Imperial College London

Imperial College London

Department of Mathematics

Efficient Methods for Dynamical Initial Margin Modelling

Author: Jingxuan Dong (CID: 01992119)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2020-2021

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

First and foremost, I would like to thank my external supervisor Gordon Lee from UBS for his support throughout the preparation of this thesis. His expertise and continuous encouragement were crucial in helping me achieve the outcomes I did over those four months. I really appreciate the good luck I had in being able to benefit from the guidance of such a supervisor.

I would also like to thank my internal supervisor, Dr Thomas Cass, for his time, suggestions and supports throughout this study. The discussions I had with him was very enriching for me.

I am very grateful to all the lecturers in the MSc Mathematics and Finance at Imperial College London, for making the course content truly enjoyable and applicable.

Finally, no words can express how grateful I am to my parents, your unwavering love, support and encouragement have always helped me overcome challenges.

Abstract

According to the margin requirements of BCBS (Basel Committee on Banking Supervision) and IOSCO (International Organization of Securities Commissions), the initial margin is the 99% quantile of the net change in portfolio value over a time interval of $(t, t + \delta]$, taking into account all market information up to time t. Calculating the future initial margin based on this definition is a difficult problem to solve from a computing aspect because it needs the use of a nested Monte Carlo simulation. This paper focuses on the use of the polynomial regression, the kernel regression, and the Dynamically Controlled Kernel Estimation methods to predict future initial margins. It is also important to find a more accurate and effective method among these methods.

Contents

List of Figures

List of Tables

Introduction

When a bank trades with a counterparty, the counterparty has a certain degree of probability of default at some uncertain point in the future. The risk assumed by the bank in this investment refers to the exposure, which represents the amount of money the bank may lose. To reduce the risk, counterparties provide collateral, which includes variation margin and initial margin. The variation margin is a type of margin payment that varies based on a number of factors. Unlike the variation margin, which is focused on reducing current risk, the initial margin is concerned with future close-out risk. The initial margin is calculated as the worst-case risk between the counterparty default date and settlement. In addition, both variation margin and initial margin need to be recalculated on a daily or periodic basis.

From the basis financial definition of the initial margin, the initial margin is the expectation of the initial margin at time t on all random path ω, conditional on all market information at time t. According to the margin requirements of BCBS (Basel Committee on Banking Supervision) and IOSCO (International Organization of Securities Commissions), the initial margin is the 99% quantile of the net change in portfolio value over a time interval of $(t, t+\delta]$, taking into account all market information up to t. This period δ is the period between the counterparty's default date and close-out, so it is called the close-out period. Usually δ is set to a period of 10 days. Define $V(\omega, t)$ to represent the value of the portfolio in case t, and the formula for the initial margin in case t for ω is

$$P\Big(V(\omega, t+\delta) - V(\omega, t) \leq IM(\omega, t) \mid \mathcal{F}_t\Big) = 99\%$$
$$\Rightarrow IM(\omega, t) = Q_{99\%}\Big(V(\omega, t+\delta) - V(\omega, t) \mid \mathcal{F}_t\Big)$$

where $Q_q(\cdot)$ is a $q\%$ quantile function. Define $\Delta(\omega, t; \delta) = V(\omega, t+\delta) - V(\omega, t)$ is the changes in the netting set's value on the scenario ω, and then the formula of the initial margin at time t on scenario ω is

$$IM(\omega, t) = Q_{99\%}\Big(V(\omega, t+\delta) - V(\omega, t) \mid \mathcal{F}_t\Big) = Q_{99\%}\Big(\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big) \tag{0.0.1}$$

From the equation (0.0.1), there are two approaches to calculate the initial margin at time t. Since the only unknown in the equation is the portfolio price at time $t+\delta$, the simplest method is to generate a list of portfolio prices at time $t+\delta$ and then apply the quantile function to the change of the portfolio prices to determine the value. The other approach is to assign a distribution to the difference between the portfolio prices at times t and $t+\delta$. The initial margin at time t can be calculated based on the characteristic of the specified distribution. In addition, we can choose to use the historical data for the portfolio price at time t and its corresponding initial margin to estimate the initial margin at time t for an unseen portfolio price.

This paper shows how to use different methods to calculate the initial margin at time t and investigate the strengths and limitations of these methods from the comparison. The purpose is to find a method with a highly accurate rate and great efficiency among all these methods.

In the whole thesis, we assume that the portfolio price follows the Black-Scholes model, and then the asset price follows a geometric Brownian motion (GBM). Due to the Black-Scholes model and the geometric Brownian motion, the initial margin at time t has a closed-form solution. This solution is called the benchmark, and we use this value to verify the results from other methods. It is the way to examine the accuracy of the methods. Chapter 1 focuses on the nested Monte Carlo method, also known as the brute-force method. In chapters 2,3 and 4, we explore how to use the polynomial regression, the kernel regression and the Dynamically Controlled Kernel Estimation

with parametric and non-parametric approaches to predict the initial margin at time t respectively. The last chapter contains an overall comparison of all methods that are mentioned in the previous chapters for calculating the initial margin at time t of a European Call option.

Chapter 1

Benchmark and Brute-force Methods

In this chapter, we use the nested Monte Carlo method to calculate the initial margin on a European Call option under the Black-Scholes model with the stock price follows a geometric Brownian motion. Also, under this background, there is a closed-form solution for the initial margin at time t.

1.1 Background

Before presenting the benchmark and nested Monte Carlo methods, it is necessary to describe the methodology used to generate the portfolio prices. Consider that portfolio prices follow the Black-Scholes model, while stock prices follow the geometric Brownian motion. In this section, we explain in detail the geometric Brownian motion and the Black-Scholes model.

1.1.1 Geometric Brownian Motion

The geometric Brownian motion is the solution of a stochastic differential equation (SDE) with linear drift and diffusion coefficients, and the formula is

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

where W_t is a Wiener process or Browian motion, and drift μ and the volatility σ are constant. The solution of the geometric Brownian motion using Itô's interpretation is

$$S_{t_2} = S_{t_1}\ exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t_2 - t_1) + \sigma\sqrt{t_2 - t_1}\theta\right) \tag{1.1.1}$$

where θ is random drawn from a standard Normal distribution.

1.1.2 The Black-Scholes Model

The Black-Scholes model is a mathematical method to calculate the option price. Consider the stock price $S(t)$ follows a geometric Brownian motion, r is the risk-free insterest rate, σ is the volatility, T is the time of maturity, K is the strike. As Black and Scholes [1] proved, the European option prices at time $t \in [0, T]$ under the Black-Scholes model are

$$C^{BS}(S_t, t; r, \sigma, T, K) = e^{-r(T-t)}\left[S_t\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2)\right] \tag{1.1.2}$$

$$P^{BS}(S_t, t; r, \sigma, T, K) = e^{-r(T-t)}\left[Ke^{-r(T-t)}\Phi(d_2) - S_t\Phi(d_1)\right] \tag{1.1.3}$$

where

$$d_1 = \frac{\log(S_t/K) + (r + \sigma^2/2)(T-t)}{\sigma(T-t)}$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

1.2 Parametric and Non-parametric Approaches

There are two approaches to calculate the initial margin at time t.

1.2.1 Parametric Approach

According to Cipolina et al. [2], assume that the change on the portfolio price at times t and $t+\delta$ follows a local Normal distribution with mean μ_V and variance σ_V, which is shown as

$$V(t+\delta) - V(t) \mid \mathcal{F}_t \sim \mathcal{N}(\mu_V, \sigma_V)$$

Recall that the formula of initial margin

$$IM(\omega, t) = Q_{99\%}\Big(V(\omega, t+\delta) - V(\omega, t) \mid \mathcal{F}_t\Big)$$

, where $\Delta(\omega, t; \delta) = V(\omega, t+\delta) - V(\omega, t)$. Since the quantile function of a distribution is the inverse of the cumulative distribution function, $IM(\omega, t)$ can be re-written as

$$\begin{aligned} IM(\omega, t) &= Q_{99\%}\Big(\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big) \\ &= \mathbb{E}\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big] + \sqrt{\mathbb{V}ar\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big]} \times \Phi^{-1}\big(99\%\big) \end{aligned}$$

where

$$\mathbb{V}ar\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big] = \mathbb{E}\Big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\Big] - \Big[\mathbb{E}\big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\big]\Big]^2$$

Without loss of generalization, we assume that the expectation of this local Normal distribution is zero. Then

$$\begin{aligned} \mathbb{E}\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big] &= 0 \\ \mathbb{V}ar\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big] &= \mathbb{E}\Big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\Big] \end{aligned}$$

Define

$$\sigma(\omega, t) \coloneqq \sqrt{\mathbb{V}ar\Big[\Delta(\omega, t; \delta) \mid \mathcal{F}_t\Big]} = \sqrt{\mathbb{E}\Big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\Big]}$$

. Thus, the equation (0.0.1) becomes

$$IM(\omega, t) = \sigma(\omega, t) \times \Phi^{-1}\big(99\%\big) \tag{1.2.1}$$

Since $\Phi^{-1}\big(99\%\big)$ is a constant, the equation (1.2.1) shows that the initial margin at time t for scenario ω can be calculated if $\sigma(\omega, t) = \sqrt{\mathbb{E}\Big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\Big]}$ can be solved.

1.2.2 Non-parametric Approach

For a non-parametric method, we use the historical data or the given samples to interpolate the result for the unseen data. More specifically, for every portfolio price at time t on scenario ω, we can determine the corresponding initial margin $IM(\omega, t)$. Then, using the model that is trained based on this data to predict the unsampled data.

1.3 Benchmark

As discussed in the previous chapter, the initial margin is the 99% quantile of the net change in the portfolio value over the time interval $(t, t+\delta]$. In other words, we are calculating the value of the portfolio at time t and $t+\delta$ as $V(t)$ and $V(t+\delta)$, respectively. Due to the Black-Scholes model, when the strike price, volatility and interest rate are fixed, the option price depends on the

stock price and the time to expiration. Therefore, in order to measure the option prices $V(t)$ and $V(t+\delta)$, the stock prices $S(t)$ and $S(t+\delta)$ need to be calculated first.

The equation (1.1.1) shows two important facts. First, the stock price $S(t+\delta)$ depends entirely on the previous stock price $S(t)$ and the random number θ. Second, θ is a random number in a normal distribution, which means that this value can be set on demand. The second fact contributes significantly to the calculation of the initial margin, which will be explained in detail later.

Based on the characteristics of the Black-Scholes model and the geometric Brownian motion, there is a closed-form solution for the initial margin as a benchmark. Recall that the initial margin is the 99% quantile of the change on the portfolio value over a period $(t, t+\delta]$. To be specific, for the fixed portfolio value $V(t)$, the initial margin is the difference between the portfolio value $V(t)$ and the portfolio value $V(t+\delta)$ on the 99% quantile. Since the stock price and the portfolio value are in one-to-one correspondence, the portfolio value on the 99% quantile represents the stock value on the 99% quantile. A stock price $S(t+\delta)$ on the 99% quantile can be calculated based on the previous stock price $S(t)$ and a 99% quantile of a standard Normal distribution θ, which can be expressed as

$$S_{t+\delta} = S_t \; exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\delta + \sigma\sqrt{\delta}\;\Phi^{-1}(0.99)\right) \tag{1.3.1}$$

where Φ is a cumulative density function of a standard Normal distribution. Then, the corresponding option price $V(t+\delta)$ can be calculated based on the stock price $S(t+\delta)$, and the stock price $S(t+\delta)$ can be calculated based on the previous stock price $S(t)$.

To summarize, the benchmark method is to calculate the stock price $S(t+\delta)$ on the 99% quantile and the corresponding option price $V(t+\delta)$ via the Black-Scholes pricing formula for each stock price $S(t)$. Then, the initial margin at time t for the stock price $S(t)$ is the difference between the option prices $V(t)$ and $V(t+\delta)$.

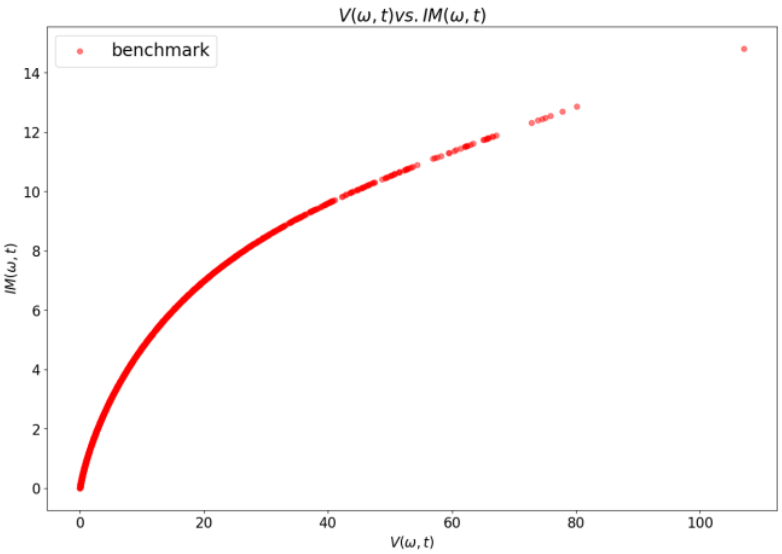


Figure 1.1: $V(\omega, t)$ vs. $IM(\omega, t)$ using the benchmark method

Figure 1.1 shows the initial margin at time t for every possible portfolio price $V(\omega, t)$ via the benchmark method. It exists a non-linear relationship between $V(\omega, t)$ and $IM(\omega, t)$ because the line has a concave trend. Therefore, we can apply polynomial regression to predict the initial margin at time t using the portfolio price as the regressor. This model will be explained in the next chapter.

1.4 Nested Monte Carlo Method

The most straightforward method is a nested Monte Carlo simulation method, also known as the brute-force method. Recall that the initial margin at time t is equal to the 99% quantile on the probability density function (PDF) distribution of the difference between the option price at time t and $t+\delta$. At the time t, the option price at time $t+\delta$ can be generated using Monte Carlo simulation with a given option price $V(t)$. However, the option price at time t is based on the previous option price. It means that the option price at time t is also generated via a Monte Carlo simulation depending on the previous option price. Mostly, the option price at the time t is simulated depending on the initial option price because the option prices between the time $t_0 = 0$ and t are unimportant in the process of the initial margin calculation. Thus, for calculating the initial margin at time t, a nested Monte Carlo simulation is needed.

Usually, one option price at time t can generate more than one possible price at time $t+\delta$. For the convergence of the value of the initial margin, the method needs to simulate a satisfactorily large number of scenarios at the time t and $t+\delta$ respectively. Moreover, since the initial margin focus on the value at the tail of the distribution of $\Delta(\omega, t; \delta_{IM})$, the scenarios at time $t+\delta$ should be a satisfactorily large number, which is larger than the scenarios at time t.

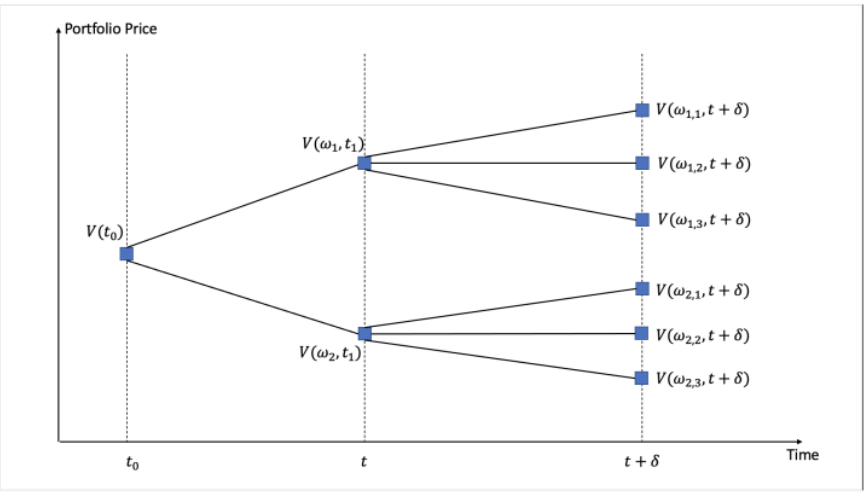


Figure 1.2: A example of the nested Monte Carlo simulation

Figure 1.2 shows a simple example of the brute-force method. For $V(t_0)$, simulating two paths at time t, and then simulating three paths at time $t+\delta$ for each $V(\omega, t)$. In this example, $IM(t) = Q_{99\%}\big(V(\omega_{i,j}, t+\delta) - V(\omega_i, t)\big)$, where $i = 1, 2$ and $j = 1, 2, 3$. Clearly, the value of the initial margin at time t maybe not converge because of the limitation of the scenarios.

The nested Monte Carlo method ought to generate an efficient result when the scenarios are large enough at two Monte Carlo simulations. However, due to a large number of the simulation at both times t and $t+\delta$, the run-time of the method will be severely affected.

There is one problem when implementing this method. The problem is that the current option price does not depend on the past option price when the option price follows the Black-Scholes model. Notice that this dependence relationship between the continuous option prices is really important because the initial margin works on the difference between the option prices at two continuous-time points t and $t+\delta$. That is the two option prices need to be related. The solution is to use the stock price efficiently. Based on the assumption, the stock price follows a geometric Brownian motion, which is a continuous-time stochastic process. From the equation (1.1.1), the stock price at time $t+\delta$ completely depends on the stock price at time t and the period δ when all other parameters remain unchanged. In addition, the option price can be determined by a unique combination of stock price $S(t)$ and time t with all other parameters are constant.

From the above analysis, instead of applying the Monte Carlo simulation for the option price at time t and $t+\delta$ separately, the brute-force method is to use the solution of the geometric Brownian

motion to generate several different stock prices at time t and $t+\delta$ and to use the Black-Scholes formula to calculate the corresponding option prices. Then, the initial margin is the 99% quantile of the differences between the option price at time t and $t+\delta$. The detailed steps are:

1. Given S_0, t, $t+\delta$, and the type of the option.
2. Using the formula $S_t = S_0\ exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\sqrt{t}\ \theta\right)$, where θ is a random drawn from a standard Normal distribution, to simulate N scenarios of stock prices $S(\omega = i, t)$ where $i = 1, 2, \ldots, N$.
3. Using the formula $S_{t+\delta} = S_t\ exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\delta + \sigma\sqrt{\delta}\ \theta\right)$, where θ is a random drawn from a standard Normal distribution, to simulate M scenarios of stock prices $S(\omega = i.j, t)$ where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$.
4. Calculating the option price $V(\omega = i, t)$ and $V(\omega = i.j, t)$ where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ using the equation (1.1.2) or (1.1.3) depending on the type of the option.
5. For each scenario ω, calculating the difference between the option prices at time t and $t+\delta$, and the initial margin $IM(\omega, t) = Q_{99\%}\big(\Delta(\omega, t; \delta)\big)$ where $\Delta(\omega, t; \delta) = V(\omega, t+\delta) - V(\omega, t)$.

The next step is to choose the number of N and M. For having a converged result of the initial margin, the scenarios between the time t and $t+\delta$ need to be a sufficiently large number. For example, if N is fixed to be 1000, setting $M = 10, 100, 1000, 10000$ separately to compare the results in different fields. For ensuring the comparison under the fair environment, we generate 1000 different stock prices at time t, named $S(\omega, t)$, and use these stock prices to estimate the initial margin at the time t. Except for the initial margin, the intermediate results during the process are also meaningful. For example, the initial margin at time t corresponding to the different stock prices at time t, the mean squared error between the initial margin at time t using the two methods, and the run-time of the methods.

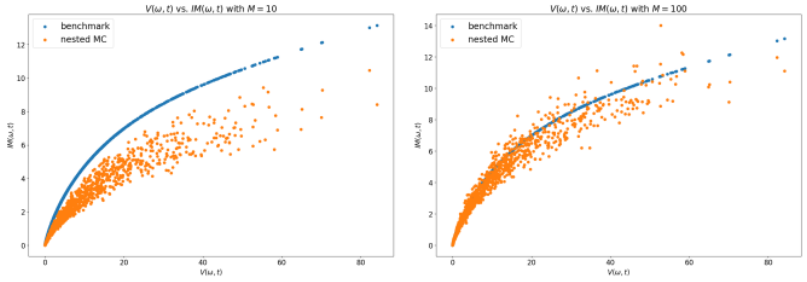
Method	MSE	Run-time
brute-force method with M=10	3.36491	2.1
brute-force method with M=100	0.43689	18.6
brute-force method with M=1000	0.05117	175.1
brute-force method with M=10000	0.00542	1908.6

Table 1.1: The initial margin at time t under the benchmark method and the brute-force method at time

Figure 1.3 shows that the comparison between the initial margin $IM(\omega, t)$ via the benchmark method and the nested Monte Carlo methods wth $N = 1000$ and $M = 10, 100, 1000, 10000$. The figure shows that the portfolio prices at time t ($V(\omega, t)$) against the corresponding initial margin ($IM(\omega, t)$). The nested-Monte-Carlo-based initial margin $IM(\omega, t)$ (orange dots) is closer to the benchmark results (blue dots) with the larger M, and The distribution of orange dots shrinks as M increases. Thus, In the bottom right corner of the graph, the nested Monte Carlo method's results almost completely cover the benchmark result.

