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance: Mathematics and Economics*, 62:138–150.

Gan, G. and Lin, X. S. (2017). Efficient greek calculation of variable annuity portfolios for dynamic hedging: A two-level metamodeling approach. *North American Actuarial Journal*, 21(2):161–177.

Gan, G. and Valdez, E. A. (2016). An empirical comparison of some experimental designs for the valuation of large variable annuity portfolios. *Dependence Modeling*, 4(1).

Gan, G. and Valdez, E. A. (2017). Valuation of large variable annuity portfolios: Monte Carlo simulation and synthetic datasets. *Dependence Modeling*, 5(1):354–374.

Gan, G. and Valdez, E. A. (2018). Regression modeling for the valuation of large variable annuity portfolios. *North American Actuarial Journal*, 22(1):40–54.

Gan, G. and Valdez, E. A. (2019). *Metamodeling for variable annuities*. CRC Press.

Gan, G. and Valdez, E. A. (2020). Data clustering with actuarial applications. *North American Actuarial Journal*, 24(2):168–186.

Goll, T. and Kallsen, J. (2000). Optimal portfolios for logarithmic utility. *Stochastic Processes and their Applications*, 89(1):31–48.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT press.

Gottesman, A. and Morey, M. (2021). What do capture ratios really capture in mutual fund performance? *The Journal of Investing*, 30(6):99–112.

Guo, X., Xu, R., and Zariphopoulou, T. (2022). Entropy regularization for mean field games with learning. *Mathematics of Operations Research*, pages 1–21.

Han, Y., Yu, P. L. H., and Mathew, T. (2019). Shrinkage estimation of Kelly portfolios. *Quantitative Finance*, 19(2):277–287.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054.

Hardy, M. (2003). *Investment guarantees: Modeling and risk management for equity-linked life insurance*, volume 168. John Wiley & Sons.

Hardy, M. R. (2000). Hedging and reserving for single-premium segregated fund contracts. *North American Actuarial Journal*, 4(2):63–74.

Haslem, J. A. (2014). Morningstar mutual fund measures and selection model. *The Journal of Wealth Management*, 17(2):19–30.

Hejazi, S. A. and Jackson, K. R. (2016). A neural network approach to efficient valuation of large portfolios of variable annuities. *Insurance: Mathematics and Economics*, 70:169–181.

Hejazi, S. A., Jackson, K. R., and Gan, G. (2017). A spatial interpolation framework for efficient valuation of large portfolios of variable annuities. *arXiv preprint arXiv:1701.04134*.

Huang, J. Z., Ng, M. K., Rong, H., and Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):657–668.

Huang, Y., Mamon, R., and Xiong, H. (2022). Valuing guaranteed minimum accumulation benefits by a change of numéraire approach. *Insurance: Mathematics and Economics*, 103:1–26.

Huang, Y. T. and Kwok, Y. K. (2016). Regression-based monte carlo methods for stochastic control models: Variable annuities with lifelong guarantees. *Quantitative Finance*, 16(6):905–928.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pages 21–34. Citeseer.

Ishii, S., Yoshida, W., and Yoshimoto, J. (2002). Control of exploitation–exploration meta-parameter in reinforcement learning. *Neural Networks*, 15(4-6):665–687.

Jacobsen, B. (2009). The value and price of active management. *Available at SSRN 1528484*.

Jiang, R., Saunders, D., and Weng, C. (2022a). The reinforcement learning kelly strategy. *Quantitative Finance*, 22(8):1445–1464.

Jiang, R., Saunders, D., and Weng, C. (2022b). The statistics of capture ratios. *Journal of Risk*, 25(1).

Jiang, R., Saunders, D., and Weng, C. (2022c). Two-phase selection of representative contracts for valuation of large variable annuity portfolios. *Available at SSRN 4277936*.

Johnson, S. G. (2022). The NLopt nonlinear-optimization package.

Kallberg, J. G. and Ziemba, W. T. (1984). Mis-specifications in portfolio selection problems. In *Risk and Capital*, pages 74–87. Springer.

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Kim, S., Pasupathy, R., and Henderson, S. G. (2015). A guide to sample average approximation. *Handbook of simulation optimization*, pages 207–243.

Kuhle, J. L. and Lin, E. C. (2018). Evaluating real estate mutual fund performance using the Morningstar upside/downside capture ratio. *Global Journal of Business Research*, 12(1):15–22.

Li, D., Chan, T.-F., and Ng, W.-L. (1998). Safety-first dynamic portfolio selection. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, 4(4):585–600.

Li, D. and Ng, W.-L. (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 10(3):387–406.

Lin, X. S. and Yang, S. (2020a). Efficient dynamic hedging for large variable annuity portfolios with multiple underlying assets. *ASTIN Bulletin: The Journal of the IAA*, 50(3):913–957.

Lin, X. S. and Yang, S. (2020b). Fast and efficient nested simulation for large variable annuity portfolios: A surrogate modeling approach. *Insurance: Mathematics and Economics*, 91:85–103.

Liu, K. and Tan, K. S. (2021). Real-time valuation of large variable annuity portfolios: A green mesh approach. *North American Actuarial Journal*, 25(3):313–333.

Lo, A. W. (2002). The statistics of Sharpe ratios. *Financial Analysts Journal*, 58(4):36–52.

MacLean, L. C., Thorp, E. O., Zhao, Y., and Ziemba, W. T. (2011). Medium term simulations of the full Kelly and fractional Kelly investment strategies. In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, pages 543–561. World Scientific.

MacLean, L. C., Thorp, E. O., and Ziemba, W. T. (2010). Long-term capital growth: The good and bad properties of the Kelly and fractional Kelly capital growth criteria. *Quantitative Finance*, 10(7):681–687.

Merton, R. (1971). Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):373–413.

Nekrasov, V. (2014). Kelly criterion for multivariate portfolios: A model-free approach. *Available at SSRN 2259133*.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. IEEE.

Powell, M. J. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer.

Powell, M. J. (1998). Direct search algorithms for optimization calculations. *Acta numerica*, 7:287–336.

Reynolds, C. and Man, S. (2008). Nested stochastic pricing: The time has come. Product Matters. *Society of Actuaries*, 71:16–20.

Rising, J. K. and Wyner, A. J. (2012). Partial Kelly portfolios and shrinkage estimators. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 1618–1622. IEEE.

Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178.

Selim, S. Z. and Ismail, M. A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1):81–87.

Shen, W., Wang, B., Pu, J., and Wang, J. (2019). The Kelly growth optimal portfolio with ensemble learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1134–1141.

Shen, Z. and Weng, C. (2019). A backward simulation method for stochastic optimal control problems. *Manuscript*.

