



Linnæus University

Sweden

Degree project

Merton Jump-Diffusion Modeling of Stock Price Data



Author: Furui Tang
Supervisor: Roger Pettersson
Examiner: Astrid Hilbert
Date: 2018-09-25
Course Code: 2MA41E
Subject: Applied Mathematics
Level: Bachelor
Department Of Mathematics



LINNAEUS UNIVERSITY

Merton Jump-Diffusion Modeling of Stock Price Data

Author:

Furui TANG

Supervisor:

Roger PETTERSSON

September 25, 2018

Contents

1	Abstract	3
2	Introduction	3
3	Basic Tools	3
3.0.1	Logarithmic return	4
3.1	Some used probability definitions	4
3.1.1	Distribution functions	4
3.1.2	Expectation & Variance	5
3.1.3	A normal distribution	5
3.2	Brownian motion	5
4	Black-Scholes Model	6
4.1	Introduction to the Black-Scholes model	6
4.2	Simulation of the BS model	7
4.3	Estimation of the parameters μ and σ for the BS model	8
4.4	Comparison of the empirical log-returns with the BS-modeled log-returns	9
5	The Poisson Process	11
6	The Compound Poisson Process	12
7	Merton Jump-Diffusion Model	13
7.1	Introduction to the Merton Jump-Diffusion model	13
7.2	Simulations for the MJD model	14
7.3	Estimate of the parameters μ_d , σ_d , λ , μ_j and σ_j	15
7.3.1	One initial parameter estimate	15
7.3.2	Maximum likelihood estimation	18
7.4	Comparison of the empirical log-returns with the MJD- modeled log-returns	20
8	Comparison of the BS Model with the MJD Model for Empirical stock prices	21
9	Conclusion	25
10	Reference	26

Appendices	26
A Matlab codes for the figures	26
B Matlab codes for parameter estimations	32

1 Abstract

Abstract

In this thesis, we investigate two stock price models, the Black-Scholes (BS) model and the Merton Jump-Diffusion (MJD) model. Comparing the logarithmic return of the BS model and the MJD model with empirical stock price data, we conclude that the Merton Jump-Diffusion Model is substantially more suitable for the stock market. This is concluded visually not only by comparing the density functions but also by analyzing mean, variance, skewness and kurtosis of the log-returns.

One technical contribution to the thesis is a suggested decision rule for initial guess of a maximum likelihood estimation of the MJD-modeled parameters.

2 Introduction

In the financial market, there are many risks for investors, for example stock price. For avoiding risks, investors can analyze historical price to predict the future distribution of stock prices by using suitable models. In this thesis, the Black-Scholes and the Merton Jump-Diffusion model, are studied as models for stock prices. Both models can be used to analyze trends of stock prices in the financial market. The MJD model is a generalization of the BS model. Here, it is investigated whether the MJD model is significantly more fit to empirical stock price data.

To approach this problem, we need some basic probability and stochastic process theory, such as mean, variance, normal distribution and probability density function, et cetera. By comparing the density functions of the two models, the Merton Jump-Diffusion model is found to be significantly more suitable for real stock prices, which is further confirmed by investigating the moments.

3 Basic Tools

To analyze the two models, we recall some setups in finance and probability.

3.0.1 Logarithmic return

Definition 1. Let S_t be a stock price at time t . During the time increment Δt , the logarithmic return $R_{\Delta t}$ of the stock price is defined as

$$R_{\Delta t} = \ln\left(\frac{S_{t+\Delta t}}{S_t}\right).$$

For convenience, we rewrite the logarithmic return in a short form: log-return. The log-return $R_{\Delta t}$ depends not only on time t but also on the time increment Δt . However, the distribution of the log-return does not depend on the time for the two models. Therefore, the time t is suppressed for the log-return $R_{\Delta t}$.

A well-known empirical fact is that for small time increment Δt , the log-return $R_{\Delta t}$ is approximately equal to the arithmetic return $(S_{t+\Delta t} - S_t)/S_t$. For the two stated models here, the log-return is more convenient to analyze than the arithmetic return. Therefore, in this thesis we will compare the log-return of empirical data with the two modeled log-returns to decide whether the MJD model can be much more suitable for the real stock market than the BS model.

3.1 Some used probability definitions

3.1.1 Distribution functions

Definition 2. For a random variable X , the cumulative distribution function (CDF) $F_X(x)$ is defined as

$$F_X(x) = P(X \leq x).$$

Definition 3. A random variable X is said to be continuous if its CDF can be written in the integral form

$$F_X(x) = \int_{-\infty}^x f_X(t) dt.$$

Here, the integrand $f_X(t)$ is called the probability density function of X .

The integral in Definition 3 is a *Lebesgue integral*, which is an extension of the classical *Riemann integral*.

3.1.2 Expectation & Variance

Definition 4. For a continuous random variable X , the expectation $E(X)$ is defined as

$$E(X) = \int_{-\infty}^{\infty} x f_X(t) dx.$$

Definition 5. For a continuous random variable X , the variance $\text{Var}(X)$ is defined as

$$\text{Var}(X) = E(x - E(X))^2 = \int_{-\infty}^{\infty} (x - E(X)) f_X dx.$$

3.1.3 A normal distribution

Definition 6. The probability density function (PDF) of a normal random variable X , denoted as $X \sim N(\mu, \sigma^2)$, is defined by:

$$f_X(x) = \varphi(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

3.2 Brownian motion

Definition 7. A stochastic process $\{W_t\}_{t \geq 0}$ is called a standard Brownian motion, if the following properties are satisfied:

- (1) $W_0 = 0$.
- (2) Independent increments: for every time point $0 \leq t_1 \leq t_2 \leq \dots \leq t_n$, the increments of W : $W_{t_n} - W_{t_{n-1}}, W_{t_{n-1}} - W_{t_{n-2}}, \dots, W_{t_2} - W_{t_1}$ are independent random variables.
- (3) Normal distribution: for every time point $0 \leq t_1 \leq t_2 \leq \dots \leq t_n$, the increments of W : $W_{t_n} - W_{t_{n-1}}, W_{t_{n-1}} - W_{t_{n-2}}, \dots, W_{t_2} - W_{t_1}$ are normal distributed with mean zero and variance $(t_n - t_{n-1}), (t_{n-1} - t_{n-2}), \dots, (t_2 - t_1)$, respectively.
- (4) With probability one, the function $t \mapsto W_t$ is a continuous function for every t .

For more about the Brownian motion, see for instance, Chapter 2 of reference [5]. Note that the increment $\Delta W_t = W_{t+\Delta t} - W_t \sim N(0, \Delta t) \sim \sqrt{\Delta t} N(0, 1)$, which means that $(W_{t+\Delta t} - W_t)/\sqrt{\Delta t} \sim N(0, 1)$.

Since $\Delta W_t \sim \sqrt{\Delta t}N(0,1)$, a standard Brownian motion is easily simulated. A path of a standard Brownian motion is illustrated in Figure 1.

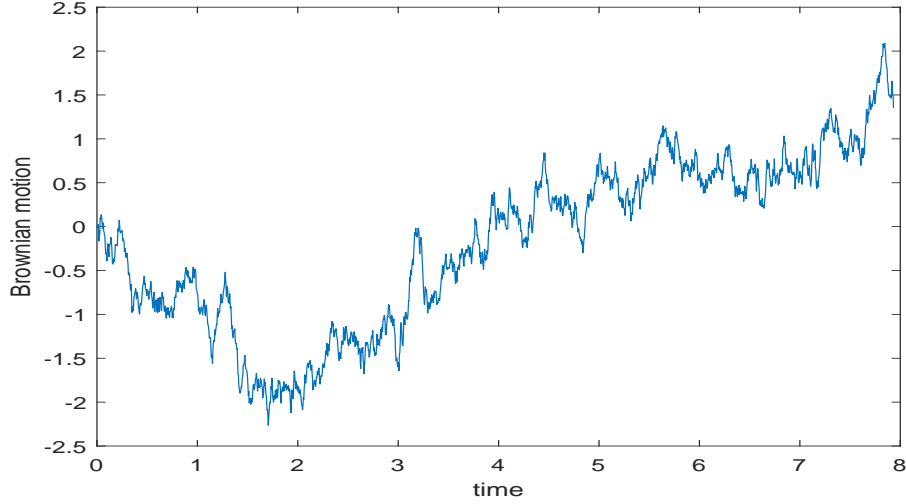


Figure 1: A simulation of a standard Brownian motion.