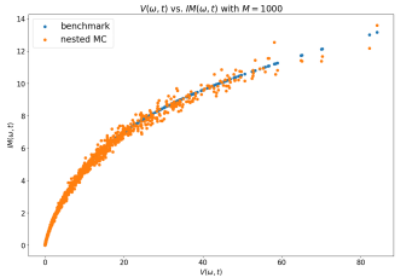
Table 1.1 and Figure 1.4 show the mean squared error between the benchmark and nested Monte Carlo methods with $N = 1000$ and $M = 10, 100, 1000, 10000$ and the run-time for every nested Monte Carlo method. Both table and figure show that the mean squared error decreases steeply as M increases. Also, when M increases, the running time is much slower. It is clear that the estimated initial margins at time t under $M = 10000$ is very close to the benchmark values, but the run-time is extremely large. The better choice is $M = 100$ and $M = 1000$ because the mean squared error is negligible and the run-time is acceptable.

All in all, for the nested Monte Carlo method, when N is a sufficiently large number and $M \geq N$, the initial margin at time t is very close to the benchmark result, but this method is quite time-consuming.

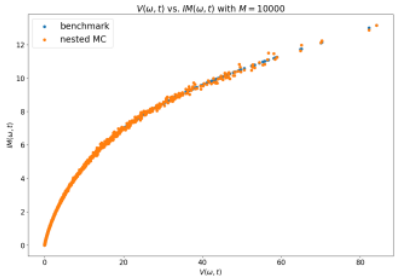


(a) M=10

(b) M=100

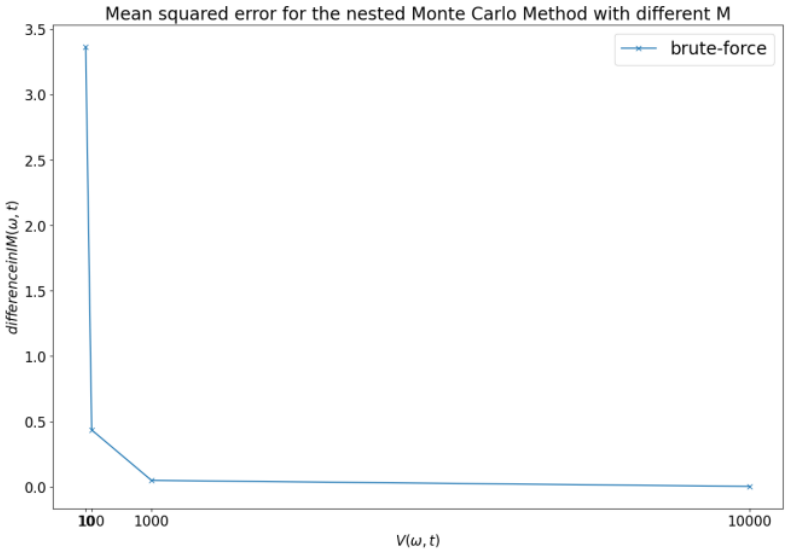


(c) M=1000



(d) M=10000

Figure 1.3: $IM(\omega, t)$ under the benchmark method and the nested Monte Carlo method over different $V(\omega, t)$ when setting $N = 1000$ and $M = 10, 100, 1000, 10000$.



(a) M=10

Figure 1.4: The mean squared error between the benchmark value and the predicted initial margin at time t using the nested Monte Carlo method over different $V(\omega, t)$ when setting $N = 1000$ and $M = 10, 100, 1000, 10000$.

Chapter 2

Polynomial Regression Method

From the benchmark method, we find a non-linear relationship between the portfolio price at time t and the corresponding initial margin. Therefore, this chapter focuses on the use of polynomial regression methods to calculate future initial margins for both parametric and non-parametric methods.

2.1 Methodology

Before applying polynomial regression to calculate the initial margin, we will introduce polynomial regression and its underlying logistic regression analysis in this section.

2.1.1 Regression Analysis

Regression analysis is a method to estimate the relationship between a dependent variable and at least one independent variable, and then use the unseen independent variables to predict the corresponding dependent variable. The general formula can be shown as:

$$\boldsymbol{Y} = f(\boldsymbol{X}, \boldsymbol{\beta}) + \boldsymbol{\epsilon}$$

where $\boldsymbol{Y}$ is a vector with one column, $\boldsymbol{X}$ is a vector or matrix depending on the number of the independent variables, $\boldsymbol{\beta}$ is the coefficients, and $\boldsymbol{\epsilon}$ is the error term. The main purpose of the regression analysis is to estimate $\boldsymbol{\beta}$, which can be used in the function $f(\boldsymbol{X}, \boldsymbol{\beta})$ for making a prediction Y of the response $\boldsymbol{X}$.

2.1.2 Polynomial Regression

Polynomial regression is a special form of regression analysis. In polynomial regression, the relationship between the independent variable X and the dependent variable Y is described as a polynomial of the n^{th} degree of X. To be precise, polynomial regression is used to fit the non-linear relationship between the value of X and the associated Y. The model has the form

$$Y = \sum_{i=0}^{P} \beta_i \phi_i(\boldsymbol{X})$$

where β is the coefficient that we want to estimate, and $\phi_i(\cdot)$ is the monomial function, Y is the dependent variable and X is the independent variable.

2.2 Implementation

Next, we will see how to predict the initial margin at time t using polynomial regression with parametric and non-parametric methods.

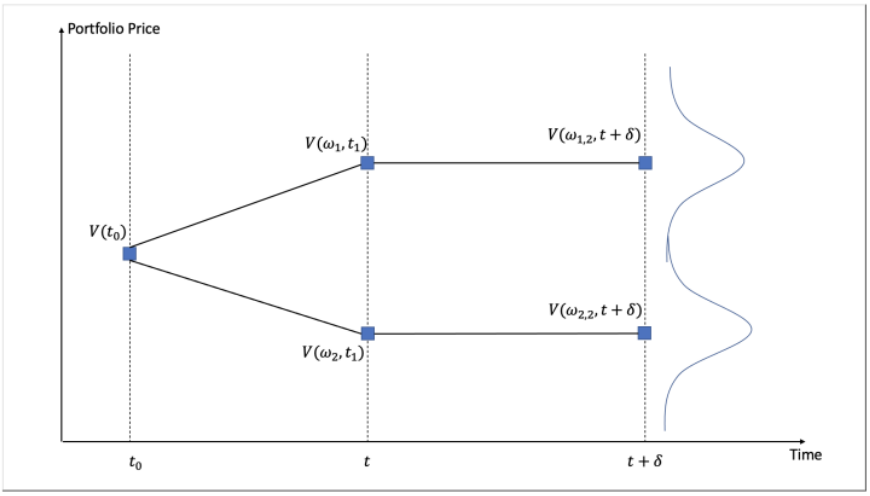


Figure 2.1: A simplification version when $\Delta(\omega, t; \delta)$ is assumed to follow a distribution

2.2.1 Parametric Approach

As mentioned at the beginning, the initial margin at time t is a 99% quantile of the changes in the netting set's value, also known as profit and loss (PnL), within a time interval $(t, t+\delta]$ given all information til the time t. More specifically, the process of calculating the initial margin at time t is to determine the changes in the option price between the time t and $t+\delta$ for all scenarios and take the 99% quantile on the values. So, we focus on estimate the initial margin at time t for each scenario, also written as $IM(\omega, t)$. In the nested Monte Carlo method, we apply the Monte Carlo simulation on on $V(t_0)$ for generating a list of $V(\omega, t)$ and on each $V(\omega, t)$ for generating a list of $V(\omega, t+\delta)$. This method predicts a relatively high accurate result with absolutely low efficiency.

For avoiding the nested Monte Carlo simulation, we assume a distribution for the changes between the portfolio price at time t and $t+\delta$. In other words, only one Monte Carlo simulation is needed to determine $V(\omega, t)$ and $V(\omega, t+\delta)$ on each path, and then the initial margin is calculated using the characteristics of the specified distribution. The visualization of the idea is shown in the figure 2.1.

Recall that

$$IM(\omega, t) = \sigma(\omega, t) \times \Phi^{-1}(99\%) = \sqrt{\mathbb{E}\big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\big]}$$

when we assume the distribution of $\big(V(\omega, t+\delta) - V(\omega, t)\big) \mid \mathcal{F}_t$ follows a local Normal distribution with zero-mean and variance $\sigma(\omega, t)^2$, which is presented as

$$\big(V(\omega, t+\delta) - V(\omega, t)\big) \mid \mathcal{F}_t \sim \mathcal{N}(0, \sigma(\omega, t))$$

Clearly, the only unknown in the formula for initial margin is

$$\sigma(\omega, t) = \sqrt{\mathbb{E}\big[\Delta(\omega, t; \delta)^2 \mid \mathcal{F}_t\big]} = \sqrt{\mathbb{E}\Big[\Big(V(\omega, t+\delta) - V(\omega, t)\Big)^2 \mid \mathcal{F}_t\Big]}$$

Thus, we can choose $(V(\omega, t+\delta) - V(\omega, t))^2$ as the response variable. The next important step is to determine the independent variable. As Cipolina et al. [2] mentioned, the regressor should satisfy two criteria: the regressor should be selected from the given market information, and there exists a relationship between the independent variable and the regressor. Since the initial margin for the scenario ω focuses on the change between the portfolio price at time t and $t+\delta$, the portfolio price $V(\omega, \cdot)$ is a better choice as the regressor. Moreover, the validity period of past information is a key factor that needs to be considered. The portfolio price is only related to the stock price that follows a Geometric Browain motion. Thus, the stock price depends on the most recent values, as well as the initial margin. As a result, the portfolio price at time t, $V(\omega, t)$, is considered as the regressor.

Define the independent variable

$$\boldsymbol{X}_t = \big[V(\omega_1, t), V(\omega_2, t), \ldots, V(\omega_N, t)\big]^T$$

, the dependent variable

$$Y_t = \Big[\big(V(\omega_1, t+\delta) - V(\omega_1, t)\big)^2, \big(V(\omega_2, t+\delta) - V(\omega_2, t)\big)^2, \ldots, \big(V(\omega_N, t+\delta) - V(\omega_N, t)\big)^2\Big]^T$$

and the basis function is $\phi_i(x) = x^i$. Thus, the form of the linear model is

$$\begin{aligned} & Y_t = \sum_{i=1}^{P} \boldsymbol{\beta}_i \phi_i(\boldsymbol{X}_t) \\ \Rightarrow & \begin{bmatrix} \big(V(\omega_1, t+\delta) - V(\omega_1, t)\big)^2 \\ \big(V(\omega_2, t+\delta) - V(\omega_2, t)\big)^2 \\ \vdots \\ \big(V(\omega_N, t+\delta) - V(\omega_N, t)\big)^2 \end{bmatrix} = \begin{bmatrix} 1 & V(\omega_1, t)^1 & \cdots & V(\omega_1, t)^P \\ 1 & V(\omega_2, t)^1 & \cdots & V(\omega_2, t)^P \\ \vdots & \vdots & \ddots & \vdots \\ 1 & V(\omega_N, t)^1 & \cdots & V(\omega_N, t)^P \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_P \end{bmatrix} \end{aligned}$$

When training the model, the training data and the highest power P of the monomial function are all needed. The response variable is easy to get, but the regressor needs to apply the monomial function to generate new columns before the training starts. After training the model, we use this model to predict $(V(\omega, t+\delta) - V(\omega, t))^2$ for the test data. Before applying the formula (1.2.1) to calculate the initial margin, the necessary step is to filter out the non-positive value in the predict list. The reason for doing this step is that we use the squared root of the response variable in the calculation, and the negative value cannot take the squared root. Finally, the initial margin at time t is the average of the initial margin at time t across all scenarios ω. The results are shown in the figure 2.2.

Method	MSE	Run-time	Method	MSE	Run-time
degree=1	15.85158	0.32	degree=11	1.98217	0.35
degree=2	15.88178	0.26	degree=12	5.16246	0.28
degree=3	15.58592	0.28	degree=13	5.74650	0.33
degree=4	15.58349	0.29	degree=14	5.77333	0.30
degree=5	16.20285	0.26	degree=15	6.32643	0.30
degree=6	0.17891	0.29	degree=16	6.79775	0.29
degree=7	0.19323	0.28	degree=17	7.17434	0.29
degree=8	16.68235	0.26	degree=18	7.46913	0.28
degree=9	0.51906	0.36	degree=19	7.71721	0.30
degree=10	1.14603	0.31	degree=20	8.45358	0.29

Table 2.1: The mean squared errors between the benchmark method and the polynomial regression methods with $degree = 1, 2, \ldots, 20$

Figure 2.2 shows the initial margin at time t on the scenario ω via the polynomial regression methods with $degree = 1, 2, \ldots, 20$. The figure at the top represents $degree = 1, 2, \ldots, 10$, and the one at the bottom represents $degree = 11, 12, \ldots, 20$. It is clear that when $degree < 9$, the predicted values from the polynomial regression methods almost cover the benchmark values. In other words, the shape of the predicted and baseline values are the same except for the tails of the predicted values. However, when $degree \geq 9$, the shape of the predicted value is significantly different from the benchmark value. Therefore, we cannot take the degree too large.

Table 2.1 shows the mean squared error between the predicted initial margin via the polynomial regression methods with $degree = 1, 2, \ldots, 20$ and the benchmark values. The table indicates that the polynomial regression method with $degree = 6$ performs the best because this method has the smallest mean squared error and the shape of the predicted initial margin is closer to the benchmark. Also, it proves that the mean squared error of the method increases as $degree \geq 9$, which means the shape of the values does not match the benchmark.

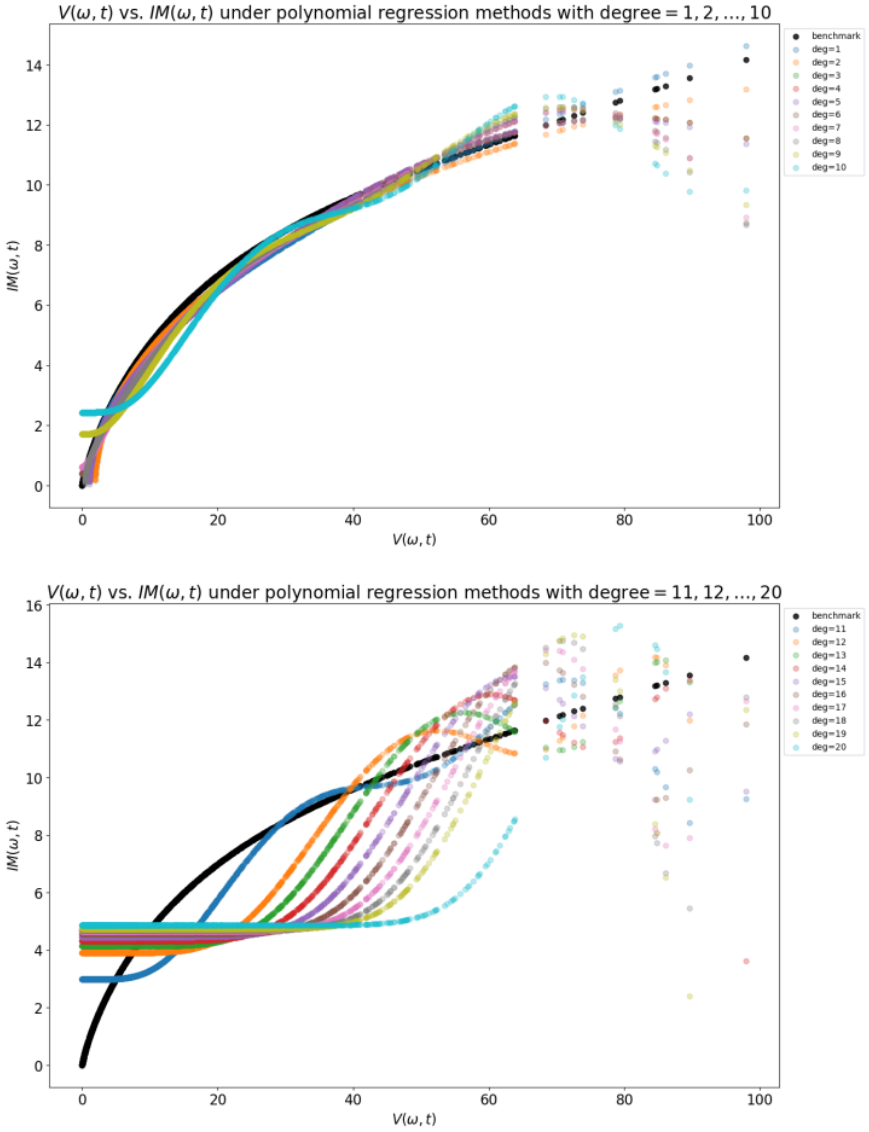


Figure 2.2: Polynomial regression methods with $degree = 1, 2, \ldots, 20$: $X_t = V(\omega, t)$ and $Y_t = \big(V(t+\delta) - V(t)\big)^2$

2.2.2 Non-parametric Approach

When calculating the initial margin, it actually calculates the change between the portfolio prices at times t and $t+\delta$ for every scenario. However, in the polynomial regression methods, the change between the portfolio prices at time t and $t+\delta$ equal to the square root of the response variable $\big(V(t+\delta)-V(t)\big)^2$. In other words, if the predicted $(V(\omega,t+\delta)-V(\omega,t))^2$ is a negative number, we cannot use the polynomial regression method to determine the corresponding initial margin. For avoiding this problem, it is possible to define another response variable for the model.

Instead of defining

$$Y_t = \Big[(V(\omega_1,t+\delta)-V(\omega_1,t))^2,(V(\omega_2,t+\delta)-V(\omega_2,t))^2,\ldots,(V(\omega_N,t+\delta)-V(\omega_N,t))^2\Big]^T$$

we define

$$Y_t = \Big[IM(\omega_1,t),IM(\omega_2,t),\ldots,IM(\omega_N,t)\Big]^T$$

When the stock price and the portfolio price at time t are given, using the solution of the geometric Brownian motion (1.3.1) to calculate the stock price at time $t+\delta$, and then using the Black-Scholes pricing formula (1.1.2) to calculate the corresponding portfolio price at time $t+\delta$. Recall that the initial margin is the 99% quantile of the change between the portfolio price at times t and $t+\delta$, conditional on all market information to time t. The initial margin at time t on each scenario ω is determined using the portfolio price at time t and $t+\delta$, which is defined as the response variable Y_t. In addition, the independent variable X_t is defined as usual. The model is trained using X_t and Y_t, and then using the model to predict initial margin at time t on scenario ω based on unseen $V(\omega,t)$.

Method	MSE	run-time	Method	MSE	Run-time
degree=1	1.04910	0.00	degree=11	2.23336	0.01
degree=2	0.25512	0.00	degree=12	4.62008	0.01
degree=3	0.07207	0.00	degree=13	5.28046	0.01
degree=4	0.02499	0.01	degree=14	5.85550	0.01
degree=5	0.00987	0.00	degree=15	6.36633	0.01
degree=6	0.00434	0.00	degree=16	6.79780	0.01
degree=7	0.00230	0.00	degree=17	7.14080	0.02
degree=8	0.00129	0.00	degree=18	7.40615	0.01
degree=9	0.61584	0.01	degree=19	7.60341	0.01
degree=10	1.36411	0.01	degree=20	8.15970	0.01

Table 2.2: The mean squared errors between the benchmark method and the polynomial regression methods with $degree = 1,2,\ldots,20$

Figure 2.3 shows the predicted initial margin at time t on scenario ω via the polynomial regression methods with $degree = 1,2,\ldots,20$ and the benchmark values. The shape of the predicted initial margin is closer and closer to the benchmark value as $degree$ increases. However, the shape of the predicted initial margin becomes increasingly inconsistent with the benchmark when $degree \geq 9$.

Furthermore, table 2.2 demonstrates the similar result. Table 2.2 shows the mean squared error between the benchmark values and the predicted initial margin from the polynomial regression methods with $degree = 1,2,\ldots,20$. It is clear that initially the mean squared error decreases with increasing degrees until $degree = 8$, and then the mean squared error increases with decreasing degrees.

2.2.3 Comparison

From the above figures and tables, we conclude that it is meaningless to set the highest degree of the basis function to a number greater than 9 in the polynomial regression method for both parametric and non-parametric approaches. So, we compare the polynomial regression method with $degree = 1,2,\ldots,10$ by setting $Y_t = (V(\omega,t+\delta)-V(\omega,t))^2$ and $Y_t = IM(\omega,t)$.