Shen, Z. and Weng, C. (2020). Pricing bounds and bang-bang analysis of the Polaris variable annuities. *Quantitative Finance*, 20:147–171.

Stanford, J. L. and Vardeman, S. B. (1994). *Statistical methods for physical science*, volume 28. Academic Press.

Steck, H. and Jaakkola, T. S. (2003). Bias-corrected bootstrap and model uncertainty. In *NIPS*, pages 521–528. Citeseer.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT press.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Venugopal, M. and Sophia, S. (2020). Examining Sharpe ratio, ASR, Sortino, Treynor and info ratio in Indian equity mutual funds during the pandemic. *International Journal of Management*, 11(11).

Wang, H., Zariphopoulou, T., and Zhou, X. Y. (2019). Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*.

Wang, H. and Zhou, X. Y. (2019). Large scale continuous-time mean-variance portfolio allocation via reinforcement learning. *Available at SSRN 3428125*.

Wang, H. and Zhou, X. Y. (2020). Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273–1308.

Weng, C. and Zhuang, S. C. (2017). CDF formulation for solving an optimal reinsurance problem. *Scandinavian Actuarial Journal*, 2017(5):395–418.

Xu, W., Chen, Y., Coleman, C., and Coleman, T. F. (2018). Moment matching machine learning methods for risk management of large variable annuity portfolios. *Journal of Economic Dynamics and Control*, 87:1–20.

Yang, S. S. and Dai, T.-S. (2013). A flexible tree for evaluating guaranteed minimum withdrawal benefits under deferred life annuity contracts with various provisions. *Insurance: Mathematics and Economics*, 52(2):231–242.

Ziemba, W. T. (2016). Understanding the Kelly capital growth investment strategy. *Investment Strategies*, 3:49–55.

APPENDICES

Appendix A

Chapter 2 Appendices

A.1 Proofs of Results

A.1.1 Proof of Theorem 2.3.1

The Lagrangian function for the maximization problem is given by

$$\begin{aligned}\mathcal{L}(\pi_t,\eta) &:= \rho\sigma v_x(t,x)\int_{\mathbb{R}} u\pi_t(u)du + \frac{1}{2}\sigma^2 v_{xx}(t,x)\int_{\mathbb{R}} u^2\pi_t(u)du \\ &\quad -\lambda_a(t)\int_{\mathbb{R}}\pi_t(u)\log\pi_t(u)du + \eta\int_{\mathbb{R}}\pi_t(u)du \\ &=: \int_{\mathbb{R}} L(u,\pi_t(u))du, \end{aligned} \tag{A.1}$$

where η denotes the Lagrangian multiplier, and $L(u,\pi_t(u))$ is given by

$$L(u,\pi_t(u)) = \rho\sigma v_x(t,x)u\pi_t(u) + \frac{1}{2}\sigma^2 v_{xx}(t,x)u^2\pi_t(u) - \lambda_a(t)\pi_t(u)\log\pi_t(u) + \eta\pi_t(u).$$

By a standard Lagrangian duality argument (e.g., see Lemma 4.3 of Weng and Zhuang (2017)), if $\pi^* := \pi^*_{\eta^*} \in \mathcal{P}(\mathbb{R})$ maximizes $\mathcal{L}(\pi,\eta^*)$ for some $\eta^* \in \mathbb{R}$ satisfying $\int_{\mathbb{R}}\pi^*(u)du = 1$, then π^* is a solution to the maximization problem in equation (2.9). Consequently, we focus on analyzing the optimizer(s) of $\mathcal{L}(\cdot,\eta)$ before we show the optimality of π^*_t in equation (2.10).

To derive a maximizer of $\mathcal{L}(\cdot,\eta)$, we apply a pointwise maximization procedure and analyze the integrand in (A.1), $L(u,\pi_t(u))$. Since $\pi\log\pi$ is convex in π while the other items in the expression of $L(u,\pi)$ are linear in π, $L(u,\pi)$ is concave as a function of π. Accordingly, the first-order optimality condition is sufficient to determine its maximizers, whereby we take the partial derivative of $L(u,\pi_t(u))$ with respect to $\pi_t(u)$ and equate it to zero to get:

$$\rho\sigma v_x(t,x)u + \frac{1}{2}\sigma^2 v_{xx}(t,x)u^2 - \lambda_a(t)\log\pi_t(u) - \lambda_a(t) + \eta = 0,$$

which gives

$$\pi_t(u) = \exp\left(\frac{1}{\lambda_a(t)}\left[\frac{1}{2}\sigma^2 v_{xx}(t,x)u^2 + \rho\sigma v_x(t,x)u\right] - \lambda_a(t) + \eta\right).$$

Taking η to scale $\pi_t(u)$ to satisfy the constraint $\int_{\mathbb{R}}\pi_t(u)du = 1$ yields the desired optimality of π_t^* in (2.10).

A.1.2 Proof of Theorem 2.3.2

We start from conjecturing the solution to the PDE (2.11) in the form $v(t,x) = f(t)\log x + g_a(t)$ for some functions f and g_a defined on $[0,T]$ with conditions $f(T) = 1$ and $g_a(T) = 0$. This yields $v_x = x^{-1}f(t)$, $v_{xx} = -x^{-2}f(t)$ and $v_t = f'(t)\log(x) + g_a'(t)$. It is straightforward to use equation (2.10) to verify equation (2.14) for the optimal control $\pi_t^*(u;x,\lambda_a(t))$. Furthermore, substituting the expressions of v_t, v_x and v_{xx} (in terms of f and g_a) into the PDE (2.11) yields:

$$\begin{aligned}
v_t(t,x) - &\frac{\rho^2(v_x(t,x))^2}{2v_{xx}(t,x)} - \frac{\lambda_a(t)}{2}\log\left(-\frac{\sigma^2 v_{xx}(t,x)}{2\pi\lambda_a(t)}\right) \\
&= f^{'}(t)\log x + g_a'(t) + \frac{\rho^2}{2}f(t) - \frac{\lambda_a(t)}{2}\log\frac{\sigma^2 x^{-2}f(t)}{2\pi\lambda_a(t)} \\
&= f^{'}(t)\log x + g_a'(t) + \frac{\rho^2}{2}f(t) - \frac{\lambda_a(t)}{2}\log x^{-2} - \frac{\lambda_a(t)}{2}\log\frac{\sigma^2 f(t)}{2\pi\lambda_a(t)} \\
&= (f^{'}(t) + \lambda_a(t))\log x + g_a'(t) + \frac{\rho^2}{2}f(t) - \frac{\lambda_a(t)}{2}\log\frac{\sigma^2 f(t)}{2\pi\lambda_a(t)} \\
&= 0.
\end{aligned}$$

The above equation implies the following ordinary differential equations (ODEs):

$$\begin{cases} f^{'}(t) + \lambda_a(t) = 0, \\ g_a'(t) + \frac{\rho^2}{2}f(t) - \frac{\lambda_a(t)}{2}\log\frac{\sigma^2 f(t)}{2\pi\lambda_a(t)} = 0, \end{cases}$$

with terminal conditions $f(T) = 1$ and $g_a(T) = 0$, which are the same as equations (2.12) and (2.13).