4 Black-Scholes Model

4.1 Introduction to the Black-Scholes model

In the financial market, the Black-Scholes model is a widely used model. For convenience, we abbreviate the Black-Scholes model as the BS model. The stock price S_t under the BS model is:

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}. \quad (1)$$

In the formula (1), S_0 is the stock price at time zero, μ is a drift rate, and σ is named as a volatility which expresses the uncertainty of the stock price. From the BS model (1), the logarithm of the stock price is:

$$\begin{aligned} \ln(S_t) &= \ln(S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}) \\ &= \ln(S_0) + \ln(e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}) \\ &= \ln(S_0) + (\mu - \frac{\sigma^2}{2})t + \sigma W_t. \end{aligned} \quad (2)$$

By Definition 1, we can obtain the log-return $R_{\Delta t}$ of the BS-modeled stock price as

$$\begin{aligned}
R_{\Delta t} &= \ln\left(\frac{S_{t+\Delta t}}{S_t}\right) \\
&= \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma(W_{t+\Delta t} - W_t) \\
&\sim N\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma^2\Delta t\right) \\
&= \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}N(0, 1),
\end{aligned} \tag{3}$$

which means that

$$\frac{R_{\Delta t} - \left(\mu - \frac{\sigma^2}{2}\right)\Delta t}{\sigma\sqrt{\Delta t}} \sim N(0, 1).$$

4.2 Simulation of the BS model

From (1) and simulation of the standard Brownian motion $\{W_t\}_{t \geq 0}$, a sequence of log-returns $R_{\Delta t}$ of the BS model is easily generated. Simulation codes for the BS model can be seen in Appendix A.

For the given parameters $\mu = 0.15$ and $\sigma = 0.19$, and $S_0 = 416$, five paths of simulated stock prices under the BS model are illustrated in Figure 2.

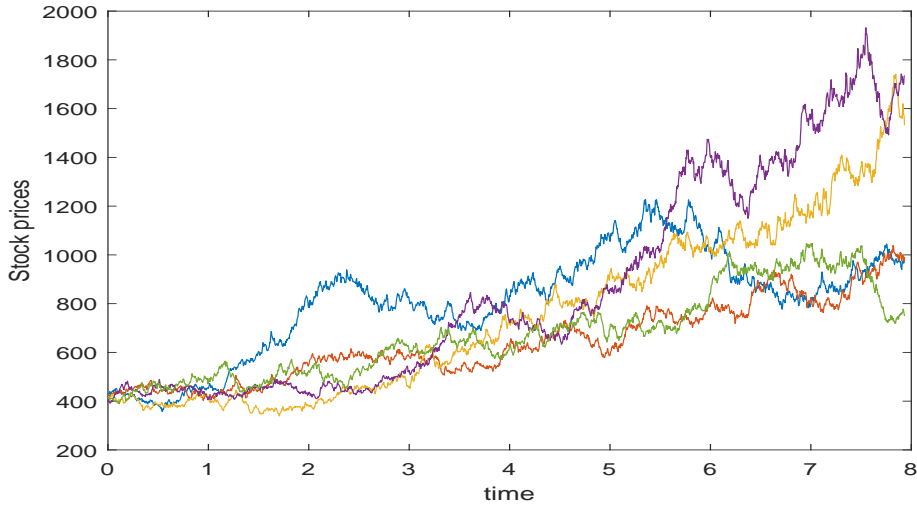


Figure 2: Five paths of BS simulated stock prices with $S_0 = 416$, $\mu = 0.15$, $\sigma = 0.19$.

4.3 Estimation of the parameters μ and σ for the BS model

To fit empirical data to the BS model, it is necessary to estimate the parameters μ and σ .

Let $E(R_{\Delta t})$ and $\text{Var}(R_{\Delta t})$ be the mean and variance of the log-return of the BS-modeled stock price, respectively. By Definitions 4, 5 and formula (3), we can get the mean and variance of the log-return $R_{\Delta t}$:

$$\begin{cases} E(R_{\Delta t}) = (\mu - \frac{\sigma^2}{2})\Delta t \\ \text{Var}(R_{\Delta t}) = \sigma^2\Delta t. \end{cases} \quad (4)$$

From (4), based on a log-return data sequence, it is natural to estimate μ and σ by

$$\begin{cases} \hat{\mu} = \frac{2\widehat{E}(R_{\Delta t}) + \widehat{\text{Var}}(R_{\Delta t})\Delta t}{2\Delta t} \\ \hat{\sigma}^2 = \frac{\widehat{\text{Var}}(R_{\Delta t})}{\Delta t}, \end{cases} \quad (5)$$

where $\widehat{E}(R_{\Delta t})$ is the sample mean of the empirical log-returns, and $\widehat{\text{Var}}(R_{\Delta t})$ is the sample variance of the empirical log-returns.

Here, the used empirical data is historical stock prices of the Google company from 2013-03-01 to 2018-03-05, with 1260 trading days [10]. The unit time is one year with for simplification 252 trading days per year, which means that the time difference between the data points is $1/252$. Hence, the estimated parameters under this empirical data are $\mu = 0.2171$ and $\sigma = 0.2210$.

The following MATLAB code is used to estimate the parameters μ and σ for the BS model for the Google stock prices.

```
S=csvread('GOOG.csv',1,5,[1,5,1260,5]);
dt=1/252;
R=diff(log(S),1); % the return of the emprical stock
    price
muhat=mean(R)/dt+var(R)/(2*dt);
sigmahat=sqrt(var(R)/dt);
```

4.4 Comparison of the empirical log-returns with the BS-modeled log-returns

According to the estimation of parameters, we import the estimated parameters $\hat{\mu} = 0.2171$ and $\hat{\sigma} = 0.2210$ into the BS model. Then we simulate log-returns for the adapted BS model, and compare it with the empirical log-returns. The simulated BS-modeled log-returns and the empirical log-returns are illustrated in Figure 3.

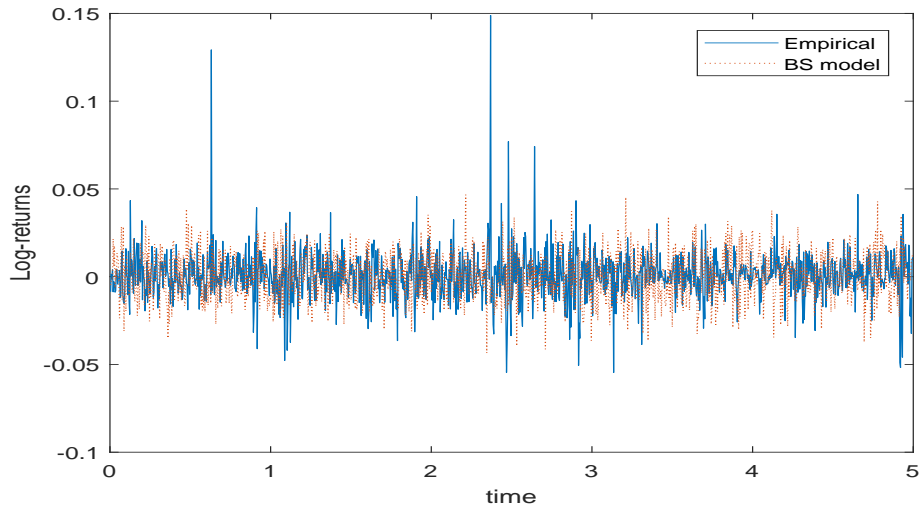


Figure 3: Comparison of simulated BS-modeled log-returns with the empirical Google log-returns.

In Figure 3, the BS-modeled log-returns seem to be similar to the empirical log-returns. However the empirical log-returns have several larger spikes. Note also that most empirical log-returns are smaller in size than the BS-modeled log-returns. That may indicate that the seemingly randomness of the empirical log-returns are from two parts, one is giving large spikes which do not often appear, the another one which appear most of the time typically gives smaller log-returns than the BS-modeled log-returns. Consequently, there may be a need for another model to analyze stock prices than the BS model.

To visualize the difference between the modeled log-returns and the empirical log-returns, a proper clearer way is to compare their probability density functions. Hence, we consider the probability density functions of the BS-modeled and the empirical log-returns.

From (4), by Definition 6, the probability density function of the BS-modeled log-return $R_{\Delta t}$ is:

$$\varphi(x | (\mu - \frac{\sigma^2}{2})\Delta t, \sigma^2\Delta t) = \frac{1}{\sqrt{2\pi\sigma^2\Delta t}} e^{-\frac{(x - (\mu - \frac{\sigma^2}{2})\Delta t)^2}{2\sigma^2\Delta t}}, \quad \text{for } x \in \mathbb{R}. \quad (6)$$

The PDF of the empirical log-returns is estimated by a kernel density estimation of the form:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - r_i}{h}\right),$$

where r_1, r_2, \dots, r_n are the empirical returns, K is the standard Gaussian density $K(x) = \varphi(x|0, 1)$, and h is a suitable bandwidth.