Figure 2.4 shows the predicted initial margin at time t from the polynomial regression method with $degree = 1,2,\ldots,10$ and the benchmark values. The squared circle represents $Y_t = (V(\omega,t+$

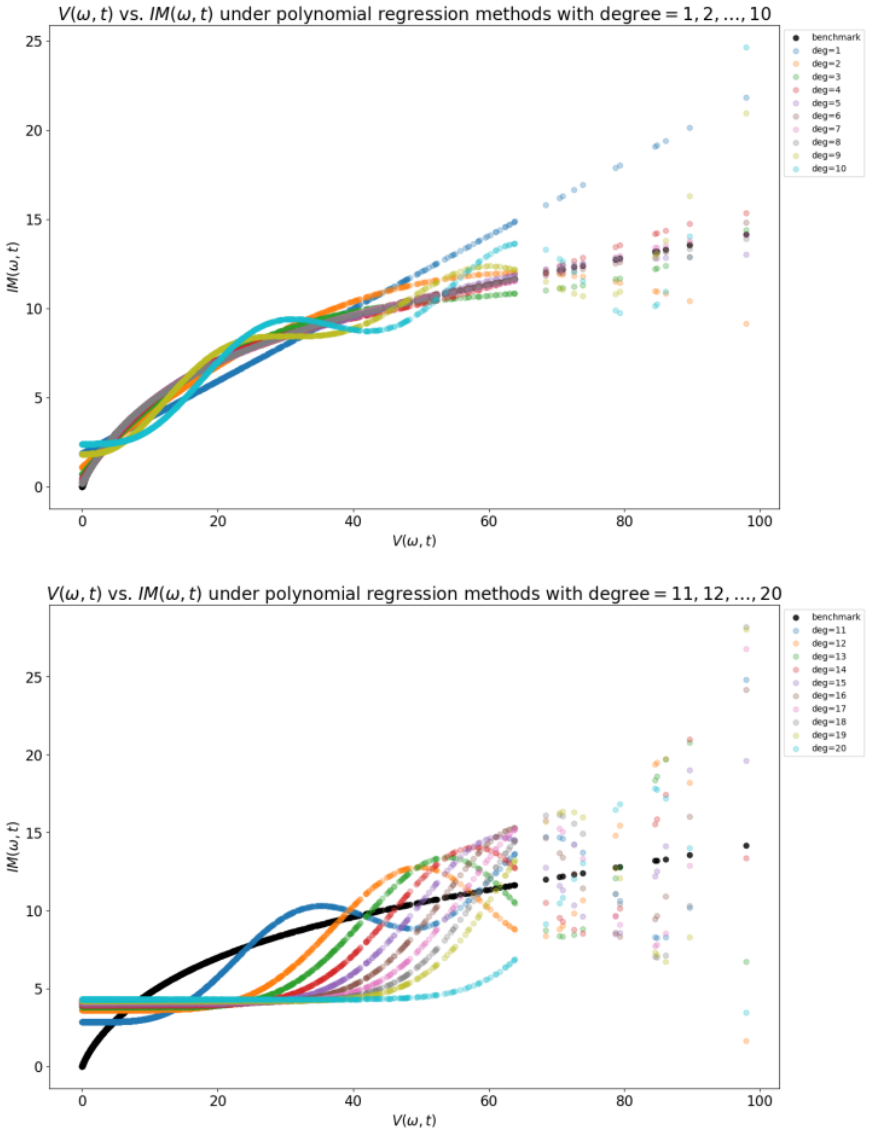


Figure 2.3: Polynomial regression method with $degree = 1, 2, \ldots, 20$: $X_t = V(\omega, t)$ and $Y_t = IM(\omega, t)$

Highest power of $\phi_i(x)$	Y_t	MSE	Run-time
$degree = 1$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	15.85158	0.31
	$Y_t = IM(\omega, t)$	1.04910	0.00
$degree = 2$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	15.88178	0.22
	$Y_t = IM(\omega, t)$	0.25512	0.00
$degree = 3$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	15.58592	0.27
	$Y_t = IM(\omega, t)$	0.07207	0.00
$degree = 4$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	15.58349	0.27
	$Y_t = IM(\omega, t)$	0.02499	0.00
$degree = 5$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	16.20285	0.26
	$Y_t = IM(\omega, t)$	0.00987	0.00
$degree = 6$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	0.17891	0.29
	$Y_t = IM(\omega, t)$	0.00434	0.01
$degree = 7$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	0.19323	0.34
	$Y_t = IM(\omega, t)$	0.00230	0.01
$degree = 8$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	16.68235	0.29
	$Y_t = IM(\omega, t)$	0.00129	0.01
$degree = 9$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	0.51906	0.33
	$Y_t = IM(\omega, t)$	0.61584	0.01
$degree = 10$	$Y_t = \big(V(t+\delta) - V(t)\big)^2$	1.14603	0.32
	$Y_t = IM(\omega, t)$	1.36411	0.01

Table 2.3: The comparison of the initial margins between $Y_t = \big(V(t+\delta) - V(t)\big)^2$ and $Y_t = IM(\omega, t)$ under the polynomial regression methods with degree$= 1, 2, \ldots, 10$

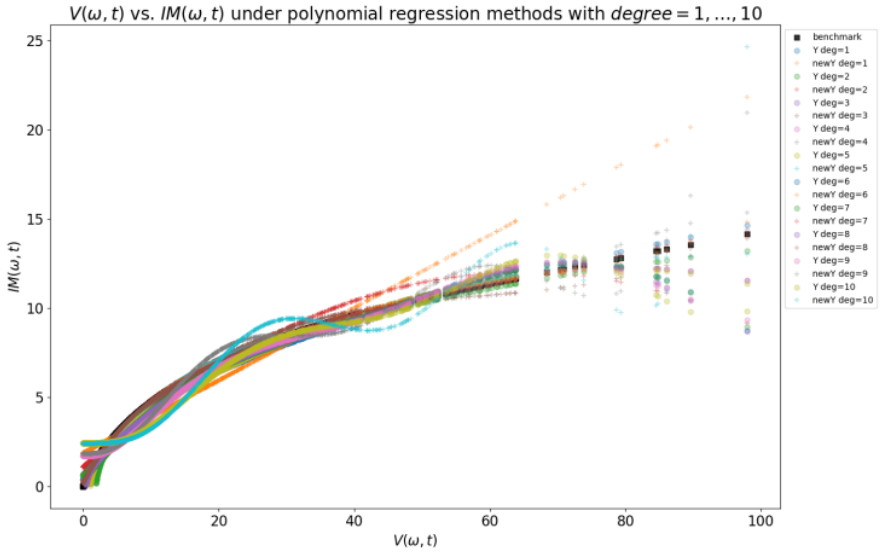


Figure 2.4: Polynomial regression method with $degree = 1, 2, \ldots, 20$: $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ and $Y_t = IM(\omega, t)$

$\delta) - V(\omega, t))^2$, and the plus sign represents $Y_t = IM(\omega, t)$. Two lines that clearly do not match the benchmark values are the polynomial regression method with $degree = 1$ and $Y_t = IM(\omega, t)$ and the polynomial regression method with $degree = 10$ and $Y_t = IM(\omega, t)$. The polynomial regression methods with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ look better overall. However, table 2.3 indicates different result. Table 2.3 shows the mean squared error between the polynomial regression method with $degree = 1, 2, \ldots, 10$ and the benchmark values. Generally, the polynomial regression methods with $Y_t = IM(\omega, t)$ has smaller mean squared error.

I think there are two reasons why we can obtain different conclusions from the table and the graph. The first reason is that non-parametric methods set the initial margin directly on the response variable instead of taking some additional steps after the prediction as parametric methods do. Thus, the mean squared error of the non-parametric method is smaller than that of the parametric method. In addition, the parametric approach needs to filter out the negative value for $(V(\omega, t+\delta) - V(\omega, t))^2$ because the formula of the initial margin (1.2.1) obtains $\sqrt{\mathbb{E}[(V(\omega, t+\delta) - V(\omega, t))^2]}$. This step takes out lots of values, so the shape of the predictions from the parametric approach looks better overall.

Chapter 3

Kernel Regression Method

Kernel regression is a non-parametric approach for estimating a random variable's conditional expectation. The purpose is to investigate a non-linear relationship between two random variables. We can use kernel regression to predict the future initial margin.

3.1 Methodology

Kernel regression estimates the conditional expectation of a random variable as a locally weighted average. There are different ways to estimate the weight, and we focus on the Nadaraya-Watson estimator and the local linear estimator in this section.

3.1.1 Kernel Regression Estimation

In a regression method, the form the function $f(\boldsymbol{X}, \boldsymbol{\beta})$ needs to be defined at the beginning based on the guessing and experience. It is a comparatively strong and case-dependent assumption. In other words, if the structure of the function is guessed correctly, the prediction will be very accurate; otherwise, a bad model is generated. Thus, this method is theoretical and hard to apply in some real-world scenarios. To avoid this specific model assumption, the non-parametric regression method, which has least assumptions, is more practical.

The aim of the method is to estimate the response variable Y with the unknown $X = x$. This problem can be seen as the estimation of the conditional expectation $m(x)$ with the known density function $f_{Y|X=x}(y)$. According to Portugués [3],

$$m(x) = \mathbb{E}[Y \mid X = x] = \int y \; f_{Y|X=x}(y) \; dy \tag{3.1.1}$$

where

$$f_{Y|X=x}(y) = \frac{f(x,y)}{f_X(x)}$$

The first step is to approximate $f_X(x)$, the density of X, by using the given sample data $X_1, \ldots, X_n$ that is normally distributed. Since the sample data is given, the simplest method to get the density of X is to use the naive estimator. The idea is, for a given bandwidth $h > 0$, to count the number of X's that fall into the interval $(x-h, x+h]$, and then use the relative frequency in $(x-h, x+h]$ to calculate the density at x. Thus, the naive estimator is a piece-wise constant, and the formula of the naive estimator can be presented as

$$\begin{aligned} f(x;h) = F'(x) &= \lim_{h \to 0^+} \frac{\mathbb{P}[x-h < X \leq x+h]}{2h} \\ \Rightarrow \hat{f}_N(x;h) &= \frac{1}{2nh} \sum_{i=1}^{n} \mathbb{1}_{\{x-h < X_i \leq x+h\}} \\ \Rightarrow \hat{f}_N(x;h) &= \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{2} \mathbb{1}_{\{-1 < \frac{x-X_i}{h} \leq 1\}} \end{aligned}$$

If define

$$K(z) \coloneqq \frac{1}{2}\ \mathbb{1}_{\{-1<z\leq 1\}}$$

and

$$K_h(z) \coloneqq \frac{1}{h}\ K\Big(\frac{z}{h}\Big)$$

, the density changes to

$$\hat{f}(x;h) = \frac{1}{nh}\ \sum_{i=1}^{n} K\Big(\frac{x-X_i}{h}\Big)$$
$$\Rightarrow \hat{f}(x;h) = \frac{1}{n}\ \sum_{i=1}^{n} K\big(x-X_i\big)$$

, which is called kernel density estimator (kde). From the above equations, Portugués [3] summarized that the kernel K is a standardized density that satisfies the followings:

- $K(-u) = K(u)$
- $\int_{-\infty}^{\infty} K(u)du = 1$
- positive semi-definiteness: $K(u) \geq 0\ \forall u$

Since the bandwidth h acts like a tuning parameter in the formula of kernel density estimator, it is necessary to consider how to choose the value of h. If the value of h is too large, the approximated density $\hat{f}(x;h)$ has a small variance but a large bias; otherwise, $\hat{f}(x;h)$ has a small bias but a large variance. It is important to determine the value of h to have a trade-off between variance and bias of the approximation.

3.1.2 Nadaraya–Watson Estimator

Based on (3.1.1), $m(x)$ can be re-written as

$$\begin{aligned} m(x) &= \mathbb{E}(Y \mid X = x) \\ &= \int\ y\ f_{Y|X=x}(y)dy \\ &= \frac{\int\ y\ f(x,y)dy}{f_X(x)} \end{aligned}$$

For estimating the non-parametric equation $m(x)$ with the given sample data $\{(X_i, Y_i)\}_{i=1}^{n}$, the solution is to use kernel density estimator to estimate the densities $f(x,y)$ and $f_X(x)$. For the joint probability density function $f(x,y)$, consider it as a two-dimensional kde with the bandwidth $\boldsymbol{h} = (h_x, h_y)'$, so

$$\hat{f}(x,y;\boldsymbol{h}) = \frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x-X_i)\ K_{h_y}(y-Y_i)$$

Similarly, estimating $f_X(x)$ with bandwidth h_x is

$$\hat{f}_X(x;\boldsymbol{h}) = \frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x-X_i)$$

Thus, the estimation of $m(x)$ is

$$\begin{aligned} m(x) &= \frac{\int\ y\ f(x,y)dy}{f_X(x)} \\ &= \frac{\int\ y\ \hat{f}(x,y)dy}{\hat{f}_X(x)} \\ &= \frac{\int\ y\ \left[\frac{1}{n}\sum_{i=1}^{n} K_x(x-X_i)\ K_y(y-Y_i)\right]\ dy}{\frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x-X_i)} \\ &= \frac{\frac{1}{n}\sum_{i=1}^{n} K_x(x-X_i)\ \left[\int yK_y(y-Y_i)dy\right]}{\frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x-X_i)} \\ &= \sum_{i=1}^{n} \frac{K_x(x-X_i)}{\sum_{i=1}^{n} K_x(x-X_i)}\ Y_i \end{aligned}$$

Therefore, based on the proposals by Nadaraya [4] and Watson [5], the Nadaraya–Watson Estimator $\hat{m}(x;0,h)$ is

$$\hat{m}(x;0,h) \coloneqq w_i^0(x)\ Y_i \tag{3.1.2}$$

where

$$w_i^0(x) = \sum_{i=1}^{n} \frac{K_x(x-X_i)}{\sum_{j=1}^{n} K_x(x-X_j)}$$

3.1.3 Local Linear Estimator

No matter which method is used to estimate $m(x)$, the goal is to minimize the sum of squared residuals (SSR), which is

$$SSR = \sum_{i=1}^{n} \Big(Y_i - \hat{m}(X_i)\Big)^2$$

. The difference between parametric and non-parametric regression methods is that whether assuming the structure of $m(x)$ or not. Usually, due to the structure of $m(x)$ is pre-defined in the parametric regression method, there exist some methods, such as Ordinary Least Squares (OLS), can be used to determine the parameters in $m(x)$. However, in the non-parametric regression method, the structure of $m(x)$ is unknown, another method is provided.

According to Portugués [3], for estimating $m(X_i)$, the alternative method is using the p^{th} Taylor polynomial at a point x, which is

$$m(X_i) \approx m(x) + m'(x)(X_i - x) + \frac{m''(x)}{2}(X_i - x)^2 + \cdots + \frac{m^{(p)}}{p!}(X_i - x)^p \tag{3.1.3}$$

$$= \sum_{j=0}^{p} \frac{m^{(j)}}{j!}(X_i - x)^j \tag{3.1.4}$$

Then, the SSR becomes

$$SSR = \sum_{i=1}^{n} \left[Y_i - \Big(\sum_{j=0}^{p} \frac{m^{(j)}}{j!}(X_i - x)^j\Big)\right]^2 \tag{3.1.5}$$

It is still impossible to minimize SSR because the equation (3.1.5) depends on the unknown function $m(x)$ and its derivatives. However, by setting $\beta_j \coloneqq \frac{m^{(j)}}{j!}$, the equation (3.1.5) changes to

$$SSR = \sum_{i=1}^{n} \left[Y_i - \Big(\sum_{j=0}^{p} \beta_j(X_i - x)^j\Big)\right]^2$$

, which is a linear regression problem for estimating $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$. Once determining the estimation of $\boldsymbol{\beta}$, the estimation of $m^j(x)$ is known as well.

Consider the different contribution of each point (X_i, Y_i) because of the distance between X_i and x, the weight can be determined through the kernel. Thus, the estimation of β_j is

$$\hat{\beta}_j \coloneqq \arg\min_{\beta \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left[Y_i - \Big(\sum_{j=0}^{p} \beta_j (X_i - x)^j \Big) \right]^2 K_h(x - X_i)$$

where $K_h(x - X_i)$ is a kernel, also a density.

Define

$$\boldsymbol{X} \coloneqq \begin{pmatrix} 1 & X_1 - x & (X_1 - x)^2 & \dots & (X_1 - x)^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_n - x & (X_n - x)^2 & \dots & (X_n - x)^p \end{pmatrix} \boldsymbol{Y} \coloneqq \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

$$\boldsymbol{w} \coloneqq diag\Big(K_h(X_1 - x), \dots, K_h(X_n - x) \Big)$$

$$\hat{\boldsymbol{\beta}} \coloneqq \Big(m(x), m'(x), \frac{m''(x)}{2} \dots, \frac{m^{(p)}}{p!} \Big)'$$

, then

$$\hat{m}(x; p, h) \coloneqq \hat{\beta_{h,0}} = \sum_{i=1}^{n} w_i^p(x) Y_i$$

where $w_i^p(x) \coloneqq e_1'(\boldsymbol{X'WX}^{-1}\boldsymbol{X'W}e_i)$

Therefore,

- When taking $p = 0$, $\hat{m}(x; 0, h)$ is a local constant estimator, also known as Nadaraya–Watson Estimator.
- When taking $p = 1$, $\hat{m}(x; 1, h)$ is a local linear estimator, and the equation is

$$\hat{m}(x; 1, h) \coloneqq \hat{\beta}_{h,0} = \sum_{i=1}^{n} w_i^1(x) Y_i$$

where

$$w_i^1(x) = \frac{1}{n} \frac{\hat{s}_2(x; h) - \hat{s}_1(x; h)(X_i - x)}{\hat{s}_2(x; h)\hat{s}_0(x; h) - \hat{s}_1(x; h)^2} K_h(x - X_i)$$

and

$$\hat{s}_r(x; h) \coloneqq \frac{1}{n} \sum_{i=1}^{n} (X_i - x)^r K_h(x - X_i)$$

3.2 Implementation

We can use both the Nadaraya-Watson estimator and the local linear estimator to calculate the future initial margin in parametric and non-parametric approaches.

3.2.1 Nadaraya-Watson Kernel Regression

As mentioned in the last section, we use the Nadaraya-Watson estimator equation to estimate the conditional expectation, where the formula is

$$\hat{m}(x; 0, h) \coloneqq \sum_{i=1}^{n} \frac{K_h(x - X_i)}{\sum_{j=1}^{n} K_h(x - X_i)} Y_i$$

where $K_h(x) = \frac{1}{h} K\Big(\frac{x}{h}\Big)$, $K(\cdot)$ is a Gaussian kernel, and h is the bandwidth. The bandwidth can be determined via the Least-Square Cross-Validation, the Scott's Rule of Thumb and the Silverman's Rule of Thumb methods.

Figure 3.1 show the predicted initial margin at time t against the portfolio prices under the Nadaraya-Watson kernel regression method with three different bandwidths. The top figure shows the predicted $IM(\omega, t)$ under the Nadaraya-Watson kernel regression method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$, and the bottom figure shows the values under the Nadaraya-Watson kernel regression method with $Y_t = IM(\omega, t)$. Generally speaking, the regression method with $Y_t = IM(\omega, t)$ generates better predictions of the initial margin at time t on scenario ω because the benchmark value is almost covered by the other three lines in the figure 3.1(b). However, the predictions via the regression method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ are very close to the benchmark values in the beginning, but they become worse as the portfolio price at time t increases. Thus, we can see from the figure 3.1(a) that the predicted values are disordered and chaotic in the tails.

3.2.2 Local Linear Kernel Regression

The local linear estimator is

$$\hat{m}(x; 1, h) \coloneqq \sum_{i=1}^{n} \frac{1}{n} \frac{\hat{s}_2(x; h) - \hat{s}_1(x; h)(X_i - x)}{\hat{s}_2(x; h)\hat{s}_0(x; h) - \hat{s}_1(x; h)^2} Y_i$$

where $\hat{s}_r(x; h) \coloneqq \frac{1}{n} \sum_{i=1}^{n} (X_i - x)^r \; K_h(x - X_i)$, $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$, $K(\cdot)$ is a Gaussian kernel, and h is the bandwidth. The bandwidth can be determined via the Least-Square Cross-Validation, the Scott's Rule of Thumb and the Silverman's Rule of Thumb methods.

Figure 3.2 shows the predicted initial margin via the kernel regression method with the local linea estimator and different bandwidths for both parametric and non-parametric approaches. For the kernel regression method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$, the predictions almost cover the benchmark values at the beginning, but the tail does not perform as well. Compared to the method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$, the method with $Y_t = IM(\omega, t)$ predits good estimations overall.

3.2.3 Comparison

Figure 3.3 shows a comparison between the parametric and non-parametric approaches under the kernel regression method with the Nadaraya-Watson estimator and the local linear estimator. The kernel regression with $Y_t = IM(\omega, t)$ makes a better prediction of the initial margin at time t due to the predicted values are at the top of the benchmark values in both the Nadaraya-Watson estimator and the local linear estimator. For the kernel regression with $Y_t = IM(\omega, t)$, its predicted value is increasingly different from the benchmark value.