Using equations (2.4)-(2.6), it is easy to get the SDE in equation (2.15) for the exploratory wealth process under the optimal control, and the verification for results in equations (2.17) and (2.18) also follows trivially.

A.1.3 Linearly Decaying $\lambda_a(t)$

Theorem A.1.1. *Consider the optimization problem (2.8) with a linearly time-decaying $\lambda_a(t)$:*

$$\lambda_a(t) = -2\lambda_0 t + \lambda_1$$

with $\lambda_0, \lambda_1 > 0$ and $2\lambda_0 T < \lambda_1$ to ensure $\lambda_a(t) > 0,\ \forall t \in [0, T]$. Then, the value function is given by

$$\begin{aligned} V^a(t,x) =& f(t)\log x + \frac{\lambda_0\rho^2}{6}(T^3 - t^3) - \frac{\lambda_1\rho^2}{4}(T^2 - t^2) + \frac{\rho^2}{2}(-\lambda_0 T^2 + \lambda_1 T + 1)(T - t) \\ &+ \left(\frac{1}{2} - \log\frac{\sigma^2}{2\pi}\right)\frac{f(t) - 1}{2} - \frac{f(t)}{2}\log f(t) + \frac{\lambda_a^2(t)}{8\lambda_0}\log\lambda_a(t) - \frac{\lambda_a^2(T)}{8\lambda_0}\log\lambda_a(T) \end{aligned}$$

where $f(t) = -\lambda_0(T^2 - t^2) + \lambda_1(T - t) + 1$, and the optimal control is given by

$$\pi_t^{\lambda *}(u; x) \sim \mathcal{N}\left(\frac{\rho x}{\sigma},\ \frac{x^2}{\sigma^2}\frac{-2\lambda_0 t + \lambda_1}{-\lambda_0(T^2 - t^2) + \lambda_1(T - t) + 1}\right)$$

for which the variance

(1) increases in $[0, T]$ if $(2\lambda_0 T - \lambda_1)^2 - 2\lambda_0 \geq 0$;

(2) decreases in $[0, T]$ if $(\lambda_1 - \lambda_0 T)^2 + \lambda_0^2 T^2 - 2\lambda_0 \leq 0$;

(3) first increases then decreases in $[0, T]$, otherwise.

Proof. We apply Theorem 2.3.2 and only show how we derive the expression for f and g_a in the conjectured value function $V^a(t, x; \lambda_a(t)) = f(t)\log x + g_a(t)$ and the variance in the optimal Gaussian control distribution.

By virtue of Theorem 2.3.2, we apply $f'(t) = -\lambda_a(t)$ to get

$$f(t) = -\lambda_0(T^2 - t^2) + \lambda_1(T - t) + 1.$$

Therefore, the variance of the optimal control is given by

$$\frac{x^2\lambda_a(t)}{\sigma^2 f(t)} = \frac{x^2}{\sigma^2}\frac{-2\lambda_0 t + \lambda_1}{-\lambda_0(T^2-t^2)+\lambda_1(T-t)+1} =: \frac{x^2}{\sigma^2}\zeta(t)$$

for which the derivative is

$$\zeta^{'}(t) = \frac{\kappa(t)}{(-\lambda_0(T^2-t^2)+\lambda_1(T-t)+1)^2}$$

with

$$\kappa(t) := 2\lambda_0^2 t^2 - 2\lambda_0\lambda_1 t + \lambda_1^2 + 2\lambda_0^2 T^2 - 2\lambda_0\lambda_1 T - 2\lambda_0.$$

Therefore, we can focus on the function $\kappa(t)$ and investigate its sign for the changing pattern of the variance term. Clearly κ is decreasing in $t \in [0,T]$ since $\lambda_1 > 2\lambda_0 T$. Furthermore, we observe that $\kappa(0) = (\lambda_1 - \lambda_0 T)^2 + \lambda_0^2 T^2 - 2\lambda_0$ and $\kappa(T) = (2\lambda_0 T - \lambda_1)^2 - 2\lambda_0$. So, we have

$$\begin{cases} \zeta^{'}(t) \geq 0, \ \forall t \in [0,T], & \text{if } (2\lambda_0 T - \lambda_1)^2 - 2\lambda_0 \geq 0, \\ \zeta^{'}(t) \leq 0, \ \forall t \in [0,T], & \text{if } (\lambda_1 - \lambda_0 T)^2 + \lambda_0^2 T^2 - 2\lambda_0 \leq 0, \\ \zeta^{'}(t) \geq 0 \text{ in } [0,\tilde{t}] \text{ and } \ \leq 0 \text{ in } [\tilde{t},T] \text{ for some } \tilde{t} \in (0,T), & \text{otherwise.} \end{cases}$$

The above properties of $\zeta^{'}(t)$ immediately imply the desired monotonicity of the variance of the optimal control as stated in the theorem.

For $g_a(t)$, we also apply Theorem 2.3.2 to get

$$\begin{aligned} g_a^{'}(t) &= -\frac{\rho^2}{2}f(t) + \frac{\lambda_a(t)}{2}\log\frac{\sigma^2 f(t)}{2\pi\lambda_a(t)} \\ &= -\frac{\rho^2}{2}f(t) - \frac{f^{'}(t)}{2}\log f(t) - \frac{f^{'}(t)}{2}\log\frac{\sigma^2}{2\pi} - \frac{\lambda_a(t)}{2}\log\lambda_a(t). \end{aligned}$$

Then the expression for $g_a(t)$ follows from the facts that

$$\int f(t)\log f(t)\,dt = f(t)\log f(t) - f(t) + C$$

and

$$\int \lambda_a(t)\log\lambda_a(t)\,dt = -\frac{\lambda_a^2(t)}{4\lambda_0}\log\lambda_a(t) + \frac{\lambda_a^2(t)}{8\lambda_0} + C.$$

□

A.1.4 Proof of Theorem 2.5.1

Since π is admissible, we have, for $\forall\ (t,x) \in [0,T] \times \mathbb{R}_+$,

$$V_t^\pi(t,x) + \int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(t,x) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(t,x) - \lambda \log \pi_t(u) \right) \pi_t(u)\, du = 0.$$