Then, the PDF of the BS-modeled log-returns is compared with the kernel density estimation of the empirical log-returns. The bandwidth h is here selected by default by MATLAB. The densities are illustrated in Figure 4.

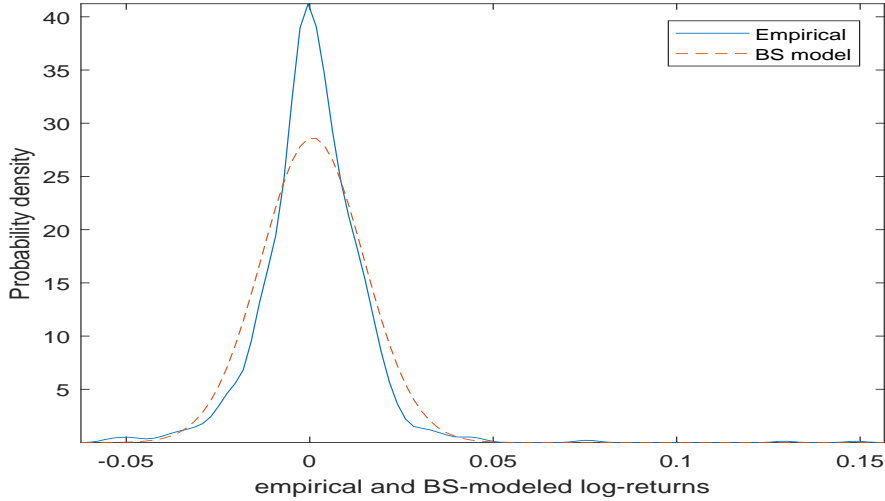


Figure 4: Kernel density estimation of the empirical log-returns and the PDF of the BS-modeled log-returns.

Since the BS-modeled log-returns are normal distributed, if the BS model is suitable for the real stock price, the empirical log-returns should be at least moderately close to a normal density. Otherwise, the BS model does not suit the real stock price.

Hence, we apply two tests, the Jarque-Bera test¹ and the Kolmogorov-Smirnov test², to judge whether the probability density function of empirical data is normal distributed. By importing the Google data into the two tests, the results for both tests are that a normal distribution hypothesis of the log-return of the Google data is rejected. Therefore, the BS model is not very suitable for the real stock market.

That motivates us to consider the Merton Jump-Diffusion (MJD) model which is a generalization of the BS model that allows jumps of the stock price. In this model, we should pay attention to the occasional larger spikes and the frequent small size log-returns, illustrated in Figure 3, which may indicate that the log-returns are frequently quite small but by and then jumps of the stock price occurs. To analyze the jumps of the stock price, we first need to study the Poisson process and the compound Poisson processes.

5 The Poisson Process

The jumps for the MJD model occur according to a Poisson process. Therefore the Poisson process is recalled here.

Definition 8. Let $\{\tau_i\}_{i \geq 1}$ be a sequence of exponential random variables with parameter λ . Let $T_n = \sum_{i=1}^n \tau_i$. Then the Poisson process $\{N_t\}_{t \geq 0}$ is defined as

$$N_t = \sum_{n \geq 1} 1_{t \geq T_n},$$

where the intensity λ is the expected number of jumps per unit time [4].

The Poisson process can be simulated in different ways. Note that

$$\Delta N_t = N_{t+\Delta t} - N_t$$

is Poisson distributed with mean $\lambda \Delta t$, i.e. $P(\Delta N_t = k) = \frac{(-\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}$, for $k = 0, 1, 2, \dots$. The implementation of ΔN_t is here given by the MATLAB Poisson random number generator.

A simulated Poisson process path $\{N_t\}_{t \geq 0}$ with intensity $\lambda = 15$ is displayed in Figure 5. For Poisson process simulation codes, see Appendix A.

¹In statistics, the null hypothesis of the Jarque-Bera test is that the data comes from a normal distribution. The test rejects the null hypothesis at the 5% significance level, which indicates that the data is far from being normal distributed and the null hypothesis is rejected.

²The null hypothesis of Kolmogorov-Smirnov test here is also that the data comes from a standard normal distribution.

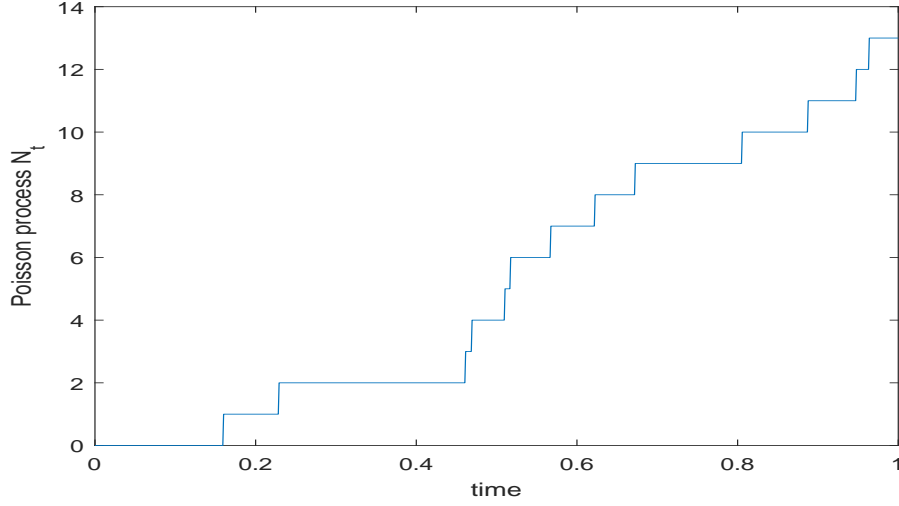


Figure 5: A simulated Poisson process path with intensity $\lambda = 15$.

For the simulation technique applied in Figure 5, $\Delta t = \frac{1}{1000}$. The cumulative number of jumps are only registered at time points $0, \Delta t, 2\Delta t, 3\Delta t, \dots, 1$. In this simulation, the number of jumps is 14, which is not far from the expected number of jumps $\lambda = 15$.

6 The Compound Poisson Process

For the MJD model not only the jump times but also the jump heights are random. The randomness is determined by a compound Poisson process which consequently is here introduced.

Definition 9. Let $\{Q_i\}_{i \geq 1}$ be a sequence of independent and identically distributed random variables, and $\{N_t\}_{t \geq 0}$ be a Poisson process with intensity parameter λ . Then the compound Poisson process $\{Y_t\}_{t \geq 0}$ is defined by

$$Y_t = \sum_{i=1}^{N_t} Q_i,$$

with jump intensity λ , [4].

If $N_t = 0$, then Y_t is defined as $Y_t = \sum_{i=1}^0 Q_i = 0$.

To simulate the compound Poisson process, the sum of jumps in $[t, t + \Delta t]$ has the same distribution as $\sum_{i=1}^{\Delta N_t} Q_i$, where we recall that the number of

jumps ΔN_t is Poisson distributed with mean $\lambda \Delta t$.

To simulate $\sum_{i=1}^{\Delta N_t} Q_i$, at first, ΔN_t is simulated from the Poisson distribution with mean $\lambda \Delta t$. Given such random integer $\Delta N_t = k$ say, $\sum_{i=1}^{\Delta N_t} Q_i = \sum_{i=1}^k Q_i$ is generated. If the Q_i is normal distributed, $Q_i \sim N(\mu_j, \sigma_j^2)$, where the j represents jumps, then $\sum_{i=1}^k Q_i \sim N(k\mu_j, k\sigma_j^2)$.

A simulated compound Poisson process N_t with intensity $\lambda = 15$ and normal distributed jumps $N(0.5, 1)$ is illustrated in Figure 6. For compound Poisson process simulation codes, see Appendix A.