Table 3.1 shows the same conclusion. The table shows the mean squared error between the benchmark and the predicted values from the kernel regression methods. Generally, the mean squared errors of the kernel regression methods with $Y_t = IM(\omega, t)$ is smaller than other methods.

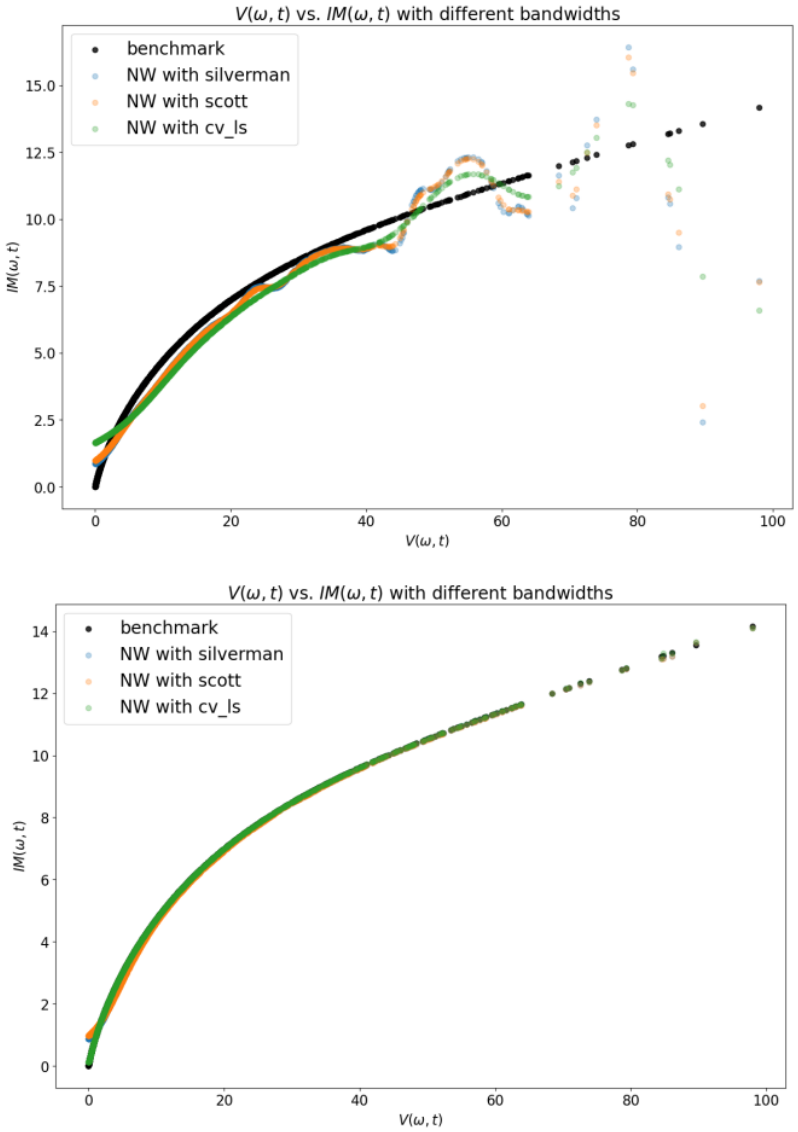


Figure 3.1: Kernel regression method with Nadaraya-Watson estimator: $Y_t = (V(\omega, t+\delta) - V(\omega,t))^2$ (top) and $Y_t = IM(\omega,t)$ (bottom)

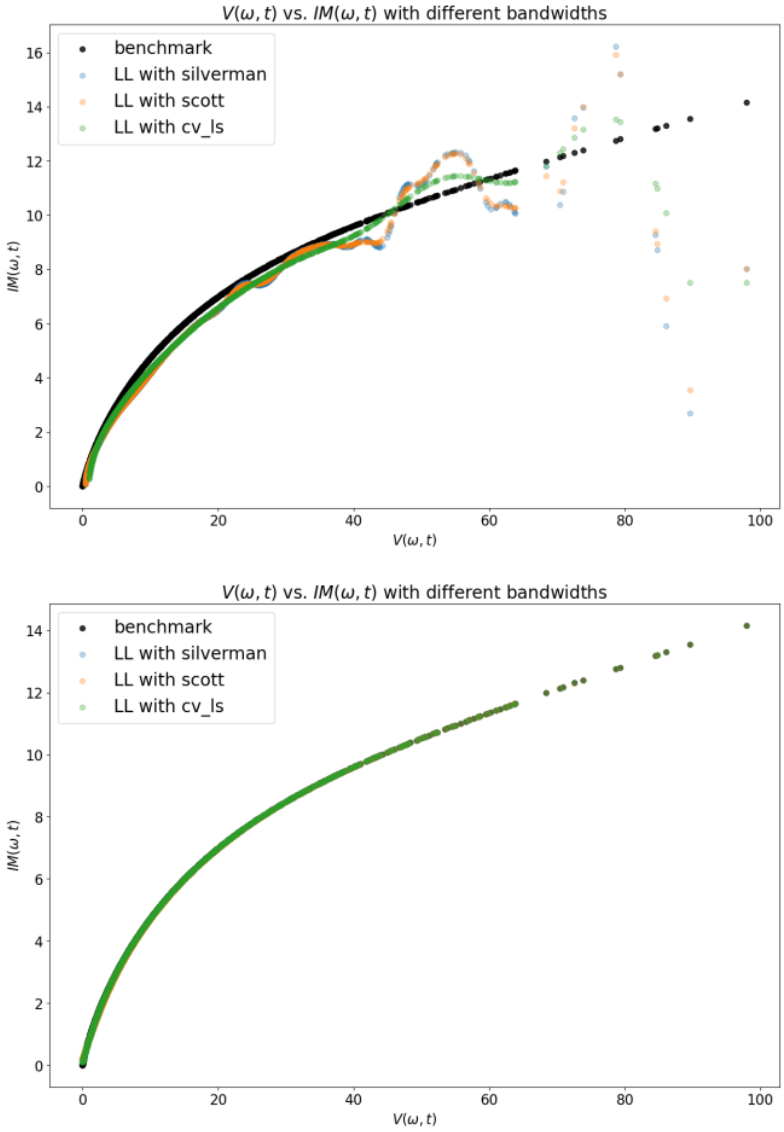


Figure 3.2: Kernel regression method with local linear estimator: $Y_t = (V(\omega, t+\delta) - V(\omega,t))^2$ (top) and $Y_t = IM(\omega,t)$ (bottom)

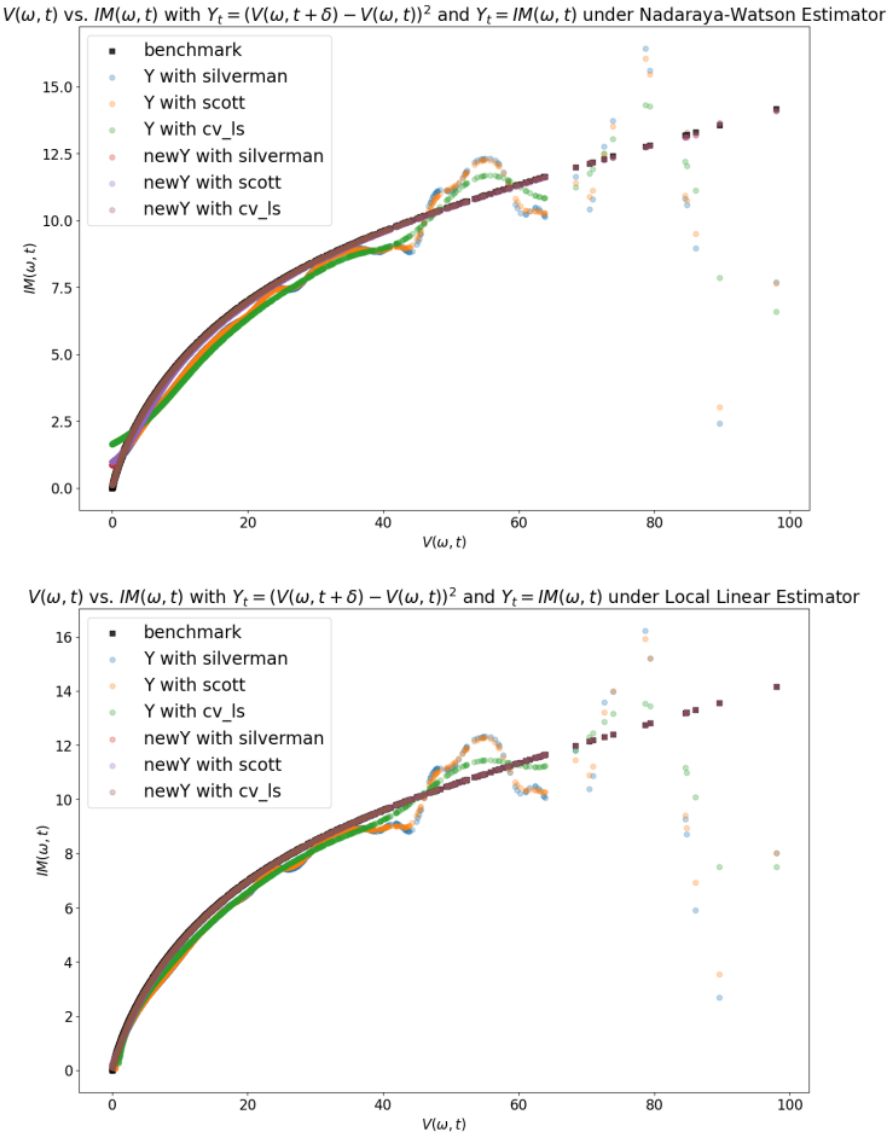


Figure 3.3: Kernel regression method with $Y_t = (V(\omega, t+\delta) - V(\omega,t))^2$ and $Y_t = IM(\omega,t)$:Nadaraya-Watson estimator (top) and local linear estimator (bottom)

Method	Y_t	MSE	Run-time
NW with silverman	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	0.36260	1.04
	$Y_t = IM(\omega, t)$	0.05199	0.70
NW with scott	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	0.36992	1.19
	$Y_t = IM(\omega, t)$	0.07401	0.70
NW with cv_ls	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	0.58075	129.54
	$Y_t = IM(\omega, t)$	0.00011	200.48
LL with silverman	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	16.31334	0.87
	$Y_t = IM(\omega, t)$	0.00254	0.76
LL with scott	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	16.24863	0.82
	$Y_t = IM(\omega, t)$	0.00380	0.74
LL with cv_ls	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	16.14926	144.15
	$Y_t = IM(\omega, t)$	0.00047	172.91

Table 3.1: The comparison of the initial margins between $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ and $Y_t = IM(\omega, t)$ under the Nadaraya-Watson estimator and the local linear estimator

Chapter 4

Dynamically Controlled Kernel Estimation Method

Dynamically Control Kernel Estimation (DCKE) is a model-free data-driven method to estimate conditional expectation according to Kienitz, Nowaczyk and Geng [6]. The main goal of this method is to generate an accurate estimation with a high level of speed.

4.1 Methodology

For having a better understanding of DCKE, we introduce the Gaussian Process Regression and control variate first.

4.1.1 Gaussian Process Regression

Gaussian Process Regression, also known as Kriging, is a non-parametric Bayesian method to interpolate the unseen data based on the given/historical sample data $(\boldsymbol{X}, \boldsymbol{Y}) = \{(X_i, Y_i)\}_{i=1}^n$ and the Bayes' Rule. The formula of the Bayes' Rule is

$$posterior = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Specifically, for using the predictive posterior distribution to predict y with the corresponding new x, it is necessary to calculate the posterior based on the Bayes' Rule and a designated prior.

Usually, assume the prior, $f(x)$, follows a Gaussian Process, which means

$$f(x) \sim GP(m(x), k(x, x'))$$

, and the noise ϵ follows an independently, identically distributed (i.i.d) Gaussian distribution, which means $\epsilon \sim N(0, \sigma^2)$. The mean function $m(x)$ can be a constant (e.g. $m(x) = 0$) or the mean of the training data, and the covariance kernel function $k(x, x')$ needs to follow the properties of a kernel, which is semi-positive definite and symmetric. The common kernel functions are constant, linear, square exponential, and radial basis function (RBF) and Matern kernels. The covariance kernel functions sometimes contain the hyperparameters, which can be used to control the learning processes. For example, the radial basis function (RBF) kernel

$$k(x, x') = \sigma^2 \ \exp\Big(-\frac{1}{2l^2}||x - x'||^2\Big)$$

contains two hyperparameters: variance σ^2 and lengthscale l. After every variables are setted up, Crepey and Dixon [7] stated that the predictive posterior distribution for the test point x^* is

$$f^*(x^*) \mid \boldsymbol{X}, \boldsymbol{Y}, x^* \sim N\big(\mathbb{E}(f^*(x^*) \mid \boldsymbol{X}, \boldsymbol{Y}, x^*), var(f^*(x^*) \mid \boldsymbol{X}, \boldsymbol{Y}, x^*)\big)$$

where

$$\mathbb{E}(f^*(x^*) \mid \boldsymbol{X}, \boldsymbol{Y}, x^*) = m(x^*) + k(x^*, \boldsymbol{X})\big[k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I\big]^{-1}\boldsymbol{Y}$$
$$\mathbb{V}ar(f^*(x^*) \mid \boldsymbol{X}, \boldsymbol{Y}, x^*)\big) = k(x^*, x^*) - k(x^*, x)\big[k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I\big]^{-1}\boldsymbol{Y}$$

To conclude, the target of the Gaussian Process Regression is to determine the posterior distribution based on the given prior distribution. The posterior distribution is a Gaussian process, and the mean and the covariance of this distribution are calculated using the given samples and the prior distribution. Then, we can use this posterior distribution to estimate the value for the unseen data. This idea is used in DCKE.

4.1.2 Control Variate

The control variate method is a standard technique for reducing variance. In other words, the purpose of using this method is to improve the efficiency of the convergence.

Let Y be a random variable, and assume that Z is also a random variable with $\mu_Z = \mathbb{E}[Z]$ and $\beta \in \mathbb{R}$. Then, define a new random variable Y^* as

$$Y^* \coloneqq Y + \beta(Z - \mu_z)$$

The mean of Y^* is

$$\begin{aligned}
\mathbb{E}\left[Y^*\right] &= \mathbb{E}\big[Y + \beta(Z - \mu_z)\big] \\
&= \mathbb{E}\big[Y\big] + \mathbb{E}\big[\beta(Z - \mu_z)\big] \\
&= \mathbb{E}\big[Y\big] + \beta\big[\mathbb{E}\big[Z\big] - \mu_z\big] \\
&= \mathbb{E}\big[Y\big]
\end{aligned}$$

The variance of Y^* is

$$\begin{aligned}
\mathbb{V}ar\left[Y^*\right] &= \mathbb{V}ar\big[Y + \beta(Z - \mu_z)\big] \\
&= \mathbb{V}ar\big[Y + \beta Z - \beta \cdot \mu_z)\big] \\
&= \mathbb{V}ar\big[Y\big] + \beta^2\ \mathbb{V}ar\big[Z\big] + 2\ \beta\ Cov\big[Y, Z\big]
\end{aligned}$$

For minimizing the variance of Y^*, we want to determine a optimal coefficient $\hat{\beta}$ to minimize $\beta^2\ \mathbb{V}ar\big[Z\big] + 2\ \beta\ Cov\big[Y, Z\big]$. Ideally,

$$\beta^2\ \mathbb{V}ar\big[Z\big] + 2\ \beta\ Cov\big[Y, Z\big] = 0 \qquad \Rightarrow \qquad \hat{\beta} = -\frac{Cov\big[Y, Z\big]}{\mathbb{V}ar\big[Z\big]}$$

Thus,

$$\begin{aligned}
\mathbb{V}ar\left[Y^*\right] &= \mathbb{V}ar\big[Y\big] + \hat{\beta}^2\ \mathbb{V}ar\big[Z\big] + 2\ \hat{\beta}\ Cov\big[Y, Z\big] \\
&= \mathbb{V}ar\big[Y\big] + \left(-\frac{Cov\big[Y, Z\big]}{\mathbb{V}ar\big[Z\big]}\right)^2 \mathbb{V}ar\big[Z\big] + 2\left(-\frac{Cov\big[Y, Z\big]}{\mathbb{V}ar\big[Z\big]}\right) Cov\big[Y, Z\big] \\
&= \mathbb{V}ar\big[Y\big] + \frac{\Big(Cov\big[Y, Z\big]\Big)^2}{\mathbb{V}ar\big[Z\big]} - 2\ \frac{\Big(Cov\big[Y, Z\big]\Big)^2}{\mathbb{V}ar\big[Z\big]} \\
&= \mathbb{V}ar\big[Y\big] - \frac{\Big(Cov\big[Y, Z\big]\Big)^2}{\mathbb{V}ar\big[Z\big]}
\end{aligned}$$

Based on the formula of correlation coefficient of Y and Z

$$\rho_{Y,Z} = \frac{Cov\big[Y, Z\big]}{\sqrt{\mathbb{V}ar\big[Y\big]}\ \sqrt{\mathbb{V}ar\big[Z\big]}} \in [-1, 1]$$

we have

$$\frac{\Big(Cov\big[Y, Z\big]\Big)^2}{\mathbb{V}ar\big[Z\big]} = \rho_{Y,Z}^2\ \mathbb{V}ar\left[Y\right]$$

Then,

$$\mathbb{V}ar\left[Y^*\right] = \mathbb{V}ar\big[Y\big] - \frac{\Big(Cov\big[Y, Z\big]\Big)^2}{\mathbb{V}ar\big[Z\big]} = \Big(1 - \rho_{Y,Z}^2\Big)\mathbb{V}ar\left[Y\right] < \mathbb{V}ar\left[Y\right]$$

Since the mean does not change and the variance decrease, we can replace Y with Y^* to obtain a faster convergence. This method is used in DCKE to improve the speed of the estimation.

4.1.3 DCKE

The basic idea of the DCKE method is to use Gaussian Process Regression to predict the conditional expectation for the given unsampled data. Since the given data usually has a large size, the speed of the Gaussian Process Regression is relatively slow. For improving the speed of the method and maintaining accuracy, we choose to decrease the size of the training data and add a control variate. Rather than running the Gaussian Process Regression with the whole sample data, we select the training data from the given samples to do the modelling. In other words, we create a mesh grid for the given samples. However, because the points on the mesh grid may not appear in the given samples, we prefer to use the local regression to estimate the conditional expectation for these points. The next step is to use the selected points and their corresponding conditional expectation that are estimated via the local regression to do a Gaussian Process Regression. To ensure the conditional expectation converges to the same value faster, we add a control variate to the local regression. In addition, this control variate should also be a conditional expectation, which has the same condition as the target conditional expectation. Under this requirement, we can apply local regression again to estimate the control variate. Meanwhile, we need to determine the optimal coefficient for reducing the variance.

To summarize, the whole process contains five steps:

1. Creating a mesh grid $\{x_{i_1}, \ldots, x_{i_k}\}$ for the given samples.
2. Estimating $\tilde{y}_{i_\nu} \approx \mathbb{E}[Y \mid X = x_{i_\nu}]$ for every x_{i_ν} in the mesh grid via the local regression.
3. Estimating $\tilde{z}_{i_\nu} \approx \mathbb{E}[Z \mid X = x_{i_\nu}]$ for every x_{i_ν} in the mesh grid via the local regression, and calculating the optimal coefficient $\hat{\beta}^*_\nu$.
4. Define $\bar{y}_{i_\nu} := \tilde{y}_{i_\nu} + \beta^*_\nu(\tilde{z}_{i_\nu} - \mu_Z(x_{i_\nu}))$. Using the results form the previous steps to calculate $\bar{y}_{i_\nu}$ for every x_{i_ν} in the mesh grid.
5. Training the Gaussian Process Regression based on the mesh grid $\{x_{i_1}, \ldots, x_{i_k}\}$ and the corresponding $\{\bar{y}_{i_1}, \ldots, \bar{y}_{i_k}\}$.

Next, we will explain how to use this method to estimate the initial margin at time t.

4.2 Implementation

For using the Dynamically Controlled Kernel Estimation to calculate the initial margin at time t, few parameters need to be determined:

- The response variable Y_t and the independent variable X_t:
  - For parametric approach, define

$$X_t = \Big[V(\omega_1, t), V(\omega_2, t), \ldots, V(\omega_N, t)\Big]^T$$
$$Y_t = \Big[(V(\omega_1, t+\delta) - V(\omega_1, t))^2, (V(\omega_2, t+\delta) - V(\omega_2, t))^2, \ldots, (V(\omega_N, t+\delta) - V(\omega_N, t))^2\Big]^T$$

  - For non-parametric approach, define

$$X_t = \Big[V(\omega_1, t), V(\omega_2, t), \ldots, V(\omega_N, t)\Big]^T$$
$$Y_t = \Big[IM(\omega_1, t), IM(\omega_2, t), \ldots, IM(\omega_N, t)\Big]^T$$

- Training data $(X, Y) = \Big\{(X(\omega_i), Y(\omega_i))\Big\}_{i=1}^N$ and test data X^*:
  For the training data, using the solution of the geometric Brownian motion and the Black-Scholes pricing formula to generate the stock prices at times t and $t+\delta$ and the corresponding portfolio prices. On each path ω, the portfolio price at time t and the change on the portfolio price at times t and $t+\delta$ are determined. These are the training data. The test data only contains the portfolio price at time t.