From the results in and after Theorem 2.3.1, the control $\tilde{\pi}$ satisfies

$$\begin{aligned} V_t^\pi(t,x) + &\int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(t,x) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(t,x) - \lambda \log \tilde{\pi}_t(u) \right) \tilde{\pi}_t(u)\, du \\ &= V_t^\pi(t,x) + \max_{\hat{\pi} \in \mathcal{P}(\mathbb{R})} \left\{ \int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(t,x) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(t,x) - \lambda \log \hat{\pi}_t(u) \right) \hat{\pi}_t(u)\, du \right\} \\ &\geq V_t^\pi(t,x) + \int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(t,x) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(t,x) - \lambda \log \pi_t(u) \right) \pi_t(u)\, du = 0. \end{aligned} \tag{A.2}$$

Let $\{X_t^{\tilde{\pi}},\ 0 \leq t \leq T\}$ denote the exploratory wealth process under the control $\tilde{\pi}$. For a fixed pair $(t,x) \in [0,T] \times \mathbb{R}_+$ and $n \geq 1$, define stopping times

$$\tau_n := \inf \left\{ s \geq t : \int_t^s \sigma^2 \int_{\mathbb{R}} u^2 \tilde{\pi}_v du (V_x^\pi(v, X_v^{\tilde{\pi}}))^2\, dv \geq n \right\}, \quad n = 1, 2, \cdots$$

and apply Itô's lemma to obtain, for $s \in [t,T]$,

$$\begin{aligned} V^\pi(s \wedge \tau_n, X_{s\wedge\tau_n}^{\tilde{\pi}}) = &V^\pi(t,x) + \int_t^{s\wedge\tau_n} V_t^\pi(v, X_v^{\tilde{\pi}})\, dv \\ &+ \int_t^{s\wedge\tau_n} \int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(v, X_v^{\tilde{\pi}}) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(v, X_v^{\tilde{\pi}}) \right) \tilde{\pi}_v(u)\, du\, dv \\ &+ \int_t^{s\wedge\tau_n} \sigma \sqrt{\int_{\mathbb{R}} u^2 \tilde{\pi}_v du} \cdot V_x^\pi(v, X_v^{\tilde{\pi}})\, dW_v. \end{aligned}$$

Rearranging the above equation and applying the inequality in (A.2), we get

$$\begin{aligned} V^\pi(t,x) = &\mathbb{E}\Bigg[V^\pi(s\wedge\tau_n, X_{s\wedge\tau_n}^{\tilde{\pi}}) - \int_t^{s\wedge\tau_n} V_t^\pi(v, X_v^{\tilde{\pi}})\, dv \\ &\quad - \int_t^{s\wedge\tau_n} \int_{\mathbb{R}} \left(\rho\sigma u V_x^\pi(v, X_v^{\tilde{\pi}}) + \frac{1}{2}\sigma^2 u^2 V_{xx}^\pi(v, X_v^{\tilde{\pi}}) \right) \tilde{\pi}_v(u)\, du\, dv \,\Bigg|\, X_t^{\tilde{\pi}} = x \Bigg] \\ \leq &\mathbb{E}\left[V^\pi(s\wedge\tau_n, X_{s\wedge\tau_n}^{\tilde{\pi}}) - \int_t^{s\wedge\tau_n} \int_{\mathbb{R}} \lambda \tilde{\pi}_v(u) \log \tilde{\pi}_v(u)\, du\, dv \,\middle|\, X_t^{\tilde{\pi}} = x \right]. \end{aligned}$$

At time $s = T$, the above inequality holds since $s \in [t, T]$. As $n \to \infty$, $T \wedge \tau_n \to T$. By the Dominated Convergence Theorem, we have, as $n \to \infty$,

$$\begin{aligned}
V^{\pi}(t,x) \leq &\mathbb{E}\left[V^{\pi}(T \wedge \tau_n, X^{\tilde{\pi}}_{T \wedge \tau_n}) - \int_t^{T \wedge \tau_n} \int_{\mathbb{R}} \lambda \tilde{\pi}_v(u) \log \tilde{\pi}_v(u) \, du \, dv \;\middle|\; X^{\tilde{\pi}}_t = x \right] \\
=&\mathbb{E}\left[V^{\pi}(T, X^{\tilde{\pi}}_{T}) - \int_t^{T} \int_{\mathbb{R}} \lambda \tilde{\pi}_v(u) \log \tilde{\pi}_v(u) \, du \, dv \;\middle|\; X^{\tilde{\pi}}_t = x \right] \\
=&\mathbb{E}\left[\log X^{\tilde{\pi}}_{T} - \int_t^{T} \int_{\mathbb{R}} \lambda \tilde{\pi}_v(u) \log \tilde{\pi}_v(u) \, du \, dv \;\middle|\; X^{\tilde{\pi}}_t = x \right] \\
=&\mathbb{E}\left[V^{\tilde{\pi}}(T, X^{\tilde{\pi}}_{T}) - \int_t^{T} \int_{\mathbb{R}} \lambda \tilde{\pi}_v(u) \log \tilde{\pi}_v(u) \, du \, dv \;\middle|\; X^{\tilde{\pi}}_t = x \right] \\
=&V^{\tilde{\pi}}(t,x).
\end{aligned}$$

A.2 RL Algorithms

A.2.1 RL Algorithm with Portion Control

For the exploratory problem controlling the investment portion, we parametrize the value function as

$$V^{\pi}(t, x; \boldsymbol{\alpha}) = \log x + \alpha(T - t).$$

We also have the mean of the Gaussian control parametrized as $\beta_1 = \frac{\rho}{\sigma}$ and

$$-\int_{\mathbb{R}} \pi_t(u; \boldsymbol{\beta}) \log \pi_t(u; \boldsymbol{\beta}) du = -\beta_2 = \frac{1}{2} \log \frac{2\pi e \lambda}{\sigma^2}.$$

The updating scheme for β_1 goes as follows:

$$\beta_1 \leftarrow \left(\frac{2\alpha + 2\lambda\beta_2 + \lambda}{4\pi\lambda\beta_1} e^{-2\beta_2 - 1} + \frac{\beta_1}{2} \right). \tag{A.3}$$

The gradients of the TD error in α and β_2 at time t_i are

$$\frac{\partial C_i}{\partial \alpha} = \sum_{(t_j, x_j) \in \mathcal{S}_i} \left(\dot{V}^{\pi}(t_j, x_j; \boldsymbol{\alpha}) - \lambda\beta_2 \right) (-\Delta t) \tag{A.4}$$

$$\frac{\partial C_i}{\partial \beta_2} = \sum_{(t_j, x_j) \in \mathcal{S}_i} \left(\dot{V}^{\pi}(t_j, x_j; \boldsymbol{\alpha}) - \lambda\beta_2 \right) (-\lambda\Delta t). \tag{A.5}$$

The algorithm is summarized by the pseudocode in Algorithm 3.