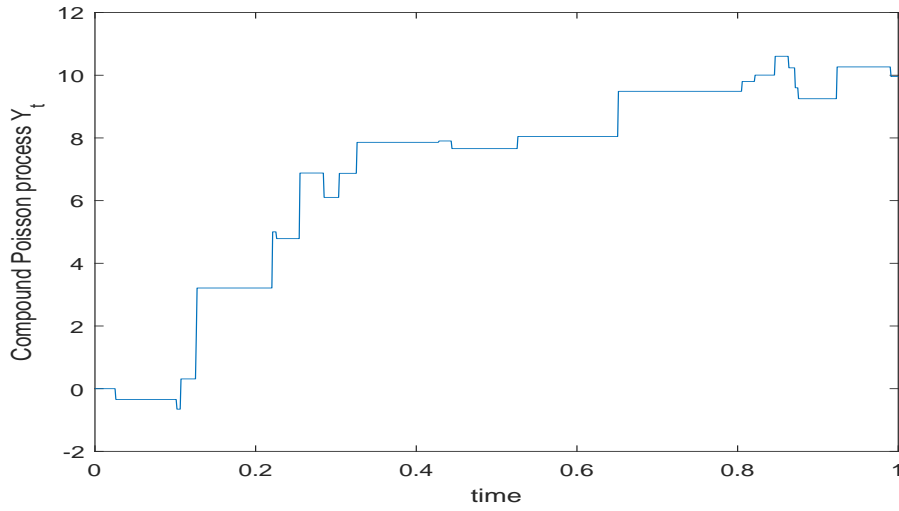


Figure 6: A simulated compound Poisson process path with jump intensity 15 and a normal jump height distribution $N(0.5, 1)$.

Here, the number of jumps is 17. The mean jump height 0.5 in the compound Poisson process implies that the jumps are mostly upwards.

7 ~~Merton Jump-Diffusion Model~~

7.1 Introduction to the Merton Jump-Diffusion model

With Chapters 4 and 5, a basic understanding of the Poisson processes and the compound Poisson processes were given. Now we are ready to present the Merton Jump-Diffusion (MJD) model.

The stock price S_t under the MJD model is:

$$S_t = S_0 e^{(\mu_d - \frac{\sigma_d^2}{2})t + \sigma_d W_t + \sum_{i=1}^{N_t} Q_i}, \quad (7)$$

where the μ_d is named as diffusion drift, σ_d is named as the volatility of diffusion, $\{W_t\}_{t \geq 0}$ is a standard Brownian motion, $\{\sum_{i=1}^{N_t} Q_i\}_{t \geq 0}$ is a compound Poisson process with normal distributed jumps $N(\mu_j, \sigma_j^2)$ and intensity λ . Here, the index d represents the diffusion of the MJD model, and j represents the jumps of the MJD model.

From the MJD model (7), the logarithm of the stock price is

$$\begin{aligned} \ln(S_t) &= \ln \left(S_0 e^{(\mu_d - \frac{\sigma_d^2}{2})t + \sigma_d W_t + \sum_{i=1}^{N_t} Q_i} \right) \\ &= \ln(S_0) + \ln \left(e^{(\mu_d - \frac{\sigma_d^2}{2})t + \sigma_d (W_{t+\Delta t} - W_t) + \sum_{i=1}^{N_t} Q_i} \right) \\ &= \ln(S_0) + (\mu_d - \frac{\sigma_d^2}{2})t + \sigma_d (W_{t+\Delta t} - W_t) + \sum_{i=1}^{N_t} Q_i. \end{aligned} \quad (8)$$

Then by Definition 1, we can get the log-return $R_{\Delta t}$ of the MJD-modeled stock price as

$$\begin{aligned} R_{\Delta t} &= \ln \left(\frac{S_{t+\Delta t}}{S_t} \right) \\ &= (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d (W_{t+\Delta t} - W_t) + \sum_{i=N_t}^{N_{t+\Delta t}} Q_i, \end{aligned} \quad (9)$$

where $\Delta W_t = W_{t+\Delta t} - W_t$ is a standard Brownian motion increment, Q_i are independent normal distributed with mean μ_j and variance σ_j^2 , and $\Delta N_t = N_{t+\Delta t} - N_t$ is a Poisson random variable with mean $\lambda \Delta t$. We observe that $\sum_{i=N_t}^{N_{t+\Delta t}} Q_i$ has the same distribution as $\sum_{i=1}^{\Delta N_t} Q_i$, where once again $\Delta N_t = N_{t+\Delta t} - N_t$.

7.2 Simulations for the MJD model

Since simulations of standard Brownian motion increments ΔW_t and compound Poisson process increments $\sum_{i=N_t}^{N_{t+\Delta t}} Q_i$ are already estimated, it is easy to simulate the log-return $R_{\Delta t}$ (9) of the MJD model. Then from (8), we can simulate MJD-modeled stock prices. Simulation codes for the MJD model can be seen in Appendix A.

For the given parameters $\mu_d = 0.17$, $\sigma_d = 0.055$, $\lambda = 10$, $\mu_j = 0.005$ and $\sigma_j = 0.08$, five paths of simulated stock prices of the MJD model are displayed in Figure 7.

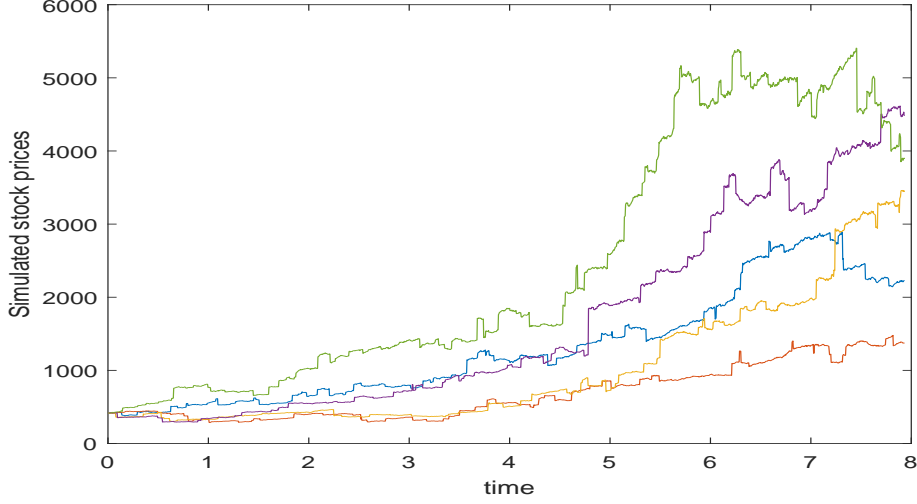


Figure 7: Five paths of MJD simulated stock prices, where $\mu_d = 0.17$, $\sigma_d = 0.055$, $\lambda = 10$, $\mu_j = 0.005$ and $\sigma_j = 0.08$.

7.3 Estimate of the parameters μ_d , σ_d , λ , μ_j and σ_j

The expectation and variance of log-return $R_{\Delta t}$ (9) of the MJD-modeled stock price are:

$$\begin{cases} E(R_{\Delta t}) = (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \mu_j\lambda\Delta t \\ \text{Var}(R_{\Delta t}) = \sigma_d^2\Delta t + (\sigma_j^2 + \mu_j^2)\lambda\Delta t. \end{cases} \quad (10)$$

For a complete proof of (10), see reference [3].

7.3.1 One initial parameter estimate

To estimate the five parameters μ_d , σ_d , λ , μ_j and σ_j , we would like to use the maximum likelihood estimation (MLE) method³. Since there are five parameters in the MJD model, and as far as the author knows, there are no analytic expressions of the optimal parameter values, we will use the MATLAB code *fminsearch* to estimate optimal parameters.

³In statistics, the maximum likelihood estimation method is used to estimate parameters in a model. Firstly, it is necessary to find the likelihood function $L(\theta; x)$, where the θ is a parameter for the model and x are the data points. Then we should find a θ maximizing the likelihood function.

To apply the MLE method, firstly, we should find an initial estimate of the parameters based on the empirical data. The empirical log-returns $R_{\Delta t}$ of the Google data are recalled in the Figure 8.

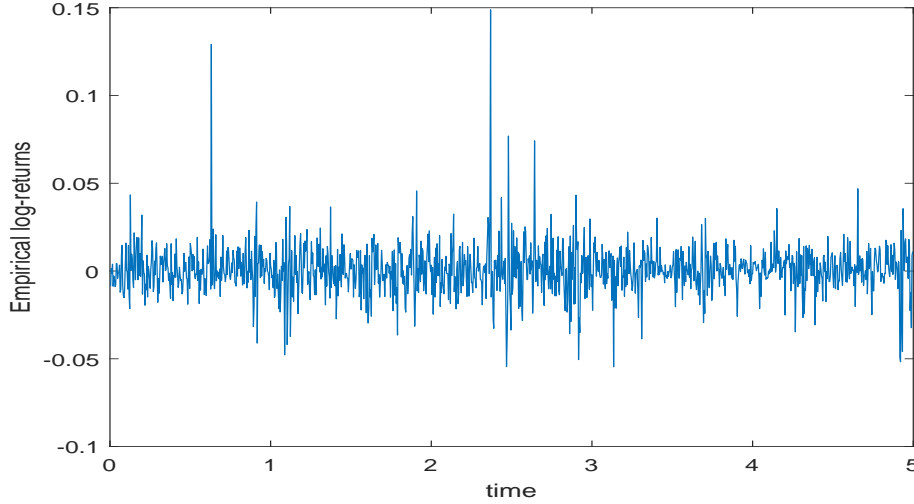


Figure 8: The log-return of the empirical stock prices.