- A mesh grid $\{x_{i_1}, \ldots, x_{i_k}\}$:
  We select several points between 1% and 99% quantile of the independent variable X_t with equal distance.

- Local regression:
  Based on the discussion in chapter 5, we can use the Nadaraya-Watson estimator and local linear estimator to interpolate the conditional expectation.

- Kernel function:
  The kernel function needs to be specified for the local regression and the Gaussian Process regression. In this chapter, we choose the radial basis function (RBF) kernel.

- control variate Z:
  We choose the portfolio price at time t as the control variate Z.

4.2.1 Parametric Approach

When define

$$X_t = \Big[V(\omega_1, t), V(\omega_2, t), \ldots, V(\omega_N, t)\Big]^T$$
$$Y_t = \Big[(V(\omega_1, t+\delta) - V(\omega_1, t))^2, (V(\omega_2, t+\delta) - V(\omega_2, t))^2, \ldots, (V(\omega_N, t+\delta) - V(\omega_N, t))^2\Big]^T$$

, the initial margin at time t on the scenario ω can be calculated in two steps. The first step is to use DCKE to estimate $\Big(V(\omega_i, t+\delta) - V(\omega_i, t)\Big)^2$, and then use the formula (1.2.1) to calculate the initial margin.

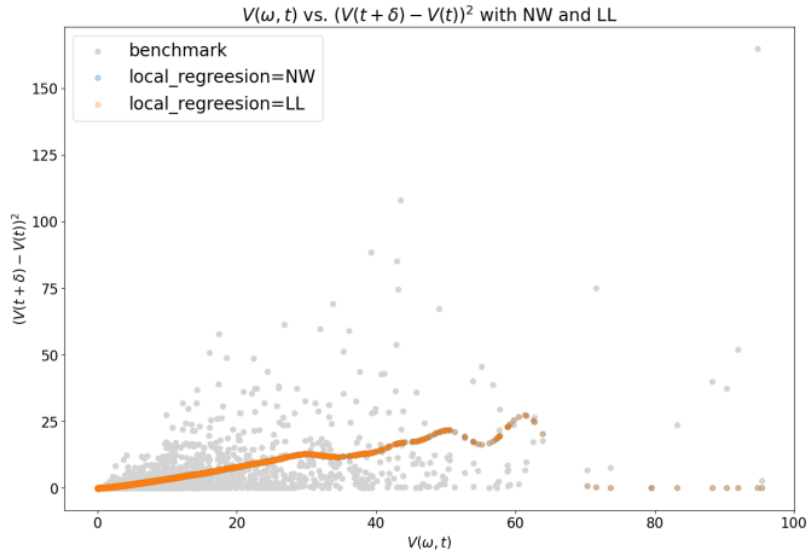


Figure 4.1: DCKE with Nadaraya-Watson and local linear estimator: $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$

Figure 4.1 shows a comparison of $(V(\omega, t+\delta) - V(\omega, t))^2$ that are generated from the Black-Scholes pricing formula and DCKE with Nadaraya-Watson and local linear estimators. The grey dots, which is the value from the Black-Sholes pricing formula, are scattered in the bottom left of the graph. The estimated values, which are the blue and orange dots, are shown as a line in the graph through the grey dots. Thus, the mean squared error of $\Big(V(\omega_i, t+\delta) - V(\omega_i, t)\Big)^2$ between the Black-Scholes pricing formula and DCKE is 78.60, which is a comparatively large value.

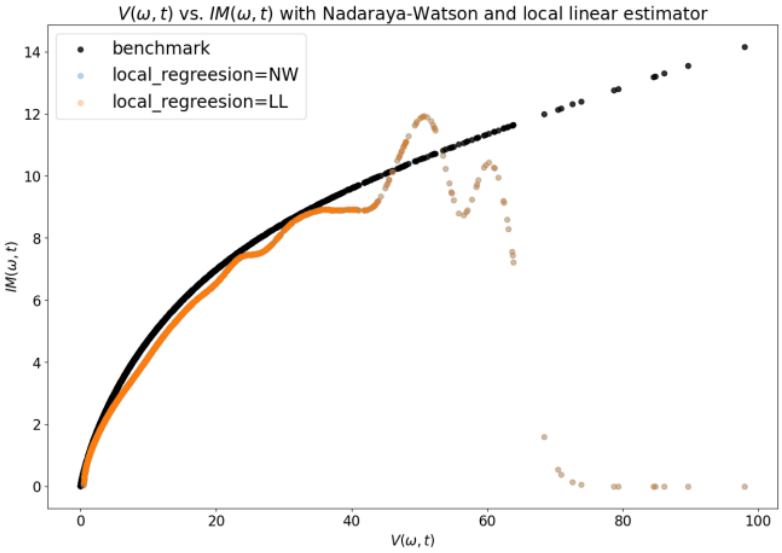


Figure 4.2: DCKE with Nadaraya-Watson and local linear estimator: $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$

Figure 4.2 shows a comparison of $IM(\omega, t)$ between the benchmark and DCKE methods. When the portfolio value at time t is less than 35, the estimated $IM(\omega, t)$ is very close to the benchmark value. However, when $V(\omega, t) > 35$, the predicted values float around the benchmark values. In the tail of the grey line, the predicted values have a large difference from the benchmark values.

4.2.2 Non-parametric Approach

When define

$$X_t = \Big[V(\omega_1, t), V(\omega_2, t), \ldots, V(\omega_N, t)\Big]^T$$
$$Y_t = \Big[IM(\omega_1, t), IM(\omega_2, t), \ldots, IM(\omega_N, t)\Big]^T$$

, the initial margin at the time t on scenario ω is directly estimated from DCKE.

Figure 4.3 shows a comparison of $IM(\omega, t)$ between the benchmark and DCKE methods. From figure 4.3, we can find out that the estimated $IM(\omega, t)$ from DCKE is almost on the top of the benchmark values, except the tail of the line. When $V(\omega, t) > 70$, the differences between the benchmarks and the predictions are large.

4.2.3 Comparison

Figure 4.4 shows the predicted initial margin from the Dynamically Controlled Kernel Estimation Method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ and $Y_t = IM(\omega, t)$ compared to the benchmark values. Generally, the Dynamically Controlled Kernel Estimation Method with $Y_t = IM(\omega, t)$ performs better. It is clear that the predictions form the method with $Y_t = IM(\omega, t)$ almost cover the benchmark values. However, the difference between the predictions of the method with $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ and the benchmark values becomes larger and larger as $V(\omega, t)$ increases. We can get the same conclusion form table 4.1 as well. Table 4.1 shows the mean squared error between the predicted initial margins from the Dynamically Controlled Kernel Estimation Method with different Y_t and the benchmark values. There is a enormous difference between the mean squared error from the parametric and non-parametric approaches. The mean squared error for the parametric approach is greater than 16, but for the non-parametric approach is around 1.

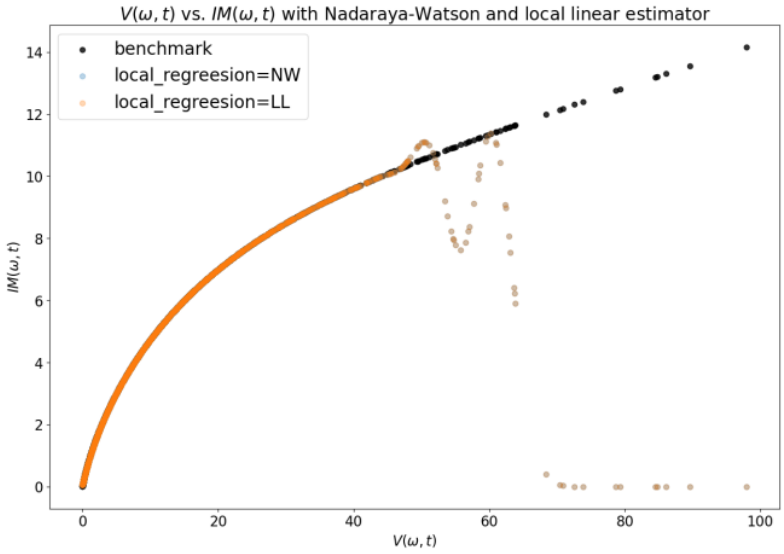


Figure 4.3: DCKE with Nadaraya-Watson and local linear estimator: $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$

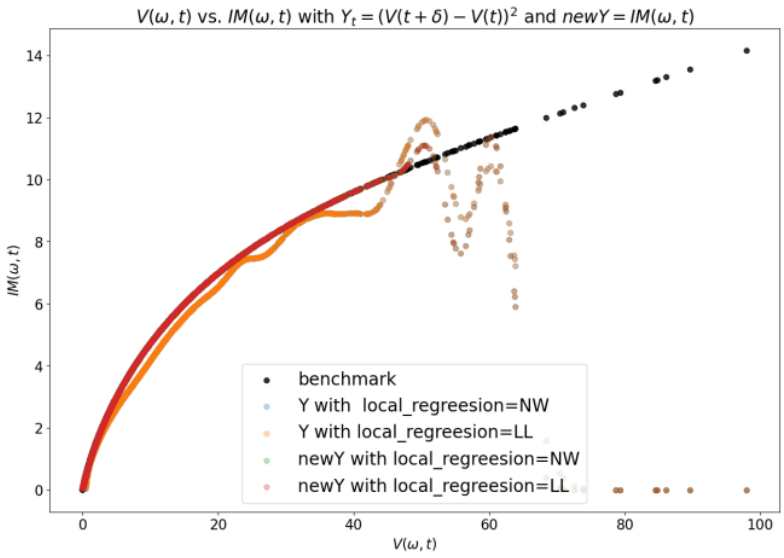


Figure 4.4: DCKE with Nadaraya-Watson and local linear estimator for different Y_t

Method	Y_t	MSE	Run-time
Nadaraya-Watson Estimator	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	16.15105	0.48
	$Y_t = IM(\omega, t)$	1.09570	0.18
Local Linear Estimator	$Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$	16.15105	0.48
	$Y_t = IM(\omega, t)$	1.09570	0.22

Table 4.1: The comparison of the initial margins between $Y_t = (V(\omega, t+\delta) - V(\omega, t))^2$ and $Y_t = IM(\omega, t)$ under DCKE with Nadaraya-Watson and local linear estimator

Chapter 5

Comparison

In this chapter, we provide a comprehensive comparison of the benchmark, the brute force, the polynomial regression, the kernel regression, and the Dynamic Control Kernel Estimation methods for the European Call option.

5.1 Preparation

Since the main purpose is to compare different methods to see which method performs more accurate and effective results, there is a need to ensure a fair environment.

The most important preparation is to generate train and test data. Whatever the method we use, the stock prices and the option prices at time t and $t+\delta$ are always needed. Therefore, at time points t and $t=\delta$, the analytical solution of the geometric Brownian motion (1.1.1) is used to generate several different stock prices and then the Black-Scholes pricing formula (1.1.2) is used to determine the corresponding portfolio prices. This data is split into training data and test data. When the methods use the same data as the training data, this setup avoids the situation where a method outputs a better model because the input data is different. After training the model, the next step is to use the sample-based model to predict the values of the same test data. The comparison results in this setup are reliable.

In addition, it is necessary to set the time t, which is the point in time t and the time period δ for calculating the initial margin. Note that the initial margin is calculated based on a small period, so the value of δ should be relatively small. Thus, there are four important time point need to be determined before the program: the start time ($t_0=0$), the time for the initial margin t, the time at the end of the period $t+\delta$, and the time of maturity T.

Now, consider a European Call option with the parameter settings in Table 5.1. The initial stock price takes from 50 to 150, and the time point t takes four different values, which are 0.2,0.5,0.8 and 0.9. For close-out period δ, we set it to 0.01, 0.02 and $\frac{10}{365}\approx 0.027$. Assume there are 10,000 asset price paths from time $t_0=0$ to time t. We split the entire data into two subsets, i.e., training data and test data. The training data contains 8,000 paths and the test data contains 2,000 paths.

The comparison is between the following methods:

- The benchmark method (abbreviation: benchmark):
  The initial margin is calculated based on the formula of the Black-Scholes model and the geometric Brownian motion. All comparisons are between the benchmark and other methods.
- The tested Monte Carlo method:
  The initial margin is predicted using the nested Monte Carlo method. Since the paths between two different time points can be placed, we set the number of paths to 100 and 1000 respectively. Thus,
  - the nested Monte Carlo method with the number of paths $M=100$ (abbreviation: bf100),
  - the nested Monte Carlo method with the number of paths $M=1000$ (abbreviation: bf1000).
- The polynomial regression method:
  In the polynomial regression, we consider the highest degree is $1,2,\ldots,10$. Thus,

Type	Parameter	number
Fixed	type of option	call
	the interest rate r	1%
	the volatility σ	3%
	the time of maturity T	1 year
	the number of paths at t_0	10,000
	the number of training data	8,000
	the number of test data	2,000
changeable	initial stock price S_0	50/80/100/120/150
	the strike price K	50/80/100/120/150
	the number of paths at t	$100/1,000$
	time t	0.2/0.5/0.8/0.9
	the close-out period δ	0.01/0.02/0.027
	the highest power of basis function deg	1/2/3/4/5/6/7/8/9/10
	the bandwidth methods	cv_ls/scott/silverman
	the local estimators	Nadaraya-Watson/Local linear

Table 5.1: Parameters

  - the polynomial regression method with the highest degree $= 1, 2, \ldots, 10$ (abbreviation: reg1, reg2, ... , reg10).

- The kernel regression method:
  The estimators of the kernel regression method are the Nadaraya-Watson estimator and the local linear estimator. Also, the bandwidths can be selected between the Silverman's Rule of Thumb, the Scott's Rule of Thumb and the Least-Square Cross-Validation. Thus, the kernel regression method contains six methods:

  - the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Silverman's Rule of Thumb (abbreviation: NW silverman),
  - the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Scott's Rule of Thumb (abbreviation: NW scott),
  - the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Least-Square Cross-Validation (abbreviation: NW cvls),
  - the kernel regression with the local linear estimator and bandwidth is the Silverman's Rule of Thumb (abbreviation: LL silverman),
  - the kernel regression with the local linear estimator and bandwidth is the Scott's Rule of Thumb (abbreviation: LL scott),
  - the kernel regression with the local linear estimator and bandwidth is the Least-Square Cross-Validation (abbreviation: LL cvls).

- The Dynamically Controlled Kernel Estimation method:
  In the Dynamically Controlled Kernel Estimation method, the mesh grid of X_t is 100. Similar to the kernel regression method, the estimators of the local regression are the Nadaraya-Watson estimator and the local linear estimator. Thus,

  - the Dynamically Controlled Kernel Estimation method with the Nadaraya-Watson estimator (abbreviation: DCKE NW),
  - the Dynamically Controlled Kernel Estimation method with the local linear estimator (abbreviation: DCKE LL).

5.2 Overview

For having a general view of the performance of all methods, we use different combinations of parameters to see the results. We decide that the initial stock price S_0, the strike price K, the time t and the position period are changed and all other parameters are fixed.

In this section, we show the performance of all methods when setting $K = 100$, $S_0 = 100$, $t = \{0.2, 0.5, 0.8, 0.9\}$ and $\delta = \{0.01, 0.02, 0.027\}$. The results for $K = 50, 80, 120, 150$ and $S_0 = 50, 80, 120, 150$ are shown in the appendix A.1.

The following figures show the predicted initial margin via all methods compared with the benchmark values when $S_0 = 100$ and $K = 100$. Figure 5.1, figure 5.2, figure 5.3 and figure 5.4 show the results when $t = 0.2$,$t = 0.5$,$t = 0.8$ and $t = 0.9$ separately. In each figure, the close-out period δ is set from top to bottom to 0.01, 0.02 and 0.027. Since the definition of Y_t is different for parametric and non-parametric approach, the left side of the figure sets $Y_t = (V(t + \delta) - V(t))^2$ and the right side of the figure sets $Y_t = IM(\omega, t)$.

In all scenarios, all methods converge universally to the benchmark, except the tail. More specifically, the predicted initial margin is far away from the benchmark value as the portfolio price $V(\omega, t)$ increases.

5.3 Discussion

Since the purpose of initial margin is to avoid the future close-out risk, we expect the true initial margin at time t to be less than the predicted initial margin in any case. In other words, the real worse case scenario is better than we expected and losses are under control.

The main purpose of estimating the initial margin at time t is to know how much money we will lose in the worst case scenario. Therefore, if the predicted initial margin is greater than the actual initial margin, this means that the forecast is valid. In addition, it is useful to know the percentage excess of incorrect forecasts. Tables 5.2, 5.3 and 5.4 shows

- $\widehat{IM}(\omega, t) < IM(\omega, t)$: Count how many valid forecasts from the method
- Invalid (%): Percentage of invalid predictions out of all predictions.
- Excess (%):

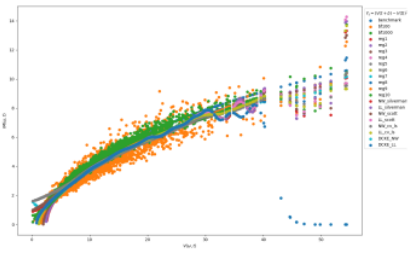
$$\text{excess} = \frac{IM(\omega, t) - \widehat{IM}(\omega, t)}{\widehat{IM}(\omega, t)}$$

  In the table, we only keep the maximum value of excess.

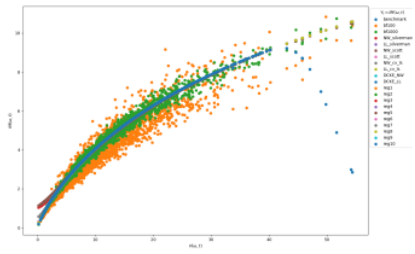
- Mean squared error (MSE): Calculate the mean of the squares of the error between every actual value and the predict value, and sum all the results.
- Run-time: The running time for the method

In general, the mean squared error should be small, which means that the model's prediction is a good estimate. In detail, the prediction should be valid, which means that the percentage of invalidity should be small. Even in the worse case where the prediction is invalid, we want the actual value to not exceed the predicted value by too much. In other words, the excess should be small.

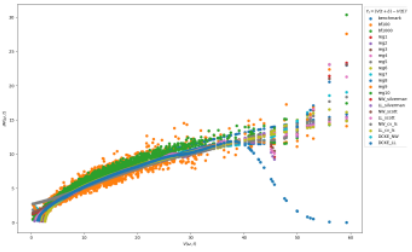
For $Y_t = (V(t+\delta)-V(t))^2$, table 5.2 shows that the nested Monte Carlo method with $M = 100$, the nested Monte Carlo method with $M = 1000$, the polynomial regression method with the highest degree = 3, the polynomial regression method with the highest degree = 7, the polynomial regression method with the highest degree = 9, the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Silverman's Rule of Thumb, the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Scott's Rule of Thumb, and the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Least-Square Cross-Validation all have $MSE < 1$, which is very small. Within these methods, the nested Monte Carlo method with $M = 1000$ and the polynomial regression method with the highest degree = 3 have the lowest excess percentage. However, since the run-time of the nested Monte Carlo method with $M = 1000$ is too large. Therefore, the polynomial regression method with the highest degree = 3 performs better. For $Y_t = IM(\omega, t)$, table 5.2 shows that the polynomial regression method with the highest degree = 8 because it has a comparatively smaller MSE, excess percentage and the run-time.