Algorithm 3: RL Algorithm with Portion Control

```
Input: Market parameters $(\mu, \sigma, r, \rho)$, learning rate $\theta_\alpha$, $\theta_\beta$, initial wealth $x_0$,
  investment horizon $T$, discretization $\Delta t$, exploration rate $\lambda$.
Initialization: $i = 1$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
while $i \leq \frac{T}{\Delta t}$ do
    Sample $(t_i, x_i)$ under $\pi(u; \boldsymbol{\beta})$
    Update set of samples $\mathcal{S}_i = \{(t_j, x_j); j = 0, \ldots, i\}$
    Update $\alpha$ as $\alpha - \theta_\alpha \nabla_\alpha C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (A.4)
    Update $\beta_2$ as $\beta_2 - \theta_\beta \nabla_\beta C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (A.5)
    Update $\beta_1$ using (A.3)
    Update $\pi_t(u; x, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as $\mathcal{N}\left(\beta_1, \ \frac{e^{-2\beta_2 - 1}}{2\pi}\right)$
    $i = i + 1$
end
```

A.2.2 RL Algorithm with Power-Decaying λ

For the exploratory problem with a power-decaying λ and $\lambda_0 \neq 1$, we parametrize the value function as

$$V^\pi(t, x; \boldsymbol{\alpha}) = \left(\frac{T + \lambda_1}{t + \lambda_1}\right)^{\lambda_0} \log x + \alpha_1 \frac{\left(\frac{T+\lambda_1}{t+\lambda_1}\right)^{\lambda_0 - 1} - 1}{\lambda_0 - 1} + \alpha_2 \left(\frac{T + \lambda_1}{t + \lambda_1}\right)^{\lambda_0}$$
$$- \frac{1}{2}\left(\frac{T + \lambda_1}{t + \lambda_1}\right)^{\lambda_0} \log\left(t + \lambda_1\right) + \alpha_3.$$

We also have the mean of the Gaussian control parametrized as $\beta_1 = \frac{\rho}{\sigma}$ and

$$-\int_{\mathbb{R}} \pi_t(u; \boldsymbol{\beta}) \log \pi_t(u; \boldsymbol{\beta}) du = \log x - \frac{1}{2} \log\left(t + \lambda_1\right) - \beta_2$$

where $\beta_2 = -\frac{1}{2} \log \frac{2\pi e \lambda_0}{\sigma^2}$. The updating scheme for β_1 goes as follows:

$$\beta_1 \leftarrow \left(\frac{\alpha_1 e^{-2\beta_2 - 1}}{2\pi \lambda_0 (T + \lambda_1) \beta_1} + \frac{\beta_1}{2}\right). \tag{A.6}$$

The updating scheme for α_3 also applies the terminal condition and goes as follows:

$$\alpha_3 \leftarrow \left(-\alpha_2 + \frac{1}{2} \log\left(T + \lambda_1\right)\right). \tag{A.7}$$

The gradients of the TD error in α_1, α_2 and β_2 at time t_i are

$$\frac{\partial C_i}{\partial \alpha_1} = \sum_{(t_j,x_j)\in\mathcal{S}_i} \left(\dot{V}^\pi(t_j,x_j;\boldsymbol{\alpha}) - \lambda(t_{j+1})\beta_2\right) \frac{\left(\frac{T+\lambda_1}{t_{j+1}+\lambda_1}\right)^{\lambda_0-1} - \left(\frac{T+\lambda_1}{t_j+\lambda_1}\right)^{\lambda_0-1}}{\lambda_0-1} \tag{A.8}$$

$$\frac{\partial C_i}{\partial \alpha_2} = \sum_{(t_j,x_j)\in\mathcal{S}_i} \left(\dot{V}^\pi(t_j,x_j;\boldsymbol{\alpha}) - \lambda(t_{j+1})\beta_2\right) \left(\left(\frac{T+\lambda_1}{t_{j+1}+\lambda_1}\right)^{\lambda_0} - \left(\frac{T+\lambda_1}{t_j+\lambda_1}\right)^{\lambda_0}\right) \tag{A.9}$$

$$\frac{\partial C_i}{\partial \beta_2} = \sum_{(t_j,x_j)\in\mathcal{S}_i} \left(\dot{V}^\pi(t_j,x_j;\boldsymbol{\alpha}) - \lambda(t_{j+1})\beta_2\right)(-\lambda(t_{j+1})\Delta t). \tag{A.10}$$

The algorithm is summarized by the pseudocode in Algorithm 4.

Algorithm 4: RL Algorithm with Power-Decaying λ

```
Input: Market parameters $(\mu, \sigma, r, \rho)$, learning rate $\theta_\alpha$, $\theta_\beta$, initial wealth $x_0$,
  investment horizon $T$, discretization $\Delta t$, exploration rates $\lambda_0$, $\lambda_1$.
Initialization: $i = 1$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
while $i \leq \frac{T}{\Delta t}$ do
    Sample $(t_i, x_i)$ under $\pi(u;\boldsymbol{\beta})$
    Update set of samples $\mathcal{S}_i = \{(t_j, x_j); j = 0, \ldots, i\}$
    Update $(\alpha_1, \alpha_2)'$ as $(\alpha_1, \alpha_2)' - \theta_\alpha \nabla_\alpha C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (A.8) and (A.9)
    Update $\alpha_3$ using (A.7)
    Update $\beta_2$ as $\beta_2 - \theta_\beta \nabla_\beta C_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ using (A.10)
    Update $\beta_1$ using (A.6)
    Update $\pi_t(u; x, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as $\mathcal{N}\left(\beta_1 x,\ \frac{e^{-2\beta_2-1}x^2}{2\pi(t+\lambda_1)}\right)$
    $i = i + 1$
end
```