A delicate matter is to decide when there is a jump. Here the decision rule is that a jump occurs if the absolute value of the log-return is larger than some positive value ε .

Here, the time is measured in years. Hence the parameter λ is estimated as

$$\begin{aligned}\hat{\lambda} &= \text{the number of jumps per year} \\ &= \frac{\text{the total number of jumps}}{\text{the total length in years}}.\end{aligned}\tag{11}$$

For the value ε , dividing the empirical log-return data into two groups \mathcal{D} and \mathcal{J} , the group \mathcal{D} includes log-returns with absolute value of log-returns less than value ε . For such log-returns, we decide that there is no jump. Oppositely, the group \mathcal{J} includes log-returns with absolute value of log-returns larger than value ε . For such log-returns, we decide that jumps have occurred. To estimate the initial parameters μ_d , σ_d , μ_j and σ_j , for simplicity, we decide that there is exactly only one jump for a log-return that belongs to group \mathcal{J} .

When there is exactly only one jump, the expectation and variance of the log-return are

$$\begin{aligned}
E(R_{\Delta t}^J) &= E[R_{\Delta t}|N_{t+\Delta t} - N_t = 1] \\
&= E[(\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d\Delta W + Q_i] \\
&= (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \mu_j
\end{aligned}$$

and

$$\begin{aligned}
\text{Var}(R_{\Delta t}^J) &= \text{Var}[R_{\Delta t}|N_{t+\Delta t} - N_t = 1] \\
&= \text{Var}[(\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d\Delta W + Q_i] \\
&= \sigma_d^2\Delta t + \sigma_j^2,
\end{aligned}$$

respectively. In summary,

$$\begin{cases} E(R_{\Delta t}^J) = (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \mu_j \\ \text{Var}(R_{\Delta t}^J) = \sigma_d^2\Delta t + \sigma_j^2. \end{cases} \quad (12)$$

The parameters μ_j and σ_j are estimated from equation (12),

$$\begin{cases} \hat{\mu}_j = \widehat{E}(R_{\Delta t}^J) - (\hat{\mu}_d - \frac{\hat{\sigma}_d^2}{2})\Delta t \\ \hat{\sigma}_j^2 = \widehat{\text{Var}}(R_{\Delta t}^J) - \hat{\sigma}_d^2\Delta t, \end{cases} \quad (13)$$

where $\widehat{E}(R_{\Delta t}^J)$ and $\widehat{\text{Var}}(R_{\Delta t}^J)$ are the sample mean and the sample variance of the empirical log-returns in group \mathcal{J} .

When there are no jumps, the expectation and variance of the log-return are

$$\begin{aligned}
E(R_{\Delta t}^D) &= E[R_{\Delta t}|N_{t+\Delta t} - N_t = 0] \\
&= E[(\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d\Delta W] \\
&= (\mu_d - \frac{\sigma_d^2}{2})\Delta t
\end{aligned}$$

and

$$\begin{aligned}
\text{Var}(R_{\Delta t}^D) &= \text{Var}[R_{\Delta t}|N_{t+\Delta t} - N_t = 0] \\
&= \text{Var}[(\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d\Delta W] \\
&= \sigma_d^2\Delta t,
\end{aligned}$$

respectively.

The parameters μ_d and σ_d can be estimated from the above formulas of $E(R_{\Delta t}^D)$ and $\text{Var}(R_{\Delta t}^D)$, similarly as for the BS model,

$$\begin{cases} \hat{\mu}_d = \frac{2\widehat{E}(R_{\Delta t}^D) + \widehat{\text{Var}}(R_{\Delta t}^D)\Delta t}{2\Delta t} \\ \hat{\sigma}_d^2 = \frac{\widehat{\text{Var}}(R_{\Delta t}^D)}{\Delta t}, \end{cases} \quad (14)$$

where $\widehat{E}(R_{\Delta t}^D)$ and $\widehat{\text{Var}}(R_{\Delta t}^D)$ are the sample mean and the sample variance of the empirical log-returns in group \mathcal{D} .

By reading off Figure 8 and from that figure, choosing $\varepsilon = 0.02$, we obtain the initial estimator of the parameters $\hat{\mu}_d = 0.23028$, $\hat{\sigma}_d = 0.13874$, $\hat{\lambda} = 25.0199$, $\hat{\mu}_j = -0.0011178$ and $\hat{\sigma}_j = 0.03452$ by (13) and (14). Here a MATLAB code is included with a hopefully clear description of the initial parameter estimate.

```
S=csvread('GOOG.csv',1,5,[1,5,1260,5]); % the emprical
    stock price
dt=1/252;
R=diff(log(S),1);
epsilon=0.02; %values to verify jump
jumpindex=find(abs(R)>epsilon);%if true then considered
    as jump
lambdahat=length(jumpindex)/((length(S)-1)*dt); % jump
    intensity, the number of jumps in per-year
Rjumps=R(jumpindex); % the data of 'jumpindex'
diffusionindex=find(abs(R)<=epsilon); % without jumps,
    the diffusion data can be consider as in BS model
Rdiffusion=R(diffusionindex); % the data of '
    diffusionindex'
sigmahat=std(Rdiffusion)/sqrt(dt);
muhat=[2*mean(Rdiffusion)+(sigmahat^2)*dt]/(2*dt);
sigma_jhat=sqrt((var(Rjumps)-sigmahat^2*dt));
mu_jhat=mean(Rjumps)-(muhat-sigmahat^2/2)*dt;
```

7.3.2 Maximum likelihood estimation

From previous subchapter, we have initial estimates of μ_d , σ_j , λ , μ_j , σ_j . Those initial estimates are used to numerically optimize the MJD likelihood. That likelihood is introduced here.

By Definition 6 and (9), the PDF of the MJD-modeled log-return $R_{\Delta t}$ is:

$$f_{R_{\Delta t}}(x) = \sum_{k=0}^{\infty} p_k(\lambda \Delta t) \varphi\left(x \middle| \left(\mu_d - \frac{\sigma_d^2}{2}\right) \Delta t + \mu_j k, \sigma_d^2 \Delta t + \sigma_j^2 k\right), \quad (15)$$

where $p_k(\lambda \Delta t) = p(\Delta N_t = k) = \frac{(\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}$. For a complete proof of the probability density function (15), see reference [2].

The objective function of the MLE method to maximize is the likelihood function

$$L(\theta; x) = \prod_{k=0}^n f_{R_{\Delta t}}(x_i),$$

where the function $f_{R_{\Delta t}}(t)$ is specified in equation (15), and $x = (x_1, \dots, x_n)$ is the empirical log-return data.

It is more convenient to minimize the minus log-likelihood function:

$$-\ln L(\theta; x) = -\sum_{k=0}^n \ln f_{R_{\Delta t}}(x_i). \quad (16)$$

By using the *fminsearch* code for minimizing $-\ln L$ and importing the initial estimates from previous subchapter, we can estimate the five parameters for different values of ε . The following table shows a relationship between the value ε and the MLE estimated parameters.

Table 1: Estimated parameters of the MLE method depending on different values of ε .

Parameter	$\varepsilon=0.07$		$\varepsilon=0.05$		$\varepsilon=0.02$	
	Initial value	Estimated value	Initial value	Estimated value	Initial value	Estimated value
μ_d	0.12661	0.22343	0.16848	0.22343	0.23028	0.22343
σ_d	0.19752	0.15442	0.19193	0.15442	0.13874	0.15442
λ	0.800635	33.9377	1.60127	33.9377	25.0199	33.9377
μ_j	0.10689	-0.00055442	0.026621	-0.00055442	-0.0011178	-0.00055441
σ_j	0.035407	0.025513	0.088274	0.025513	0.03452	0.025513
$-\ln L$	-3.7457e+03		-3.7457e+03		-3.7457e+03	

From Table 1, even though we have chosen different values of ε , the estimated maximum likelihood parameters are the same, $\hat{\mu}_d = 0.22343$, $\hat{\sigma}_d = 0.15442$, $\hat{\lambda} = 33.9377$, $\hat{\mu}_j = -0.00055442$ and $\hat{\sigma}_j = 0.025513$. This indicates that the MLE method is not strongly depending on the value of ε . However, if the value ε is larger than the maximum absolute value of empirical log-returns $R_{\Delta t}$, it is decided that there are no jumps. Then the parameter λ is estimated to be 0, and μ_j and σ_j can not be estimated.