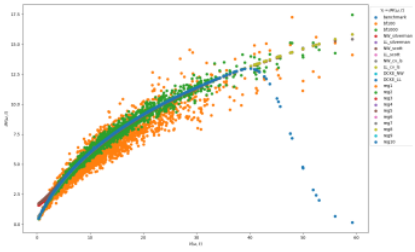
(a) $t = 0.2$ and $\delta = 0.01$: $Y_t = (V(t+\delta) - V(t))^2$



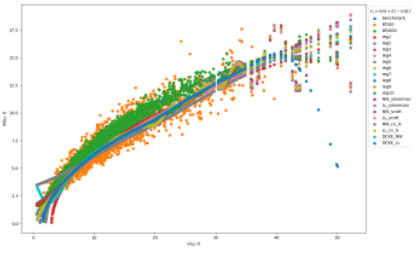
(b) $t = 0.2$ and $\delta = 0.01$: $Y_t = IM(\omega, t)$



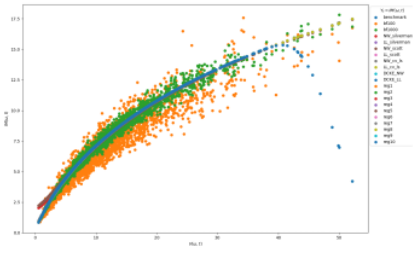
(c) $t = 0.2$ and $\delta = 0.02$: $Y_t = (V(t+\delta) - V(t))^2$



(d) $t = 0.2$ and $\delta = 0.02$: $Y_t = IM(\omega, t)$

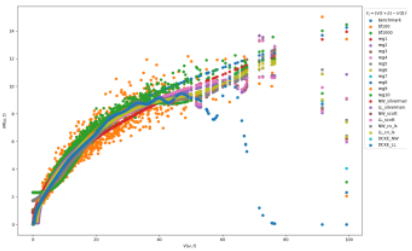


(e) $t = 0.2$ and $\delta = 0.027$: $Y_t = (V(t+\delta) - V(t))^2$

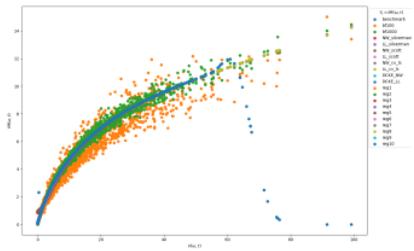


(f) $t = 0.2$ and $\delta = 0.027$: $Y_t = IM(\omega, t)$

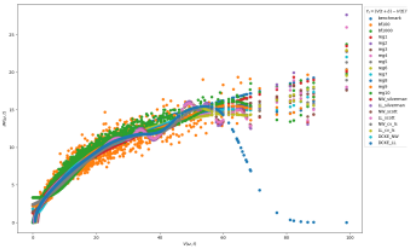
Figure 5.1: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ (left) and $Y_t = IM(\omega, t)$ (right) under different methods with $K = 100$, $S_0 = 100$, $t = 0.2$, and $\delta = \{0.01, 0.02, 0.027\}$



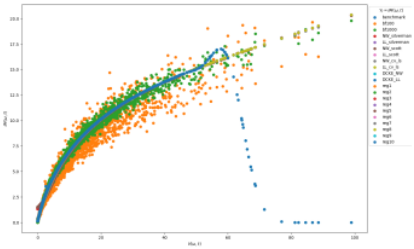
(a) $t = 0.5$ and $\delta = 0.01$: $Y_t = (V(t+\delta) - V(t))^2$



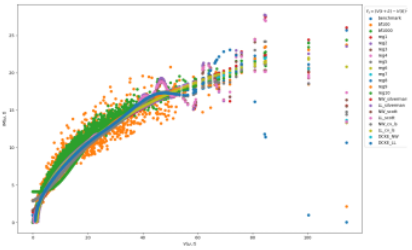
(b) $t = 0.5$ and $\delta = 0.01$: $Y_t = IM(\omega, t)$



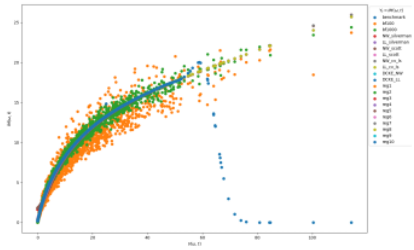
(c) $t = 0.5$ and $\delta = 0.02$: $Y_t = (V(t+\delta) - V(t))^2$



(d) $t = 0.5$ and $\delta = 0.02$: $Y_t = IM(\omega, t)$

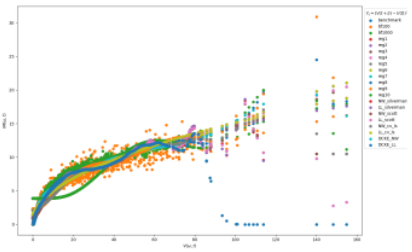


(e) $t = 0.5$ and $\delta = 0.027$: $Y_t = (V(t+\delta) - V(t))^2$

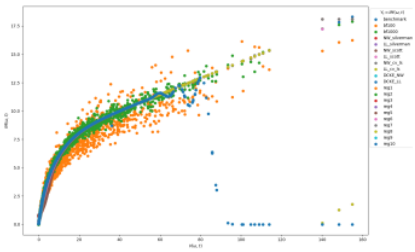


(f) $t = 0.5$ and $\delta = 0.027$: $Y_t = IM(\omega, t)$

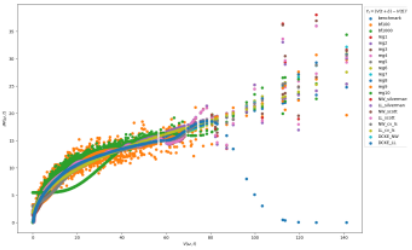
Figure 5.2: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ (left) and $Y_t = IM(\omega, t)$ (right) under different methods with $K = 100$, $S_0 = 100$, $t = 0.5$, and $\delta = \{0.01, 0.02, 0.027\}$



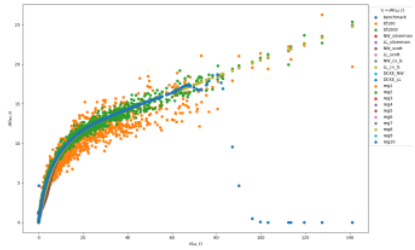
(a) $t = 0.8$ and $\delta = 0.01$: $Y_t = (V(t+\delta) - V(t))^2$



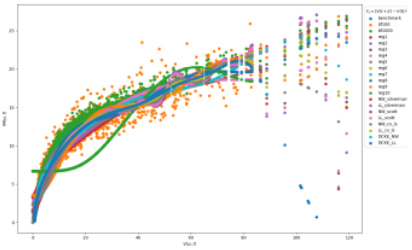
(b) $t = 0.8$ and $\delta = 0.01$: $Y_t = IM(\omega, t)$



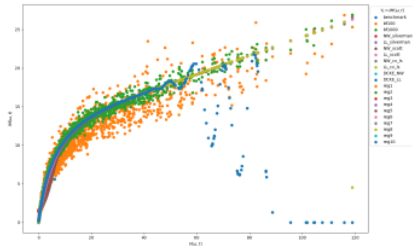
(c) $t = 0.8$ and $\delta = 0.02$: $Y_t = (V(t+\delta) - V(t))^2$



(d) $t = 0.8$ and $\delta = 0.02$: $Y_t = IM(\omega, t)$

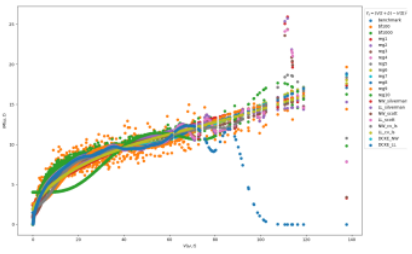


(e) $t = 0.8$ and $\delta = 0.027$: $Y_t = (V(t+\delta) - V(t))^2$

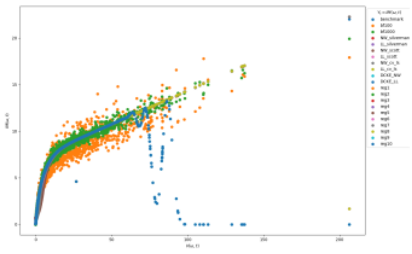


(f) $t = 0.8$ and $\delta = 0.027$: $Y_t = IM(\omega, t)$

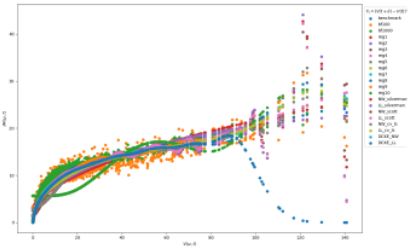
Figure 5.3: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ (left) and $Y_t = IM(\omega, t)$ (right) under different methods with $K = 100$, $S_0 = 100$, $t = 0.8$, and $\delta = \{0.01, 0.02, 0.027\}$



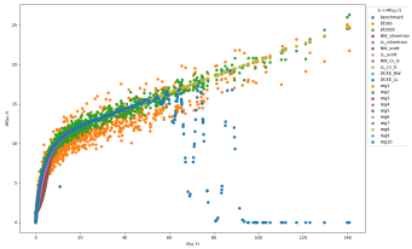
(a) $t = 0.9$ and $\delta = 0.01$: $Y_t = (V(t+\delta) - V(t))^2$



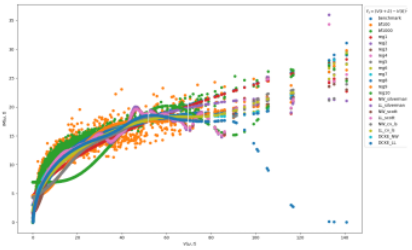
(b) $t = 0.9$ and $\delta = 0.01$: $Y_t = IM(\omega, t)$



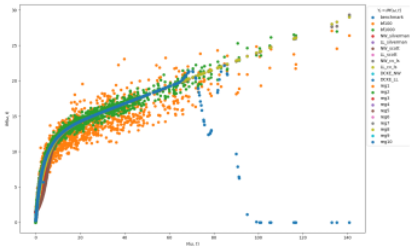
(c) $t = 0.9$ and $\delta = 0.02$: $Y_t = (V(t+\delta) - V(t))^2$



(d) $t = 0.9$ and $\delta = 0.02$: $Y_t = IM(\omega, t)$



(e) $t = 0.9$ and $\delta = 0.027$: $Y_t = (V(t+\delta) - V(t))^2$



(f) $t = 0.9$ and $\delta = 0.027$: $Y_t = IM(\omega, t)$

Figure 5.4: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ (left) and $Y_t = IM(\omega, t)$ (right) under different methods with $K = 100$, $S_0 = 100$, $t = 0.9$, and $\delta = \{0.01, 0.02, 0.027\}$

Y_t	Method	$\widehat{IM}(\omega,t) < IM(\omega,t)$	invalid(%)	excess(%)	MSE	Run-time
$Y_t = (V(t+\delta)-V(t))^2$	bf100	1565	0.78	0.60	0.40933	31.44
	bf1000	1212	0.61	0.16	0.04901	307.75
	reg1	721	0.44	36.77	14.73415	0.30
	reg2	769	0.46	27.74	15.01386	0.27
	reg3	1361	0.68	0.17	0.28119	0.28
	reg4	785	0.46	29.81	14.86567	0.21
	reg5	855	0.49	115.13	15.66387	0.23
	reg6	1222	0.61	191.42	12.80693	0.24
	reg7	1809	0.90	0.35	0.21047	0.26
	reg8	854	0.47	40.67	14.59653	0.28
	reg9	1458	0.73	0.32	0.62846	0.30
	reg10	1336	0.67	0.40	1.15621	0.27
	NW silverman	1627	0.81	0.39	0.34685	0.76
	LL silverman	989	0.52	153.46	15.41267	0.83
	NW scott	1618	0.81	0.32	0.35880	0.81
	LL scott	969	0.51	309.61	14.91912	0.86
	NW cvls	1341	0.67	0.23	0.75678	105.97
	LL cvls	800	0.47	62.82	14.52293	164.95
	DCKE NW	950	0.51	2.96×10^{10}	15.01247	0.59
	DCKE LL	950	0.51	2.96×10^{10}	15.01247	0.48
$Y_t = IM(\omega,t)$	bf100	1565	0.78	0.60	0.40933	31.44
	bf1000	1212	0.61	0.16	0.04901	307.75
	reg1	1185	0.59	0.23	0.97496	0.00
	reg2	1152	0.58	0.68	0.25501	0.00
	reg3	1123	0.56	0.09	0.06222	0.00
	reg4	1059	0.53	0.07	0.01915	0.01
	reg5	1039	0.52	0.06	0.00761	0.01
	reg6	1040	0.52	0.05	0.00362	0.00
	reg7	1026	0.51	0.05	0.00185	0.00
	reg8	1064	0.53	0.04	0.00103	0.00
	reg9	1087	0.54	0.43	0.53244	0.00
	reg10	1069	0.53	0.49	1.21797	0.00
	NW silverman	1665	0.83	0.12	0.04646	0.51
	LL silverman	1874	0.94	0.08	0.00239	0.66
	NW scott	1629	0.81	0.13	0.06726	0.50
	LL scott	1860	0.93	0.08	0.00357	0.56
	NW cvls	1100	0.55	0.06	0.00001	269.36
	LL cvls	1964	0.98	0.05	0.00003	264.83
	DCKE NW	1908	0.95	2.33×10^{22}	0.52365	0.16
	DCKE LL	1908	0.95	2.33×10^{22}	0.52365	0.25

Table 5.2: All methods with $S_0 = 100, t = 0.5, \delta = 0.01$

Y_t	Method	$\widehat{IM}(\omega,t) < IM(\omega,t)$	invalid(%)	excess(%)	MSE	Run-time
$Y_t = (V(t+\delta) - V(t))^2$	bf100	1587	0.79	0.63	1.01340	34.98
	bf1000	1176	0.59	0.19	0.11869	329.78
	reg1	796	0.47	37.09	34.36317	0.28
	reg2	944	0.51	40.34	33.46812	0.29
	reg3	719	0.45	125.71	31.91374	0.22
	reg4	954	0.52	177.72	34.99639	0.27
	reg5	880	0.49	305.20	33.57734	0.27
	reg6	1142	0.58	248.74	35.53057	0.30
	reg7	1854	0.93	0.28	0.62886	0.30
	reg8	869	0.48	142.04	33.34409	0.30
	reg9	1473	0.74	0.42	1.73343	0.34
	reg10	1361	0.68	0.48	2.74920	0.32
	NW silverman	1720	0.86	0.41	0.94525	0.86
	LL silverman	1002	0.53	104.79	32.45831	1.05
	NW scott	1670	0.83	0.34	0.96626	0.88
	LL scott	994	0.53	92.83	33.73500	1.00
	NW cvls	1413	0.71	0.30	1.74813	111.12
	LL cvls	934	0.51	174.65	33.10329	118.12
	DCKE NW	905	0.50	78.60	33.89703	0.39
	DCKE LL	905	0.50	78.60	33.89703	0.41
$Y_t = IM(\omega,t)$	bf100	1587	0.79	0.63	1.01340	34.98
	bf1000	1176	0.59	0.19	0.11869	329.78
	reg1	1121	0.56	0.23	2.26615	0.01
	reg2	1101	0.55	0.43	0.55436	0.00
	reg3	1097	0.55	0.10	0.14779	0.01
	reg4	1087	0.54	0.08	0.05327	0.00
	reg5	1071	0.54	0.06	0.02248	0.00
	reg6	1057	0.53	0.06	0.01031	0.02
	reg7	1048	0.52	0.05	0.00552	0.01
	reg8	1043	0.52	0.05	0.00299	0.00
	reg9	1057	0.53	0.36	1.18832	0.01
	reg10	1033	0.52	0.47	2.62298	0.00
	NW silverman	1658	0.83	0.13	0.11312	0.53
	LL silverman	1870	0.94	0.08	0.00632	0.70
	NW scott	1621	0.81	0.14	0.15948	0.53
	LL scott	1854	0.93	0.08	0.00940	0.64
	NW cvls	1492	0.75	0.05	0.00120	239.78
	LL cvls	1892	0.95	0.07	0.00392	209.10
	DCKE NW	1873	0.94	8.01×10^{10}	1.93878	0.34
	DCKE LL	1873	0.94	8.01×10^{10}	1.93878	0.28

Table 5.3: All methods with $S_0 = 100, t = 0.5, \delta = 0.02$

Y_t	Method	$\widehat{IM}(\omega,t) < IM(\omega,t)$	invalid(%)	excess(%)	MSE	Run-time
$Y_t = (V(t+\delta) - V(t))^2$	bf100	1588	0.79	0.78	1.41800	34.31
	bf1000	1212	0.61	0.21	0.16015	329.64
	reg1	811	0.48	58.40	41.85233	0.31
	reg2	825	0.48	37.29	42.86726	0.27
	reg3	903	0.51	42.36	43.81887	0.25
	reg4	822	0.48	65.18	43.86661	0.31
	reg5	876	0.50	41.01	42.40864	0.26
	reg6	1004	0.53	140.77	45.58682	0.33
	reg7	1768	0.88	0.47	1.30679	0.31
	reg8	1779	0.89	0.41	1.24283	0.31
	reg9	1619	0.81	2.43	1.95848	0.34
	reg10	1243	0.62	5.10	25.25425	0.33
	NW silverman	1718	0.86	1.09	1.82585	0.84
	LL silverman	1004	0.53	128.42	47.04700	0.83
	NW scott	1690	0.84	0.97	1.80085	0.80
	LL scott	990	0.53	54.13	44.83982	0.94
	NW cvls	1508	0.75	0.33	2.33198	107.05
	LL cvls	916	0.50	66.47	43.37862	133.47
	DCKE NW	909	0.51	490.61	45.36411	0.45
	DCKE LL	909	0.51	490.68	45.36409	0.45
$Y\_t = IM(\omega,t)$	bf100	1588	0.79	0.78	1.41800	34.31
	bf1000	1212	0.61	0.21	0.16015	329.64
	reg1	1141	0.57	0.23	3.22885	0.00
	reg2	1150	0.57	1.87	0.98724	0.00
	reg3	1131	0.57	0.10	0.29125	0.00
	reg4	1113	0.56	0.16	0.09605	0.00
	reg5	1099	0.55	0.07	0.04057	0.00
	reg6	1077	0.54	0.06	0.02074	0.00
	reg7	1067	0.53	0.09	0.01215	0.00
	reg8	1039	0.52	0.05	0.00815	0.01
	reg9	1099	0.55	0.36	2.86014	0.00
	reg10	1029	0.51	0.46	7.79051	0.00
	NWsilverman	1672	0.84	0.12	0.16601	0.53
	LLsilverman	1872	0.94	0.08	0.01011	0.59
	NWscott	1647	0.82	0.13	0.23320	0.52
	LLscott	1861	0.93	0.09	0.01502	0.66
	NWcvls	1694	0.85	0.08	0.00575	218.05
	LLcvls	1872	0.94	0.08	0.00995	145.67
	DCKENW	1874	0.94	7.84×10^{32}	3.10955	0.17
	DCKELL	1874	0.94	7.88×10^{32}	3.10956	0.30

Table 5.4: All methods with $S_0 = 100, t = 0.5, \delta = 0.027$

For $Y_t = (V(t+\delta) - V(t))^2$, table 5.3 shows that the nested Monte Carlo method with $M = 100$, the nested Monte Carlo method with $M = 1000$, the polynomial regression method with the highest degree = 7, the polynomial regression method with the highest degree = 9, the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Scott's Rule of Thumb, and the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Least-Square Cross-Validation all have very small MSE. Within these methods, the nested Monte Carlo method with $M = 1000$ and the polynomial regression method with the highest degree = 7 have the lowest excess percentage, which is less than 1. However, since the run-time of the nested Monte Carlo method with $M = 1000$ is too large. Therefore, the polynomial regression method with the highest degree = 7 performs better. For $Y_t = IM(\omega, t)$, table 5.3 shows that the polynomial regression method with the highest degree = 8 because it has a comparatively smaller MSE, excess percentage and the run-time.

For $Y_t = (V(t+\delta) - V(t))^2$, table 5.4 shows that the nested Monte Carlo method with $M = 100$, the nested Monte Carlo method with $M = 1000$, the polynomial regression method with the highest degree = 7, the polynomial regression method with the highest degree = 8, the polynomial regression method with the highest degree = 9, the kernel regression with Nadaraya-Watson estimator and bandwidth is the Scott's Rule of Thumb, and the kernel regression with the Nadaraya-Watson estimator and bandwidth is the Silverman's Rule of Thumb all have very small MSE. Within these methods, the nested Monte Carlo method with $M = 1000$ and the polynomial regression method with the highest degree = 8 have the lowest excess percentage, which is less than 1. However, since the run-time of the nested Monte Carlo method with $M = 1000$ is too large. Therefore, the polynomial regression method with the highest degree = 8 performs better. For $Y_t = IM(\omega, t)$, table 5.4 shows that the polynomial regression method with the highest degree = 8 because it has a comparatively smaller MSE, excess percentage and the run-time.

In conclusion, regardless of the definition of Y_t, the polynomial regression performs better based on accuracy and efficiency. From the figures 5.1, 5.2, 5.3 and 5.4, we can find that as $V(t)$ increases, $IM(t)$ first increases rapidly and then increases slowly. This indicates that there is a kind of non-linear relationship between $V(\omega, t)$ and $IM(\omega, t)$, so it is reasonable that polynomial regression predicts more accurate $IM(\omega, t)$ at a faster rate.

Conclusion

In this paper, we focus on the calculate of the initial margin at time t. The initial margin is a 99% quantile of the change on the net change in the portfolio price within the period $(t, t+\delta]$ based on the definition of BCBS and IOSCO. There are parametric and non-parametric approaches for calculating the future initial margin, depending on whether the assumption of portfolio price changes is taken into account. Both methods take $V(\omega, t)$ as the independent variable, but the difference lies in the definition of Y_t. Specially, we define $Y_t = (V(t+\delta) - V(t))^2$ for parametric approach and $Y_t = IM(\omega, t)$ as non-parametric approach. In addition, for each approach, we can apply polynomial regression, kernel regression, and Dynamically Controlled KernelEstimation methods.