Appendix B

Chapter 3 Appendix

B.1 Solving EMV via Dynamic Programming

Now solve the exploratory auxiliary problem $A^E(w)$ by dynamic programming. At time $T - \Delta t$, given $x_{T-\Delta t}$, the optimization problem is

$$
\begin{aligned}
&\max_{f_{T-\Delta t}} J_{T-\Delta t}(f_{T-\Delta t}|x_{T-\Delta t}) \\
&= \max_{f_{T-\Delta t}} \left\{ \mathbb{E}[-(w - x_T)^2] - \lambda \Delta t \int_{\mathbb{R}} \log\left(f_{T-\Delta t}(u)\right) f_{T-\Delta t}(u) du \right\} \\
&= \max_{f_{T-\Delta t}} \left\{ \mathbb{E}[-(w - x_{T-\Delta t} - R_{T-\Delta t} u_{T-\Delta t})^2] - \lambda \Delta t \int_{\mathbb{R}} \log\left(f_{T-\Delta t}(u)\right) f_{T-\Delta t}(u) du \right\} \\
&= \max_{f_{T-\Delta t}} \Bigg\{ - \mathbb{E}[R^2_{T-\Delta t}] \mathbb{E}[u^2_{T-\Delta t}] + 2(w - x_{T-\Delta t}) \mathbb{E}[R_{T-\Delta t}] \mathbb{E}[u_{T-\Delta t}] \\
&\qquad - \lambda \Delta t \int_{\mathbb{R}} \log\left(f_{T-\Delta t}(u)\right) f_{T-\Delta t}(u) du - (w - x_{T-\Delta t})^2 \Bigg\}.
\end{aligned}
$$

The maximization part over $f_{T-\Delta t}$ is

$$\begin{aligned}
&\max\Bigg\{-\mathbb{E}[R^2_{T-\Delta t}]\mathbb{E}[u^2_{T-\Delta t}]+2(w-x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]\mathbb{E}[u_{T-\Delta t}]\\
&\qquad-\lambda\Delta t\int_{\mathbb{R}}\log\left(f_{T-\Delta t}(u)\right)f_{T-\Delta t}(u)du\Bigg\}\\
&=\max\int_{\mathbb{R}}\left[-\mathbb{E}[R^2_{T-\Delta t}]u^2+2(w-x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]u-\lambda\Delta t\log\left(f_{T-\Delta t}(u)\right)\right]f_{T-\Delta t}(u)du.
\end{aligned}$$

A quick calculation gives us the optimal allocation distribution at time $T-\Delta t$

$$f^*_{T-\Delta t}(u)=\frac{1}{c_{T-\Delta t}}\exp\left\{-\frac{\mathbb{E}[R^2_{T-\Delta t}]}{\lambda\Delta t}\left(u-\frac{(w-x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]}{\mathbb{E}[R^2_{T-\Delta t}]}\right)^2\right\}$$

where $c_{T-\Delta t}>0$ is a constant such that $\int_{\mathbb{R}}f_{T-\Delta t}(u)du=1$. The distribution is a normal distribution with mean $\mu_{T-\Delta t}$ and variance $\sigma^2_{T-\Delta t}$ defined as follows

$$\mu_{T-\Delta t}=\frac{(w-x_{T-\Delta t})\mathbb{E}[R_{T-\Delta t}]}{\mathbb{E}[R^2_{T-\Delta t}]}=\frac{\rho}{\sigma(1+\rho^2\Delta t)}\left(w-x_{T-\Delta t}\right),$$

$$\sigma^2_{T-\Delta t}=\frac{\lambda\Delta t}{2\mathbb{E}[R^2_{T-\Delta t}]}=\frac{\lambda}{2\sigma^2(1+\rho^2\Delta t)}.$$

Appendix C

Chapter 4 Appendices

C.1 Estimation of the Semivariogram Function

The semivariogram function is an important element in both the conventional clustering procedure and our proposed one. It is used to model the covariance structure of the Gaussian process in the universal kriging method, and to obtain the conditional distributions of the value random variables. In practice, we need to estimate the parameters in the semivariogram function $\gamma(h)$, i.e. σ^2, a, and r. For our method with the Wasserstein distance, we need to estimate the parameters twice, when calculating the conditional distribution and applying the universal kriging method.

When we have selected representatives, we calculate the empirical semivariogram function with their values y,

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{(i,j)\in N(h)} (y_i - y_j)^2$$

where $N(h) = \left\{(i,j) : h - \frac{\Delta h}{2} \leq D(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq h + \frac{\Delta h}{2}\right\}$. The interval size Δh could be fixed, e.g. 0.1, or be set such that the number of intervals is fixed, e.g. 100 intervals.

Robust estimates of the parameters σ^2, a, and r could be obtained by minimizing the following loss function (Stanford and Vardeman (1994))

$$\mathcal{L}(\sigma^2, a, r) = \sum_{|N(h)|\geq 30} \left[\frac{\gamma(h) - \hat{\gamma}(h)}{\gamma(h)}\right]^2 |N(h)|.$$

Note that we only consider h such that $|N(h)| \geq 30$ to assure the reliability of the empirical function $\hat{\gamma}(h)$. We use the "NLoptr" package in R (Johnson (2022)) with the COBYLA (constrained optimization by linear approximations) algorithm (Powell (1994, 1998)) to estimate the parameters.

C.2 Proof of Lemma 4.3.2

For any $j \in \{k_1+1, k_1+2, \dots, k\}$ and i such that $z_i = j$, we have

$$\begin{aligned}(W(\boldsymbol{\theta}_i, \boldsymbol{c}_j^*))^2 &= ||\boldsymbol{\theta}_i - \boldsymbol{c}_j^*||^2 \\ &= ||\boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z}) + \alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^*||^2 \\ &= ||\boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z})||^2 + ||\alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^*||^2 + 2\langle \boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z}), \alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^* \rangle \\ &\geq ||\boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z})||^2 + 2\langle \boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z}), \alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^* \rangle \\ &= (W(\boldsymbol{\theta}_i, \alpha_j(\boldsymbol{z})))^2 + 2\langle \boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z}), \alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^* \rangle.\end{aligned}$$

Hence,

$$\begin{aligned}\sum_{z_i=j} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j^*))^2 &\geq \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \alpha_j(\boldsymbol{z})))^2 + 2\left\langle \sum_{z_i=j} (\boldsymbol{\theta}_i - \alpha_j(\boldsymbol{z})), \alpha_j(\boldsymbol{z}) - \boldsymbol{c}_j^* \right\rangle \\ &= \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \alpha_j(\boldsymbol{z})))^2.\end{aligned}$$

Therefore, we have

$$\begin{aligned}L(\boldsymbol{z}, \boldsymbol{c}^*) &= \sum_{j=1}^{k_1} \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j))^2 + \sum_{j=k_1+1}^{k} \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j^*))^2 \\ &\geq \sum_{j=1}^{k_1} \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \boldsymbol{c}_j))^2 + \sum_{j=k_1+1}^{k} \sum_{z_i=j} (W(\boldsymbol{\theta}_i, \alpha_j(\boldsymbol{z})))^2 = L(\boldsymbol{z}, \alpha(\boldsymbol{z})).\end{aligned}$$

C.3 Guarantee Types

This appendix explains the abbreviations of the guarantee types. They are also in Table 1 of Gan and Valdez (2017).