7.4 Comparison of the empirical log-returns with the MJD-modeled log-returns

From the estimated parameters $\hat{\mu}_d = 0.22343$, $\hat{\sigma}_d = 0.15442$, $\hat{\lambda} = 33.9377$, $\hat{\mu}_j = -0.00055442$ and $\hat{\sigma}_j = 0.025513$, we can draw an approximation of the PDF (15), illustrated in the Figure 9. Here the number of terms in (15) is 100. For the estimated parameters, the number of terms seems to be sufficient.

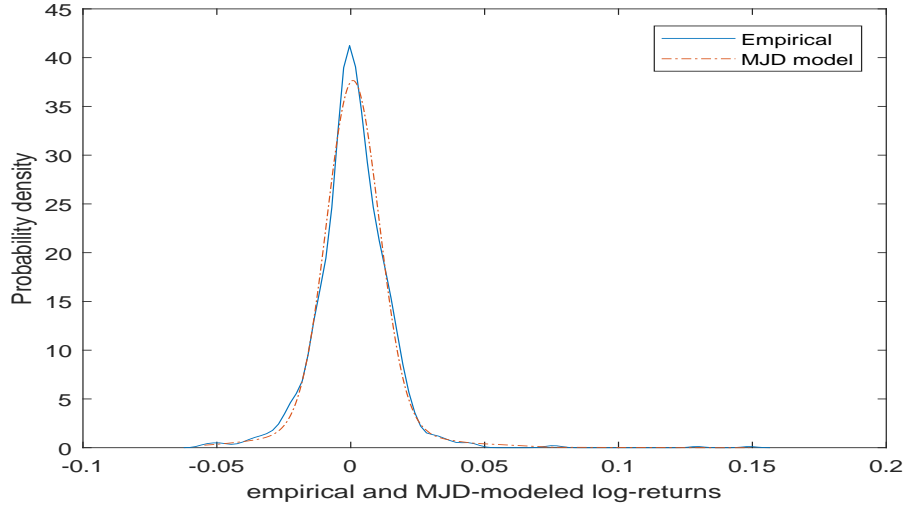


Figure 9: The PDF of MJD-modeled log-returns and the kernel density estimation of empirical log-returns.

From Figure 9, we find that the density function of the MJD-modeled log-returns is quite close to the kernel density estimation of empirical log-returns. This is also indicated in Figure 10, where the empirical log-returns are compared with the MJD-modeled log-returns.

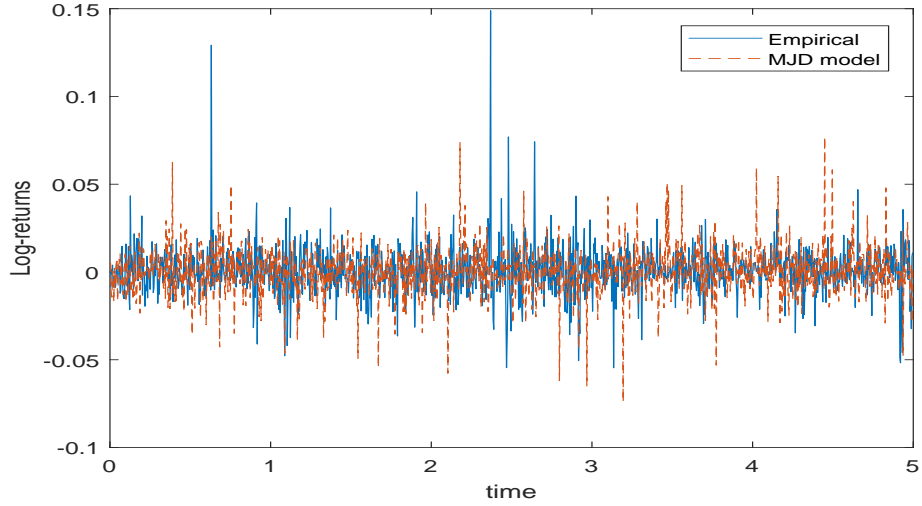


Figure 10: Comparison of the empirical log-returns with the MJD-modeled log-returns.

We recall the comparison of empirical log-returns with the BS-modeled log-returns in Figure 3, which indicated that some occasional larger spikes. From Figure 10, there are also some larger spikes for the MJD-modeled log-returns, indicating that the MJD model can pay attention to the occasional larger spikes. Therefore, the MJD model may be more suitable for the real stock price.

8 Comparison of the BS Model with the MJD Model for Empirical stock prices

Since the BS model has normal distributed log-returns and our empirical log-returns are far from being normal distributed, the BS model is not a very suitable stock price model. However, the MJD model seems to be closer to the empirical stock price, at least for the Google data considered in this thesis.

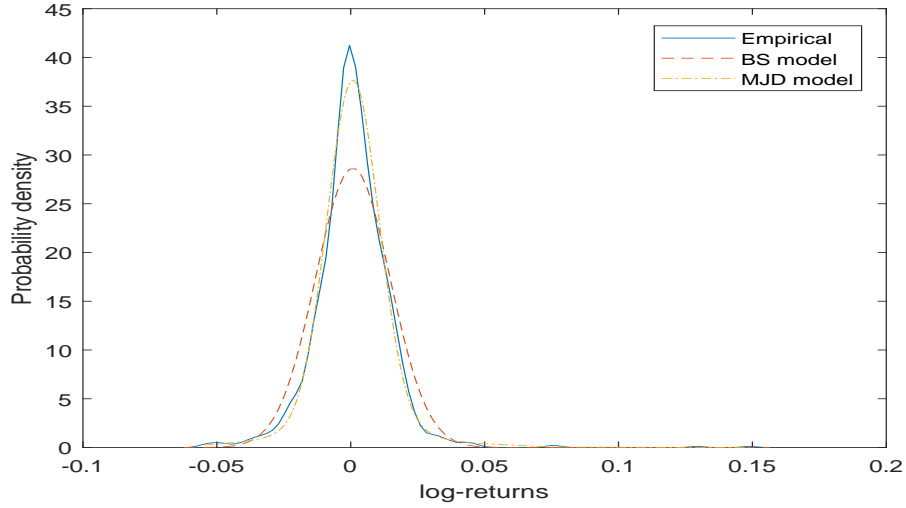


Figure 11: The PDF of two modeled log-returns and the kernel density estimation of empirical log-returns.

From Figure 11, the PDF of the MJD model is quite closer to reality than the BS model.

Now we compare the BS model and the MJD model with the real data by investigating their moments.

We recall that the log-returns of the BS model is

$$\begin{aligned} R_{\Delta t} &= \ln\left(\frac{S_{t+\Delta t}}{S_t}\right) \\ &= \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\Delta W_t, \end{aligned}$$

where ΔW_t is a standard Brownian motion increment, $\Delta W_t \sim \sqrt{\Delta t}N(0, 1)$. The mean, variance, skewness and kurtosis for the BS-model log-returns are

shown below:

$$\begin{aligned}
E(R_{\Delta t}) &= (\mu - \frac{\sigma^2}{2})\Delta t, \\
\text{Var}(R_{\Delta t}) &= \sigma^2 \Delta t, \\
\text{Ske}(R_{\Delta t}) &= \frac{E[(R_{\Delta t} - E(R_{\Delta t}))^3]}{(\text{Var}(R_{\Delta t}))^{\frac{3}{2}}} \\
&= \frac{E[\sigma \Delta W_t]^3}{\sigma^2 (\Delta t)^{\frac{3}{2}}} \\
&= \frac{E[\sigma \sqrt{\Delta t} N(0, 1)]^3}{\sigma^3 \Delta t^{\frac{3}{2}}} \\
&= \frac{\sigma^3 (\Delta t)^{\frac{3}{2}} E(N(0, 1))^3}{\sigma^3 (\Delta t)^{\frac{3}{2}}} \\
&= E(N(0, 1)^3), \\
\text{Kur}(R_{\Delta t}) &= \frac{E[(R_{\Delta t} - E(R_{\Delta t}))^4]}{(\text{Var}(R_{\Delta t}))^2} \\
&= \frac{E[\sigma \sqrt{\Delta t} N(0, 1)]^4}{\sigma^4 (\Delta t)^2} \\
&= \frac{\sigma^4 (\Delta t)^2 E(N(0, 1))^4}{\sigma^4 \Delta t^2} \\
&= E(N(0, 1)^4),
\end{aligned}$$

where the PDF of the standard normal distribution $N(0, 1)$ is $\varphi(x|0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$.