First of all, we assume the portfolio price follows a Black-Scholes model and the stock price follows a geometric Brownian motion. We determine the benchmark method according to the characteristic of the Black-Scholes model and the geometric Brownian motion. Moreover, we have a nested Monte Carlo method because the initial margin is calculated as the difference between the two portfolio prices. When we generate more paths from time t to $t+\delta$, the predicted initial margin at time t is closer to the benchmark value. However, the run-time is much longer as well.

The second method is the polynomial regression method. In this method, we set $V(\omega, t)$ as the regressor and the monomial function as the basis function. In both methods, as the highest degree increases, the prediction gets closer to the benchmark until the degree is greater than 10. Furthermore, the run-time for the polynomial regression is fast, especially in the parametric approach.

The next method is the kernel regression method, which contains the Nadaraya-Watson estimator and the local linear estimator. Also, the bandwidth of the method can be chosen among the Least-Square Cross-Validation method, Scott's rule of thumb, and Silverman's rule of thumb. Local linear estimators do a better prediction than the Nadaraya-Watson estimators regardless of the choice of the bandwidth. The kernel regression with the local linear estimator and the bandwidth is the Least-Square Cross-Validation method performs the best estimation, but the run-time is much slower.

The last method is the Dynamically Controlled Kernel Estimation method. The basis of this method is Gaussian Process regression, and this method uses local regression with control variate to determine the training data. Under these techniques, the predictions via the Dynamically Controlled Kernel Estimation has a relatively small difference with a faster speed.

For the European Call option, the polynomial method performs better compared with other methods. The reason is that there is a non-linear relationship between the portfolio price $V(\omega, t)$ and the initial margin $IM(\omega, t)$ when the portfolio price follows the Black-Scholes model and the stock price follows the geometric Brownian motion.

Appendix A

Figures

A.1 Chapter 5 Figures

For having a general view of the performance of all methods, we use different combinations of parameters to see the results.

A.1.1 When The Initial Stock Price Changes

When the initial stock price S_0, the time t and the close-out period change and all other parameters are fixed, the following figures show how all the methods perform in general.

Figure A.1 and figure A.2 show when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark for $t = 0.2$ and $t = 0.5$ regardless of the definition of Y_t and the values of δ. However, when $t = 0.8$ and $t = 0.9$, the methods have a similar trend but they are not close to the benchmark.

Figure A.3 and figure A.4 show when $S_0 = 80$ and all other parameters are held constant, most of the methods converge universally to the benchmark regardless of the definition of Y_t, the values of t and the values of δ. However, the predictions via few methods are not close to the benchmark values in all scenarios.

Figure A.5 and figure A.6 show when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark for variations in time t and close-out period δ regardless of the definition of Y_t. However, for every time t, the methods still converges to the benchmark at the beginning, but not at the end.

Figure A.7 and figure A.8 shows when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark for variations in time t and close-out period δ regardless of the definition of Y_t. Moreover, there appears to be a non-linear relationship between $V(\omega, t)$ and $IM(\omega, t)$.

A.1.2 When The Strike Price Changes

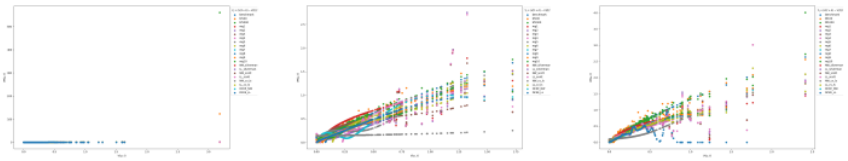
In this section, we change the setting of the parameters. Consider the strike price K, the time t and the close-out period change and all other parameters are fixed.

Figure A.9 and figure A.10 show when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark regardless of the definition of Y_t and the values of δ. However, the predicted results by all methods are not near the benchmark values at either end.

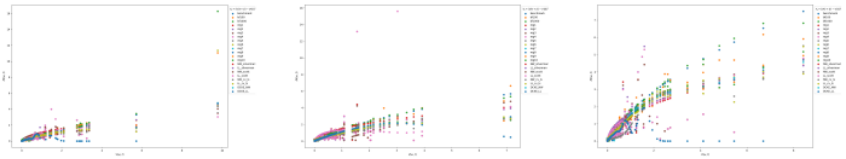
Figure A.11 and figure A.12 show when $S_0 = 80$ and all other parameters are held constant, most of the methods converge universally to the benchmark regardless of the definition of Y_t, the values of t and the values of δ.

Figure A.13 and figure A.14 show when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark for variations in time t and close-out period δ regardless of the definition of Y_t. However, for every time t, the methods still converges to the benchmark at the beginning, but not at the end.

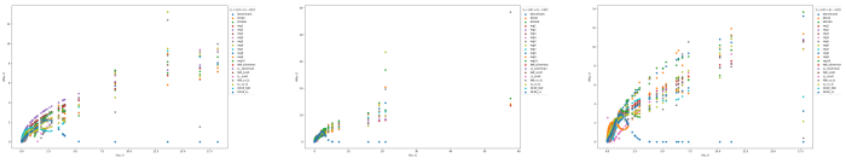
Figure A.15 and figure A.16 show when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark regardless of the definition of Y_t, the values of t and the values of δ. However, the predictions via few methods are not close to the benchmark values in all scenarios.



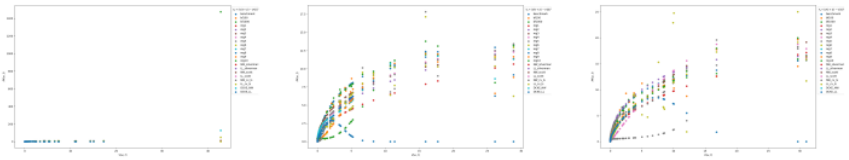
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

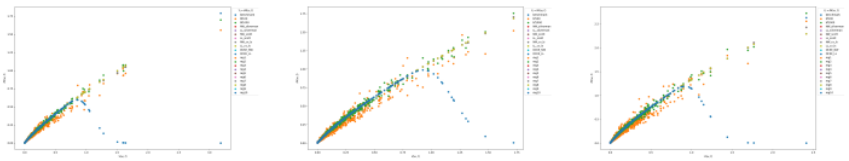


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

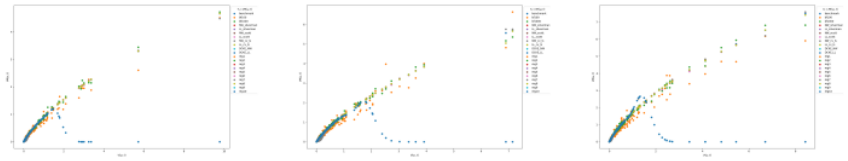


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

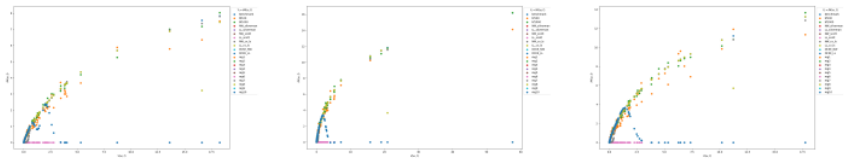
Figure A.1: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ under different methods with $K = 100$,$S_0 = 50$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



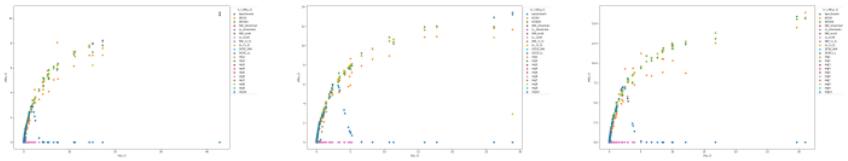
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

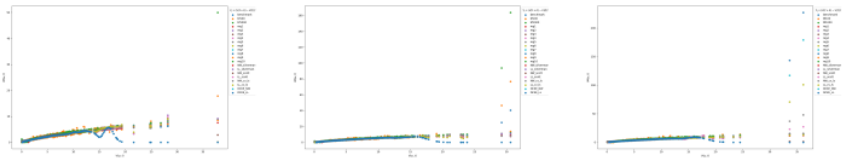


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

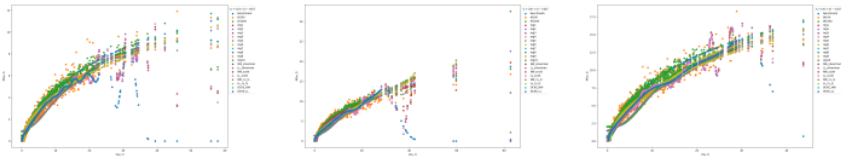


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

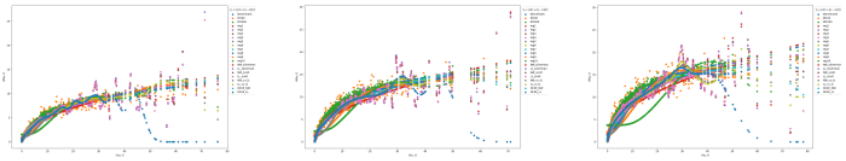
Figure A.2: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $K = 100$, $S_0 = 50$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



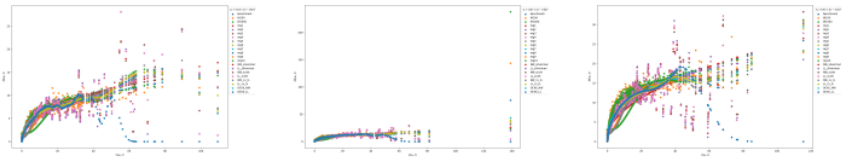
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

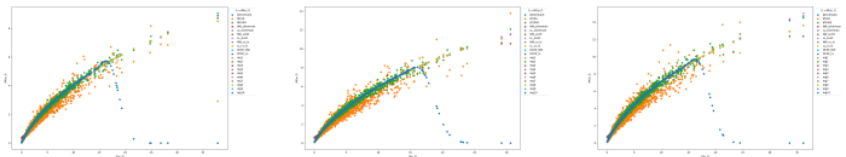


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

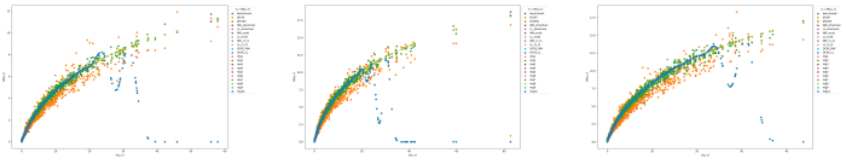


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

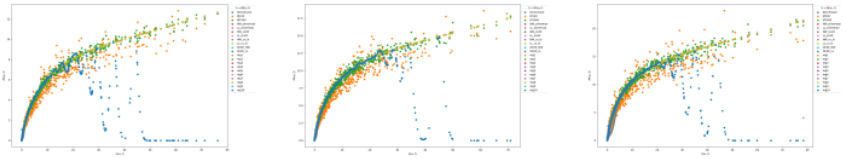
Figure A.3: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $K = 100$, $S_0 = 80$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



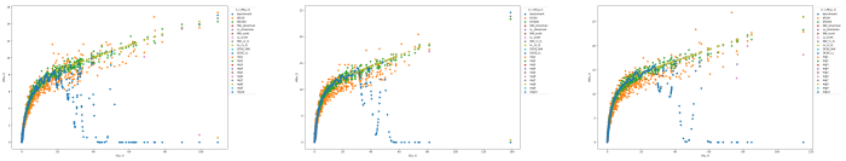
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

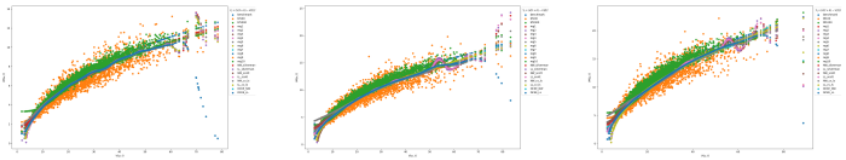


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

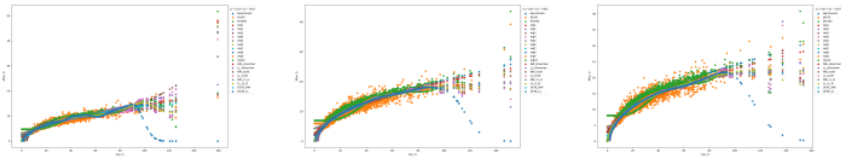


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

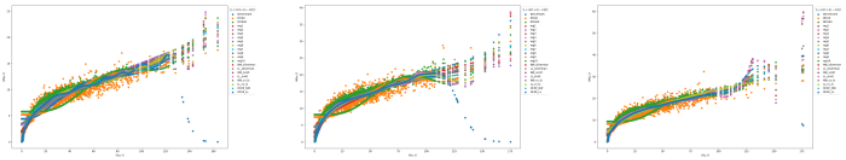
Figure A.4: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $K = 100$, $S_0 = 80$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



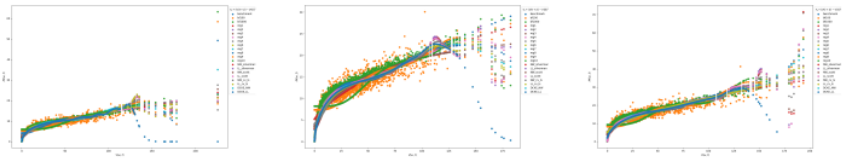
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

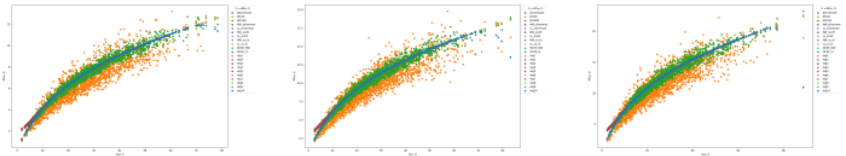


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

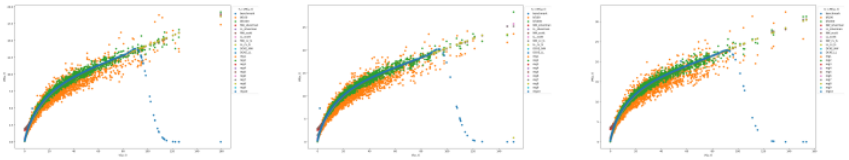


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

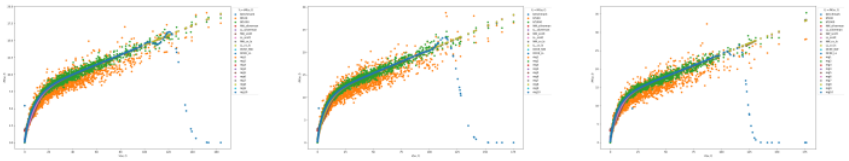
Figure A.5: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t+\delta) - V(t))^2$ under different methods with $K = 100$, $S_0 = 120$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



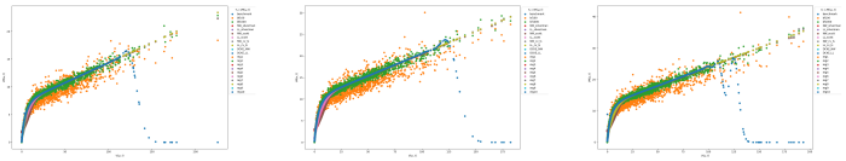
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

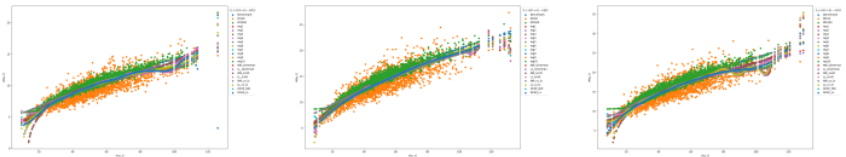


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

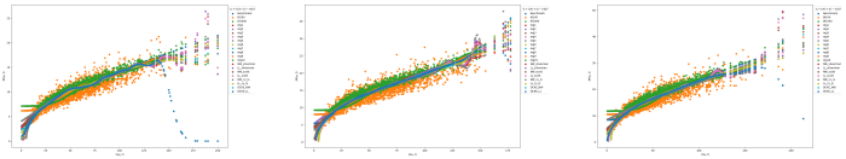


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

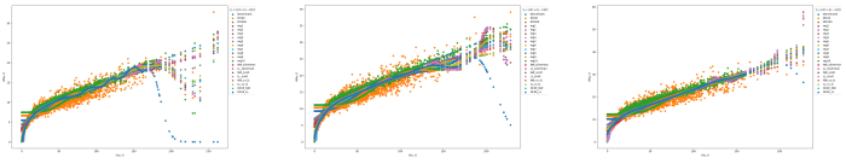
Figure A.6: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $K = 100$, $S_0 = 120$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



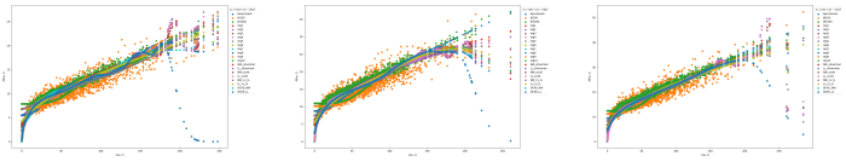
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

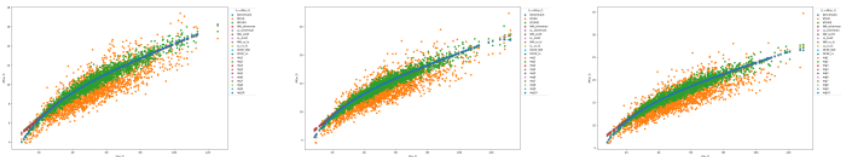


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

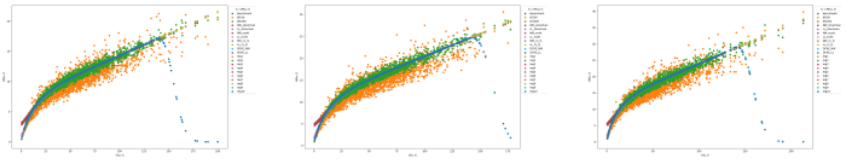


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

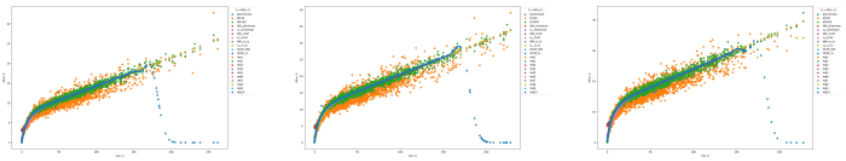
Figure A.7: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $K = 100$, $S_0 = 150$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



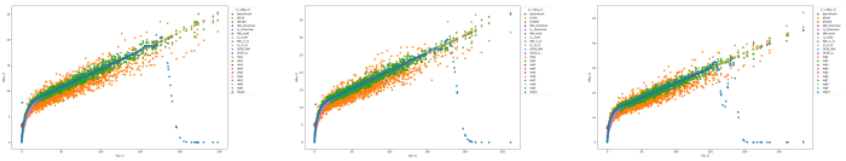
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$



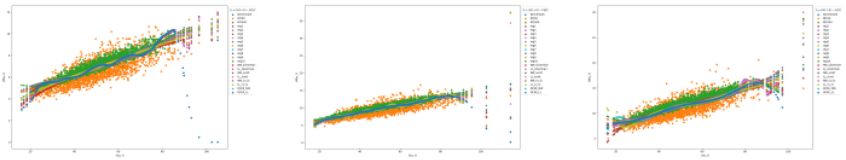
(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$



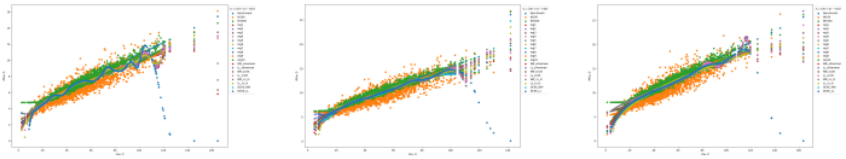
(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

Figure A.8: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $K = 100$, $S_0 = 150$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$

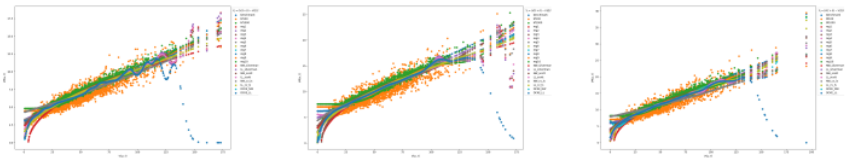
Figure A.17 and figure A.18 shows when $S_0 = 50$ and all other parameters are held constant, all methods converge universally to the benchmark for $t = 0.2$ and $t = 0.5$ regardless of the definition of Y_t and the values of δ. However, when $t = 0.8$ and $t = 0.9$, the methods have a similar trend but they are not close to the benchmark.