Abbreviation	Explanation
GMDB	Guaranteed minimum death benefit
GMAB	Guaranteed minimum accumulation benefit
GMIB	Guaranteed minimum income benefit
GMMB	Guaranteed minimum maturity benefit
GMWB	Guaranteed minimum withdrawal benefit
DBRP	GMDB with return of premium
DBRU	GMDB with annual roll-up
DBSU	GMDB with annual ratchet
ABRP	GMAB with return of premium
ABRU	GMAB with annual roll-up
ABSU	GMAB with annual ratchet
IBRP	GMIB with return of premium
IBRU	GMIB with annual roll-up
IBSU	GMIB with annual ratchet
MBRP	GMMB with return of premium
MBRU	GMMB with annual roll-up
MBSU	GMMB with annual ratchet
WBRP	GMWB with return of premium
WBRU	GMWB with annual roll-up
WBSU	GMWB with annual ratchet
DBAB	GMDB + GMAB with annual ratchet
DBIB	GMDB + GMIB with annual ratchet
DBMB	GMDB + GMMB with annual ratchet
DBWB	GMDB + GMWB with annual ratchet

Table C.1: Explanations of guarantee types

C.4 Tables of Study 1

This appendix provides more numerical results for the first study in Section 4.4.2.

Seed	k-prototypes	TP.SD	Seed	k-prototypes	TP.SD
1	0.0505	0.0144	51	0.0280	0.0002
2	0.0100	0.0225	52	0.0283	0.0611
3	0.0041	0.0010	53	0.0024	0.0153
4	0.0371	0.0473	54	0.0257	0.0202
5	0.0507	0.0113	55	0.0022	0.0283
6	0.0248	0.0411	56	0.0162	0.0030
7	0.0131	0.0212	57	0.0033	0.0084
8	0.0367	0.0020	58	0.0091	0.0083
9	0.0119	0.0039	59	0.0375	0.0017
10	0.0464	0.0171	60	0.0683	0.0089
11	0.0217	0.0155	61	0.0034	0.0036
12	0.0063	0.0236	62	0.0210	0.0088
13	0.0309	0.0185	63	0.0322	0.0083
14	0.0074	0.0361	64	0.0101	0.0060
15	0.0251	0.0408	65	0.0129	0.0099
16	0.0723	0.0332	66	0.0524	0.0179
17	0.0059	0.0096	67	0.0461	0.0240
18	0.0039	0.0339	68	0.0153	0.0191
19	0.0283	0.0089	69	0.0181	0.0196
20	0.0555	0.0144	70	0.0205	0.0002
21	0.0295	0.0212	71	0.0488	0.0209
22	0.0025	0.0252	72	0.0428	0.0387
23	0.0247	0.0106	73	0.0555	0.0419
24	0.0163	0.0254	74	0.0205	0.0209
25	0.0180	0.0002	75	0.0552	0.0027
26	0.0145	0.0050	76	0.0181	0.0498
27	0.0144	0.0359	77	0.0040	0.0168
28	0.0079	0.0195	78	0.0322	0.0299
29	0.0631	0.0016	79	0.0068	0.0149
30	0.0025	0.0120	80	0.0139	0.0132
31	0.0089	0.0075	81	0.0354	0.0376
32	0.0021	0.0275	82	0.0310	0.0064
33	0.0031	0.0110	83	0.0097	0.0224
34	0.0364	0.0146	84	0.0052	0.0017
35	0.0185	0.0007	85	0.0009	0.0403
36	0.0039	0.0029	86	0.0442	0.0076
37	0.0509	0.0205	87	0.0351	0.0304
38	0.0532	0.0016	88	0.0117	0.0073
39	0.0327	0.0114	89	0.0054	0.0281
40	0.0134	0.0205	90	0.0158	0.0073
41	0.0224	0.0056	91	0.0653	0.0302
42	0.0359	0.0076	92	0.0088	0.0134
43	0.0108	0.0296	93	0.0257	0.0542
44	0.0016	0.0101	94	0.0204	0.0128
45	0.0028	0.0116	95	0.0209	0.0170
46	0.0411	0.0045	96	0.0149	0.0002
47	0.0726	0.0437	97	0.0148	0.0315
48	0.0155	0.0202	98	0.0098	0.0386
49	0.0144	0.0350	99	0.0239	0.0067
50	0.0414	0.0481	100	0.0000	0.0158
Avg. Error	0.0237 (0.0019)	0.0184 (0.0014)	Avg. Time	2717.05s	3246.08s

Table C.2: Performance of methods: fair market value with $k = 150$. The values in parentheses are standard errors.

Seed	k-prototypes	TP.SD	Seed	k-prototypes	TP.SD
1	0.0430	0.0088	51	0.0109	0.0056
2	0.0163	0.0025	52	0.0024	0.0074
3	0.0496	0.0100	53	0.0169	0.0003
4	0.0242	0.0374	54	0.0424	0.0145
5	0.0102	0.0088	55	0.0040	0.0040
6	0.0033	0.0310	56	0.0095	0.0085
7	0.0442	0.0290	57	0.0260	0.0095
8	0.0208	0.0110	58	0.0207	0.0165
9	0.0154	0.0280	59	0.0057	0.0009
10	0.0182	0.0030	60	0.0080	0.0020
11	0.0238	0.0017	61	0.0218	0.0328
12	0.0017	0.0166	62	0.0158	0.0109
13	0.0234	0.0164	63	0.0181	0.0074
14	0.0297	0.0122	64	0.0096	0.0254
15	0.0108	0.0135	65	0.0222	0.0191
16	0.0242	0.0003	66	0.0418	0.0199
17	0.0580	0.0438	67	0.0014	0.0249
18	0.0470	0.0064	68	0.0244	0.0078
19	0.0134	0.0176	69	0.0016	0.0115
20	0.0123	0.0178	70	0.0001	0.0111
21	0.0489	0.0192	71	0.0219	0.0141
22	0.0220	0.0200	72	0.0043	0.0310
23	0.0057	0.0026	73	0.0108	0.0055
24	0.0254	0.0046	74	0.0023	0.0050
25	0.0219	0.0139	75	0.0414	0.0041
26	0.0103	0.0262	76	0.0496	0.0203
27	0.0151	0.0039	77	0.0220	0.0225
28	0.0054	0.0078	78	0.0151	0.0071
29	0.0049	0.0098	79	0.0796	0.0098
30	0.0031	0.0001	80	0.0263	0.0151
31	0.0135	0.0076	81	0.0029	0.0006
32	0.0037	0.0036	82	0.0104	0.0128
33	0.0066	0.0057	83	0.0633	0.0077
34	0.0143	0.0060	84	0.0070	0.0214
35	0.0423	0.0246	85	0.0122	0.0005
36	0.0104	0.0066	86	0.0096	0.0306
37	0.0310	0.0091	87	0.0372	0.0263
38	0.0012	0.0188	88	0.0177	0.0060
39	0.0298	0.0009	89	0.0071	0.0032
40	0.0079	0.0001	90	0.0143	0.0070
41	0.0261	0.0045	91	0.0298	0.0156
42	0.0150	0.0230	92	0.0454	0.0010
43	0.0299	0.0174	93	0.0369	0.0253
44	0.0138	0.0065	94	0.0440	0.0089
45	0.0180	0.0021	95	0.0202	0.0120
46	0.0087	0.0051	96	0.0070	0.0105
47	0.0034	0.0010	97	0.0064	0.0313
48	0.0052	0.0001	98	0.0271	0.0020
49	0.0137	0.0101	99	0.0283	0.0196
50	0.0361	0.0282	100	0.0194	0.0111
Avg. Error	0.0201 (0.0016)	0.0123 (0.0010)	Avg. Time	3501.16s	4011.43s