Thus, $E(N(0, 1)^3) = \int_{-\infty}^{\infty} x^3 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 0$, $E(N(0, 1)^4) = \int_{-\infty}^{\infty} x^4 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 3$. Therefore, $\text{Ske}(R_{\Delta t}^{bs}) = 0$ and $\text{Kur}(R_{\Delta t}^{bs}) = 3$.

From $\hat{\mu} = 0.2171$, $\hat{\sigma} = 0.2210$ and $\Delta t = \frac{1}{252}$, $\hat{E}(R_{\Delta t}) = 0.0008$, and $\widehat{\text{Var}}(R_{\Delta t}) = 0.0002$. Obviously, the above computed values of $\hat{E}(R_{\Delta t})$ and $\widehat{\text{Var}}(R_{\Delta t})$ are equal to the sample mean and sample variance of the empirical data.

We also recall that the log-return of the MJD model is:

$$R_{\Delta t} = (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \sigma_d \Delta W_t + \sum_{i=N_t}^{N_t+\Delta t} Q_i,$$

where ΔW_t is a standard Brownian motion increment, $\Delta W_t \sim \sqrt{\Delta t} N(0, 1)$, and Q_i are independent normal random variables, $Q_i \sim N(\mu_j, \sigma_j^2)$. We recall that $\sum_{i=N_t}^{N_t+\Delta t} Q_i$ has the same distribution as $\sum_{i=1}^{\Delta N_t} Q_i$, where ΔN_t is a Poisson

process increment with mean $\lambda\Delta t$. Then $E(\Delta N_t) = \text{Var}(\Delta N_t) = \lambda\Delta t$. The moments of the MJD model are presented as follows.

$$\begin{aligned} E(R_{\Delta t}) &= (\mu_d - \frac{\sigma_d^2}{2})\Delta t + E(Q)E(\Delta N_t) \\ &= (\mu_d - \frac{\sigma_d^2}{2})\Delta t + \mu_j\lambda\Delta t, \end{aligned} \quad (17)$$

$$\begin{aligned} \text{Var}(R_{\Delta t}) &= E[(R_{\Delta t})^2] - [E(R_{\Delta t})]^2 \\ &= E[(\sigma_d\Delta W_T + Q\Delta N_t - \mu_j\lambda\Delta t)^2] \\ &= E[(\sigma_d\Delta W_T)^2 + (Q\Delta N_t)^2 + (\mu_j\lambda\Delta t)^2 + 2\sigma_d\Delta W_T Q\Delta N_t \\ &\quad - 2\sigma_d\Delta W_T \mu_j\lambda\Delta t - 2Q\Delta N_t \mu_j\lambda\Delta t] \\ &= \sigma_d^2\Delta t + (\sigma_j^2 + \mu_j^2)\lambda\Delta t, \end{aligned} \quad (18)$$

$$\begin{aligned} \text{Ske}(R_{\Delta t}) &= \frac{E[(R_{\Delta t} - E(R_{\Delta t}))^3]}{(\text{Var}(R_{\Delta t}))^{\frac{3}{2}}} \\ &= \frac{(3\sigma_j^2 + \mu_j^2)\mu_j\lambda\Delta t}{[\sigma_d^2\Delta t + (\sigma_j^2 + \mu_j^2)\lambda\Delta t]^{\frac{3}{2}}}, \end{aligned} \quad (19)$$

$$\begin{aligned} \text{Kur}(R_{\Delta t}) &= \frac{E[(R_{\Delta t} - E(R_{\Delta t}))^4]}{(\text{Var}(R_{\Delta t}))^2} \\ &= \frac{(3\sigma_j^4 + 6\mu_j^2\sigma_j^2 + \mu_j^4)\lambda\Delta t + (3\lambda^2(\sigma_j^2 + \mu_j^2)^2 + 6\lambda\sigma_d^2(\sigma_j^2 + \mu_j^2) + 3\sigma_d^4)(\Delta t)^2}{[\sigma_d^2\Delta t + (\sigma_j^2 + \mu_j^2)\lambda\Delta t]^2}. \end{aligned} \quad (20)$$

For the complete proof of the above moments, see Chapter 2 of the reference [2].

From Chapter 6, the estimated parameters for the MJD model are $\hat{\mu}_d = 0.22343$, $\hat{\sigma}_d = 0.15442$, $\hat{\lambda} = 33.9377$, $\hat{\mu}_j = -0.00055441$ and $\hat{\sigma}_j = 0.025513$. Then we can estimate the moments by inserting the estimated parameters into (17), (18), (19) and (20): $\widehat{E}(R_{\Delta t}) = 0.0008$, $\widehat{\text{Var}}(R_{\Delta t}) = 0.0002$, $\widehat{\text{Ske}}(R_{\Delta t}) = -0.0592$, and $\widehat{\text{Kur}}(R_{\Delta t}) = 8.1541$.

Table 2: Comparison of the empirical log-returns with the two modeled log-returns.

	Mean	Variance	Skewness coefficient	Kurtosis coefficient
Reality	0.0008	0.0002	1.5215	20.5341
BS model	0.0008	0.0002	0	3
MJD model	0.0008	0.0002	-0.0592	8.1541

From Table 2, the estimated means and variances of the log-returns for the two models coincide with the empirical log-returns. The high positive

skewness for the empirical log-returns can be explained by the two relatively large positive values illustrated in Figure 8. The skewness coefficients for the two models are not very close to the real data, indicating that the two models do not capture the positive skewness of the empirical log-returns. The kurtosis coefficients for the empirical data is larger than for the BS model and the MJD model. However, the kurtosis coefficient for the MJD model seems to be clearly closer to the empirical kurtosis than for the BS model.

From Figure 3 and 8, we find that there are several larger spikes in empirical log-returns, and the MJD model is a way to take care of these spikes. Besides, the MJD model indicates that most of the log-returns are small in size as for the empirical data. That can also be read off from the PDF of the MJD model.

Hence, we can conclude that the MJD model is substantially more suitable for the real stock market than the BS model at least for our empirical Google data.

In this thesis, we only compare two models and get the table 2 under the empirical Google data. Other empirical data have also been investigated and similar results have been obtained.

9 Conclusion

To sum up, we have discussed the Black-Scholes model and the Merton Jump-Diffusion model. We try to find whether the MJD model is significantly more suitable for empirical stock prices. From the probability density functions of the two models, we find that the probability density function of MJD model is clearly closer to the kernel density estimation for the Google empirical data. To verify this conclusion, we compare the moments of the BS model and the MJD model with empirical moments. We get that the kurtosis coefficient of the MJD model is closer to the empirical kurtosis than for the BS model. Based on this evidence, it is reasonable to say that the MJD model is better to predict the future distribution of stock prices.

For initial guess of the parameters for the MJD model, the author did not find a reference of an analytic method. Therefore, a jump decision rule was introduced based on a threshold ε . The decision rule seems to give robust MLE estimates of optimal parameters.

10 Reference

References

- [1] Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy*, 81(3), 637-654.
- [2] Hanson, F. B., & Westman, J. J. Stochastic Analysis of Jump-Diffusions for Financial Log-Return Processes (corrected).
- [3] Hanson, F. B., & Zhu, Z. (2004, December). Comparison of market parameters for jump-diffusion distributions using multinomial maximum likelihood estimation. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on* (Vol. 4, pp. 3919-3924). IEEE.
- [4] HÅKANSSON, J., & ALBIN, P. (2015). Option Pricing and Exponential Levy Models.
- [5] Lawler, G. F. (2010). *Stochastic Calculus: An Introduction with Applications*. American Mathematical Society.
- [6] Moore, D. S., McCabe, G. P., & Evans, M. J. (2005). *Introduction to the practice of statistics Minitab manual and Minitab version 14*. WH Freeman & Co.
- [7] Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, 3(1-2), 125-144.
- [8] Synowiec, D. (2008). Jump-diffusion models with constant parameters for financial log-return processes. *Computers & Mathematics with Applications*, 56(8), 2120-2127.
- [9] Tankov, P. (2003). *Financial modelling with jump processes* (Vol. 2). CRC press.
- [10] 2013/03/01 ~ 2018/03/05 Yahoo finance <https://finance.yahoo.com/quote/GOOG/history/>