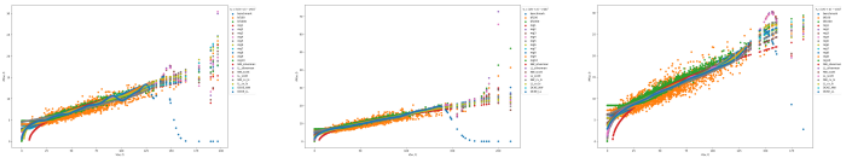
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

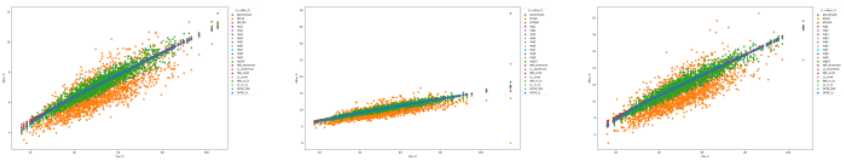


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

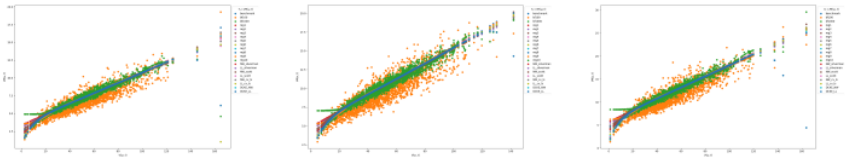


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

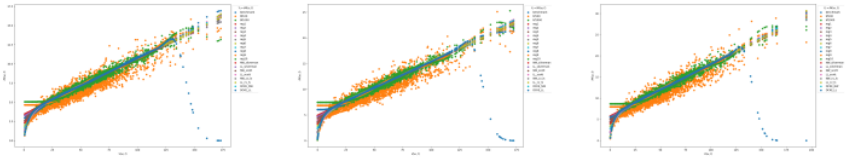
Figure A.9: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $S_0 = 100$, $K = 50$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



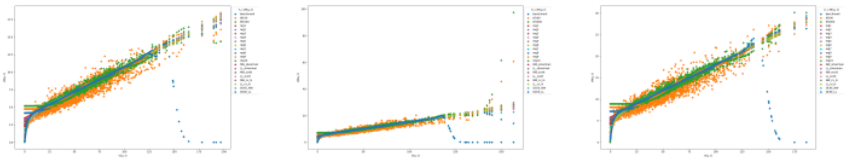
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

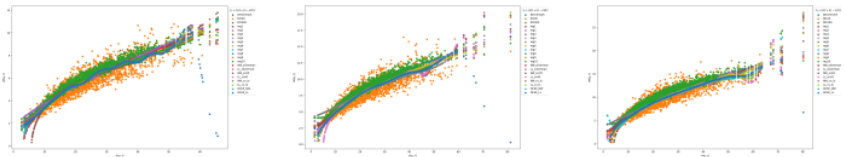


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

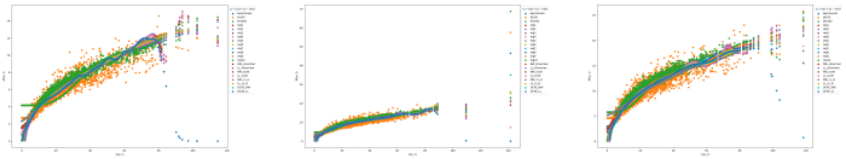


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

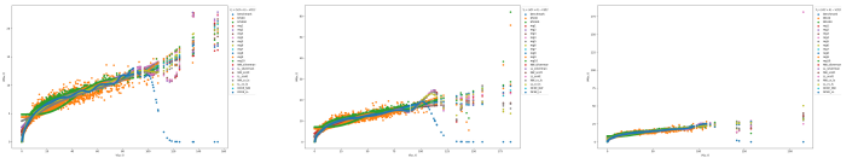
Figure A.10: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $S_0 = 100$, $K = 50$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



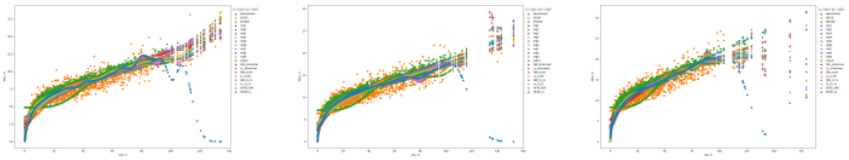
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

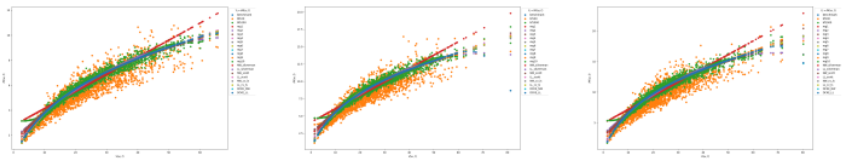


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

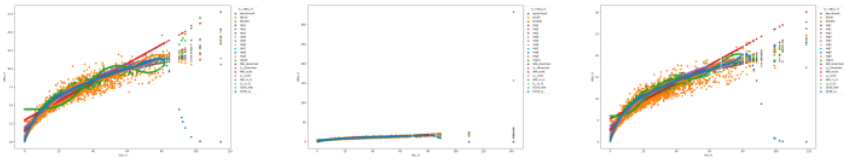


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

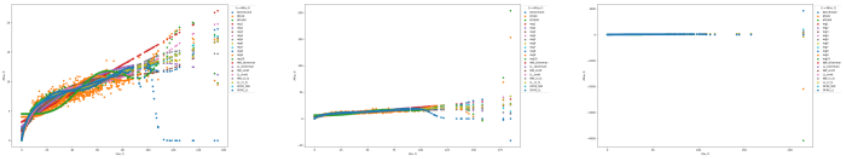
Figure A.11: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $S_0 = 100$, $K = 80$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



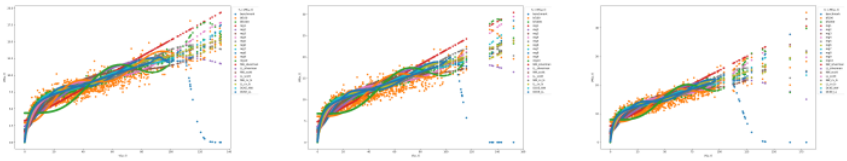
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

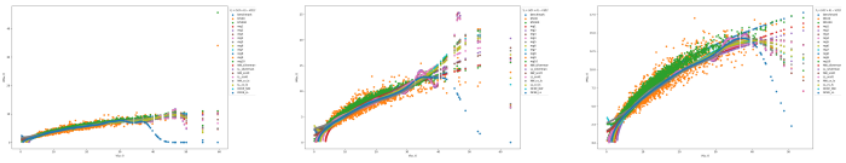


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

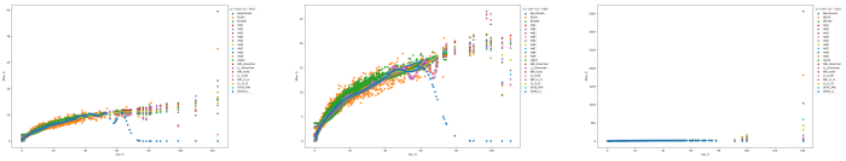


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

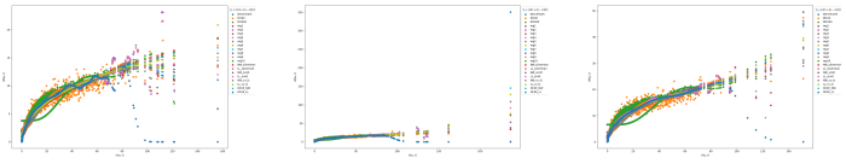
Figure A.12: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $S_0 = 100$, $K = 80$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



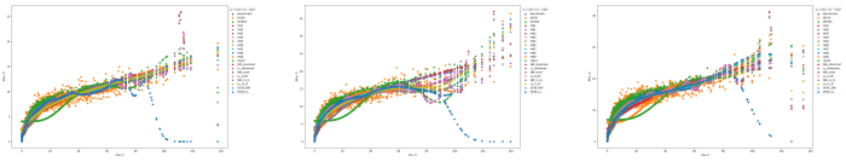
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

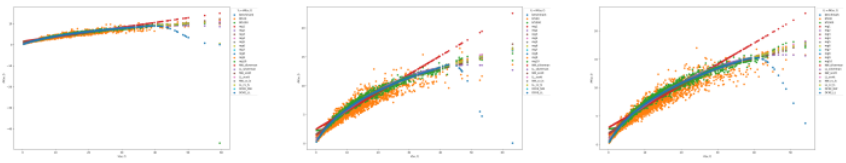


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

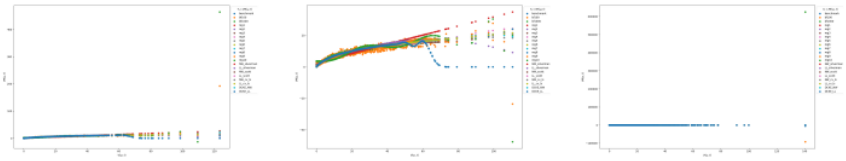


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

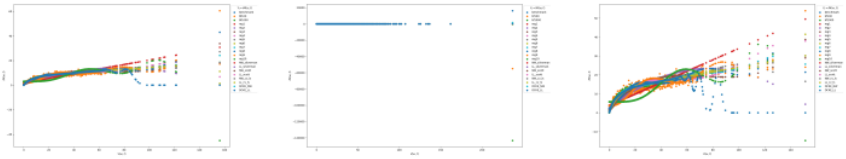
Figure A.13: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $S_0 = 100$, $K = 100$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



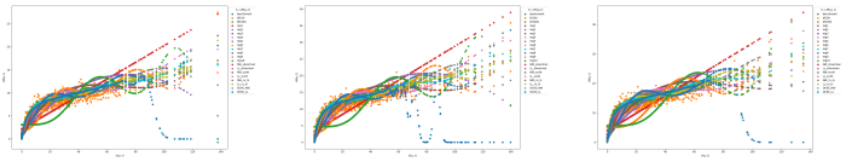
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

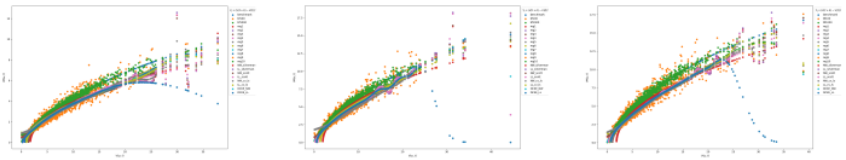


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

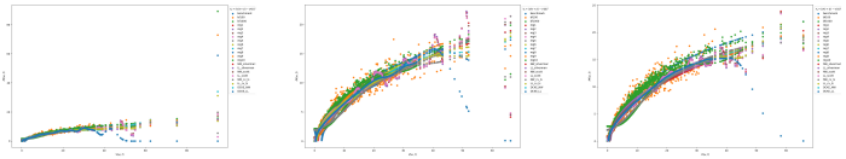


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

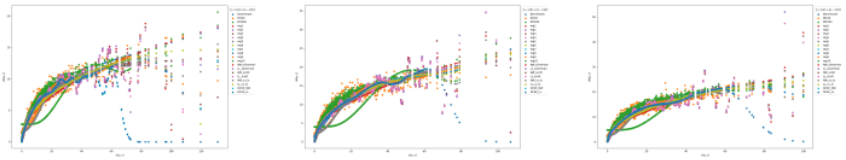
Figure A.14: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $S_0 = 100$, $K = 100$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



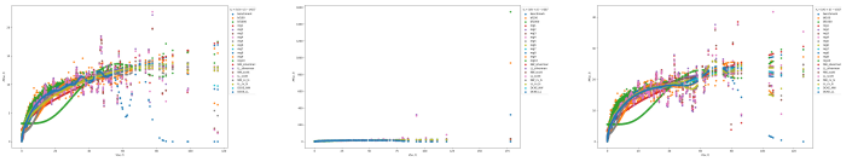
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

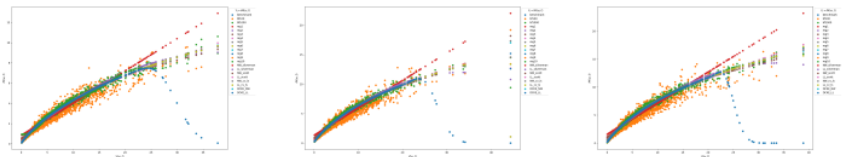


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

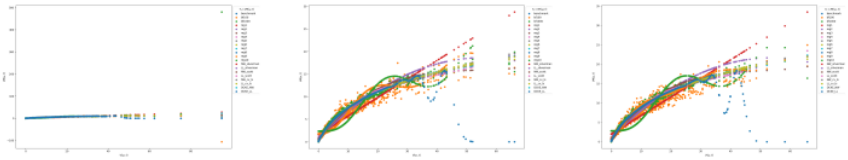


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

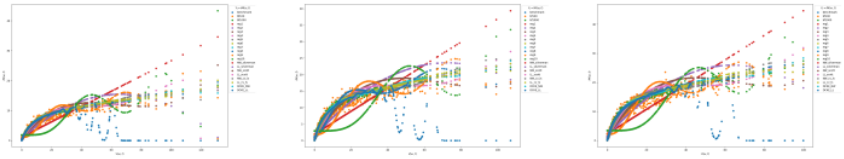
Figure A.15: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $S_0 = 100$, $K = 120$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



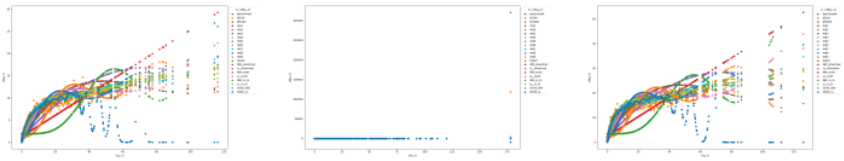
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

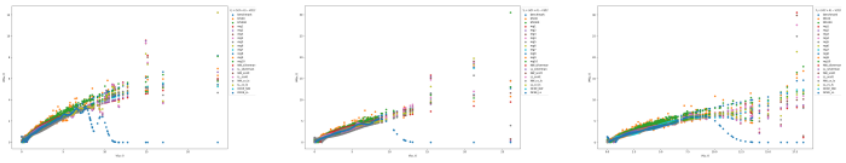


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

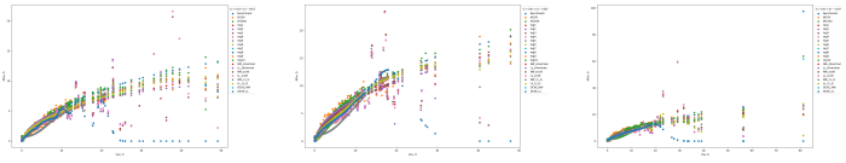


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

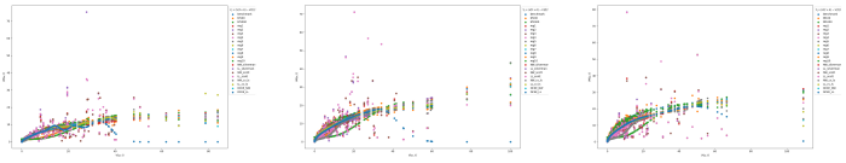
Figure A.16: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $S_0 = 100$, $K = 120$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



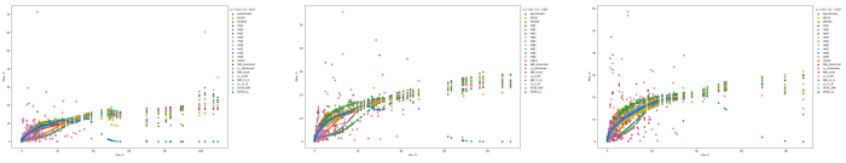
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$

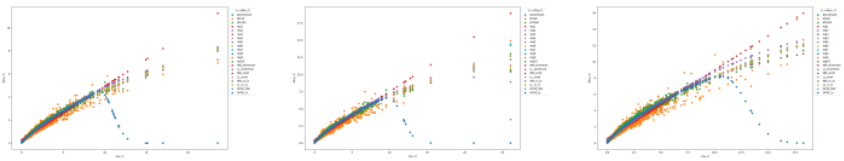


(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$

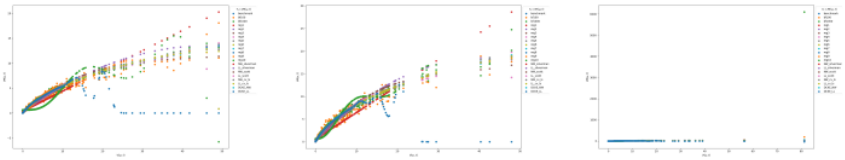


(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

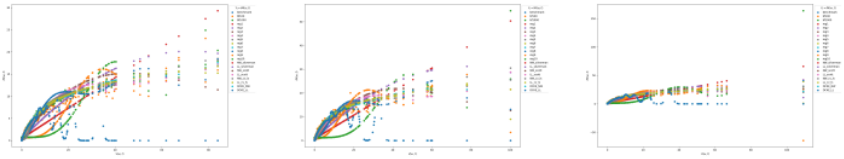
Figure A.17: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = (V(t + \delta) - V(t))^2$ under different methods with $S_0 = 100$, $K = 150$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$



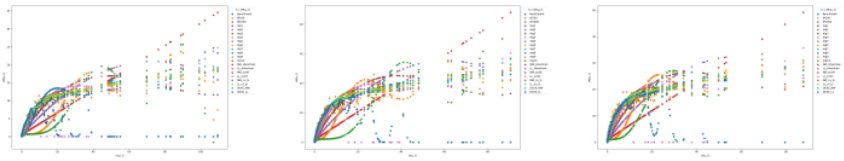
(a) $t = 0.2$ $\delta = 0.01$ (b) $t = 0.2$ $\delta = 0.02$ (c) $t = 0.2$ $\delta = 0.027$



(d) $t = 0.5$ $\delta = 0.01$ (e) $t = 0.5$ $\delta = 0.02$ (f) $t = 0.5$ $\delta = 0.027$



(g) $t = 0.8$ $\delta = 0.01$ (h) $t = 0.8$ $\delta = 0.02$ (i) $t = 0.8$ $\delta = 0.027$



(j) $t = 0.9$ $\delta = 0.01$ (k) $t = 0.9$ $\delta = 0.02$ (l) $t = 0.9$ $\delta = 0.027$

Figure A.18: $V(\omega, t)$ versus. $IM(\omega, t)$: set $Y_t = IM(\omega, t)$ under different methods with $S_0 = 100$, $K = 150$, $t = \{0.2, 0.5, 0.8, 0.9\}$, and $\delta = \{0.01, 0.02, 0.027\}$

Bibliography

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 1973.

[2] Lucia Cipolina-Kun, Simone Caenazzo, and Ksenia Ponomareva. Mathematical foundations of regression methods for approximating the forward initial margin. *Available at SSRN 3620523*, 2020.

[3] Eduardo García Portugués. Notes for predictive modeling.

[4] Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[5] Geoffrey S Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

[6] Joerg Kienitz, Nikolai Nowaczyk, and Nancy Qingxin Geng. Dynamically controlled kernel estimation. *Available at SSRN 3829701*, 2021.

[7] Stéphane Crépey and Matthew Dixon. Gaussian process regression for derivative portfolio modeling and application to cva computations. *arXiv preprint arXiv:1901.11081*, 2019.

[8] Justin Chan, Shengyao Zhu, and Boris Tourtzevitch. Practical approximation approaches to forecasting and backtesting initial margin requirements. *Available at SSRN 3079782*, 2017.

[9] Leif BG Andersen, Michael Pykhtin, and Alexander Sokol. Credit exposure in the presence of initial margin. *Available at SSRN 2806156*, 2016.

[10] Ursula Silveira Monteiro de Lima and Carlos Patricio Samanez. Complex derivatives valuation: applying the least-squares monte carlo simulation method with several polynomial basis. *Financial Innovation*, 2(1):1–14, 2016.

[11] Peter Caspers, Paul Giltinan, Roland Lichters, and Nikolai Nowaczyk. Forecasting initial margin requirements: A model evaluation. *Journal of Risk Management in Financial Institutions*, 10(4):365–394, 2017.

[12] Leif BG Andersen, Michael Pykhtin, and Alexander Sokol. Rethinking the margin period of risk. *Journal of Credit Risk*, 13(1), 2017.

[13] Fabrizio Anfuso, Daniel Aziz, Paul Giltinan, and Klearchos Loukopoulos. A sound modelling and backtesting framework for forecasting initial margin requirements. *Available at SSRN 2716279*, 2017.

[14] Steffen Kuehn. Kernel regression by mode calculation of the conditional probability distribution. *arXiv preprint arXiv:0811.3499*, 2008.

Dong_Jingxuan_01992119

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72