Table C.3: Performance of methods: fair market value with $k = 300$. The values in parentheses are standard errors.

Seed	k-prototypes	TP.SD	Seed	k-prototypes	TP.SD
1	0.0295	0.0080	51	0.0227	0.0114
2	0.0104	0.0209	52	0.0122	0.0028
3	0.0484	0.0300	53	0.0092	0.0001
4	0.0225	0.0002	54	0.0029	0.0196
5	0.0260	0.0071	55	0.0389	0.0114
6	0.0114	0.0002	56	0.0152	0.0004
7	0.0442	0.0066	57	0.0153	0.0069
8	0.0271	0.0059	58	0.0030	0.0103
9	0.0163	0.0049	59	0.0006	0.0058
10	0.0263	0.0084	60	0.0066	0.0038
11	0.0017	0.0118	61	0.0086	0.0048
12	0.0049	0.0042	62	0.0392	0.0125
13	0.0166	0.0029	63	0.0312	0.0006
14	0.0135	0.0014	64	0.0189	0.0005
15	0.0155	0.0188	65	0.0114	0.0264
16	0.0170	0.0117	66	0.0174	0.0014
17	0.0233	0.0080	67	0.0167	0.0061
18	0.0203	0.0065	68	0.0013	0.0177
19	0.0403	0.0005	69	0.0162	0.0196
20	0.0216	0.0229	70	0.0073	0.0118
21	0.0431	0.0148	71	0.0351	0.0009
22	0.0087	0.0074	72	0.0438	0.0054
23	0.0150	0.0073	73	0.0213	0.0021
24	0.0462	0.0015	74	0.0082	0.0040
25	0.0510	0.0018	75	0.0074	0.0094
26	0.0257	0.0013	76	0.0131	0.0166
27	0.0208	0.0261	77	0.0170	0.0131
28	0.0122	0.0000	78	0.0368	0.0212
29	0.0027	0.0004	79	0.0197	0.0033
30	0.0033	0.0138	80	0.0080	0.0204
31	0.0327	0.0220	81	0.0099	0.0018
32	0.0087	0.0048	82	0.0147	0.0122
33	0.0312	0.0135	83	0.0081	0.0071
34	0.0416	0.0183	84	0.0023	0.0053
35	0.0066	0.0048	85	0.0347	0.0123
36	0.0381	0.0146	86	0.0040	0.0039
37	0.0059	0.0096	87	0.0344	0.0084
38	0.0042	0.0068	88	0.0088	0.0043
39	0.0128	0.0127	89	0.0071	0.0161
40	0.0049	0.0079	90	0.0171	0.0083
41	0.0268	0.0117	91	0.0133	0.0029
42	0.0282	0.0137	92	0.0348	0.0083
43	0.0255	0.0066	93	0.0445	0.0258
44	0.0218	0.0041	94	0.0117	0.0110
45	0.0227	0.0068	95	0.0034	0.0069
46	0.0090	0.0120	96	0.0431	0.0025
47	0.0083	0.0113	97	0.0047	0.0156
48	0.0287	0.0201	98	0.0122	0.0047
49	0.0035	0.0047	99	0.0216	0.0046
50	0.0229	0.0020	100	0.0341	0.0055
Avg. Error	0.0192 (0.0013)	0.0090 (0.0007)	Avg. Time	5657.59s	6155.04s

Table C.4: Performance of methods: fair market value with $k = 450$. The values in parentheses are standard errors.

Appendix D

Chapter 5 Appendix

D.1 Proof of the Asymptotic Distributions of Capture Ratio Estimators

We prove the asymptotic distribution of $\tilde{R}_N$ in the independence case. From (5.8),

$$\tilde{R}_N = \phi\left(C_N \bar{Z}_N\right) \cdot \mathbb{1}_{S_N^C} + \xi \mathbb{1}_{S_N}$$

We claim that $\mathbb{1}_{S_N} \xrightarrow{P} 0$ or equivalently $\mathbb{1}_{S_N^C} \xrightarrow{P} 1$. Indeed, for any $\epsilon > 0$,

$$\begin{aligned}
\mathbb{P}(|\mathbb{1}_{S_N} - 0| \geq \epsilon) \leq & \mathbb{P}(S_N) = \mathbb{P}(M_N^U = 0 \text{ or } M_N^D = 0) \\
\leq & \mathbb{P}(U_n = 0 \text{ for all } n = 1, \cdots, N) + \mathbb{P}(D_n = 0 \text{ for all } n = 1, \cdots, N) \\
= & (p^U)^N + (p^D)^N \\
\to & 0, \text{ as } N \to \infty.
\end{aligned}$$

For the first term in $\tilde{R}_N$, since we have the asymptotic distribution of $C_N \bar{Z}_N$, by applying the Delta method (Theorem 3.1, Van der Vaart (2000)), we have

$$\sqrt{N}(\phi(C_N \bar{Z}_N) - \phi(C\mu)) \xrightarrow{d} N\left(\mathbf{0}, \left(\nabla\phi\left(C\mu\right)\right)' C\Sigma C' \left(\nabla\phi\left(C\mu\right)\right)\right).$$

By Slutsky's Theorem, we have

$$\sqrt{N}(\tilde{R}_N - R) = \sqrt{N}(\phi(C_N \bar{Z}_N)\mathbb{1}_{S_N^C} - R \cdot 1) \xrightarrow{d} N\left(\mathbf{0}, \left(\nabla\phi\left(C\mu\right)\right)' C\Sigma C' \left(\nabla\phi\left(C\mu\right)\right)\right).$$