Appendices

A Matlab codes for the figures

```

% Simulation of Brownian motion and BS model(Figure
1&2)
Ns=5; %the number of simulation
dt=1/252;
t=linspace(0,(2000-1)*dt,2000)';
W=cumsum([zeros(1,Ns);sqrt(dt).*normrnd(0,1,length(t)
-1,Ns)]);
% standard brownian motion
mu=0.15;sigma=0.19;S0=416;
Ssim=S0*exp((mu-sigma^2/2)*t*ones(1,Ns)+sigma*W);
% simulation of stock price
subplot(1,2,1)
plot(t,W)
title('Standard_Brownian_motion')
xlabel('time')
ylabel('Brownian_motion')
subplot(1,2,2)
plot(t,Ssim)
title('Simulation_Stock_price')
xlabel('time')
ylabel('Stock_prices')

```

```

%Code for Figure 3&4
Ns=1; %number of simulation
dt=1/252;
t=linspace(0,(1260-1)*dt,1260)';
W=cumsum([zeros(1,Ns);sqrt(dt).*normrnd(0,1,length(t)
-1,Ns)]);
mu=0.2171;sigma=0.2210;S0=416;% estimate the parameter
mu and sigma
Ssim=S0*exp((mu-sigma^2/2)*t*ones(1,Ns)+sigma*W);
%stock price of simulation
Rsim=diff(log(Ssim),1); % return of the simulated stock
price
S=csvread('GOOG.csv',1,5,[1,5,1260,5]); % emprical
stock data
R=diff(log(S),1); % return of the real stock price
figure(1)
x=t(2:end);
plot(x,R,x,Rsim,':')

```

```

xlabel('time')
ylabel('Log-returns')
legend('Empirical','BS_model')
figure(2)
subplot(2,1,1)
[fks,xks]=ksdensity(R); % probability density estimate
plot(xks,fks,xks,normpdf(xks,(mu-sigma^2/2)*dt,sigma*
    sqrt(dt))),axis tight
xlabel('difference_of_logarithmic_return')
ylabel('Probability_density')
legend('Empirical','BS_model')
% Jarque-Bera test and Kolmogorov-Smirnov test
[h,p]=jbtest(R)% If the result is 1, the test rejects
    the normal hypothesis
[h,p] = kstest(R) % If the result is 1, the test
    rejects the standard normal hypothesis

```

```

% the code of simulated Poisson process
Ns=1;% the number of simulation
T=1; % the time period
lambda=15; % the jump intensity
ngrid=1000;
t=linspace(0,T,ngrid)';
dt=t(2)-t(1); % the per-time
N=cumsum([zeros(1,Ns);poissrnd(lambda*dt,length(t)-1,Ns
    )]); % poisson distribution
plot(t,N)
xlabel('time')
ylabel('Poisson_process_N_t')

```

```

% the code of simulated compound Poisson process
Ns=1;% the number of simulation
T=1; % the time period
lambda=15; % the jump intensity
m=0.5; %the mean of jump size
delta=1;% the standard deviation of jump size
ngrid=1000;
t=linspace(0,T,ngrid)';
dt=t(2)-t(1); % the per-time
dN=poissrnd(lambda*dt,ngrid-1,Ns); % the poisson
    distribution

```

```

sumD=m*dN+delta*sqrt(dN).*randn(ngrid-1,Ns);% the jump
    size is normal distributed
Y_t=cumsum([zeros(1,Ns);sumD]);
plot(t,Y_t)
xlabel('time')
ylabel('Compound_Poisson_process_Y_t')

```

```

%the code of figure 7
Ns=5;% the number of simulation
dt=1/252;%the per-time in one year
t=linspace(0,(2000-1)*dt,2000)';
mu=0.17;sigma=0.055;lambda=10;muj=0.005;sigmaj=0.08; S0
=416;
Rsim=logmertonrnd(dt,mu,sigma,lambda,muj,sigmaj,t,Ns);
Ssim=S0*exp(cumsum([zeros(1,size(Rsim,2));Rsim]))
plot(t,Ssim)
    xlabel('time')
    ylabel('Simulated_stock_prices')
function Rsim=logmertonrnd(dt,mu,sigma,lambda,muj,
    sigmaj,t,Ns)%define simulated return function
dN=poissrnd(lambda*dt,length(t)-1,Ns);% number of jump
    between grid point
Y=muj*dN+sigmaj*sqrt(dN).*randn(length(t)-1,Ns);% sum
    of the normal jumps between grid points
dW=sqrt(dt).*normrnd(0,1,length(t)-1,Ns);% standard
    brownian motion
Rsim=(mu-sigma^2/2)*dt+sigma*dW+Y;
end

```

```

% the code of figure 8
S=csvread('GOOG.csv',1,5,[1,5,1260,5]);% empirical
    stock price
dt=1/252;%per-time in a year
t=linspace(0,(length(S)-1)*dt,length(S))';
R=diff(log(S),1); % the return of logarithmic stock
    price
x=t(2:end); % time period of return
plot(x,R)
xlabel('time')
ylabel('Empirical_log-return')

```

```

%Code of figure 9 &10&11
S=csvread( 'GOOG.csv' ,1,5,[1,5,1260,5]);
dt=1/252;
t=linspace(0,(length(S)-1)*dt,length(S));
R=diff(log(S),1); % the empirical log-return
L=0.02; %values to verify jump
jumpindex=find(abs(R)>L);%if true then considered as
jump
lambdahat=length(jumpindex)/((length(S)-1)*dt); % jump
intensity, the number of jumps in per-year
Rjumps=R(jumpindex); % the data of 'jumpindex'
diffusionindex=find(abs(R)<=L); % without jumps, the
diffusion data can be consider as in BS model
Rdiffusion=R(diffusionindex); % the data of '
diffusionindex'
sigmahat=std(Rdiffusion)/sqrt(dt);
muhat=[2*mean(Rdiffusion)+(sigmahat^2)*dt]/(2*dt);
sigma_jhat=sqrt((var(Rjumps)-sigmahat^2*dt));
mu_jhat=mean(Rjumps)-(muhat-sigmahat^2/2)*dt;
theta0=[muhat sigmahat lambdahat mu_jhat sigma_jhat];
theta=fminsearch(@(theta)Logmerton(theta(1),theta(2),
theta(3),theta(4),theta(5)),theta0);
figure 1
[fks,xks]=ksdensity(R);
plot(xks,fks,sort(R),logmertonpdf(sort(R),dt,theta(1),
theta(2),theta(3),theta(4),theta(5)),'-')
xlabel('empirical_and_MJD-modeled_log-returns')
ylabel('Probability_density')
legend('Empirical','MJD_model')
figure 2
Rsim=logmertonrnd(dt,theta(1),theta(2),theta(3),theta
(4),theta(5),t,1);
x=t(2:end);
plot(x,R,x,Rsim,'—')
xlabel('time')
ylabel('Log-returns')
legend('Empirical','MJD_model')
figure 3
mubs=0.2171;sigmabs=0.2210; % mean and variance of BS
model

```



```

[fks , xks]=ksdensity(R); % probability density estimate
plot(xks , fks , xks , normpdf(xks , (mubs-sigmabs ^2/2)*dt ,
    sigmabs*sqrt(dt)) , '—' , sort(R) , logmertonpdf(sort(R)
    , dt , theta(1) , theta(2) , theta(3) , theta(4) , theta(5)) , '
    :')
xlabel('log-returns')
ylabel('Probability density')
legend('Empirical' , 'BS_model' , 'MJD_model')
% build a density function of MJD model
function pdflog=logmertonpdf(r , dt , mu , sigma , lambda , mu_j ,
    sigma_j)
    if lambda>0
        nterm=100;
        term=zeros(length(r) , nterm); % return a matrix
            length(r)*100
        for k=1:nterm
            poisson=(lambda*dt)^k/prod(1:k)*exp(-lambda*dt)
                ; % poisson distribution
            normal=1/sqrt(2*pi*(sigma^2*dt+sigma_j^2*k))*
                exp(-(r-((mu-sigma^2/2)*dt+mu_j*k)).^2/(2*(
                sigma^2*dt+sigma_j^2*k))); % normal
                distribution
            term(:,k) =poisson*normal;
        end
        pdflog=sum([1/sqrt(2*pi*sigma^2*dt)*exp(-(r-(mu-
            sigma^2/2)*dt).^2/(2*sigma^2*dt))*exp(-lambda*
            dt) term] , 2);
    else % lambda <=0
        pdflog=1/sqrt(2*pi*sigma^2)*exp(-(r-(mu-sigma
            ^2/2)*dt).^2/(2*(sigma^2*dt)));
    end
end
function Rsim=logmertonrnd(dt , mu , sigma , lambda , mu_j ,
    sigma_j , t , Ns)
dN=poissrnd(lambda*dt , length(t)-1 , Ns); % number of jump
    between grid point
Y=mu_j*dN+sigma_j*sqrt(dN).*randn(length(t)-1 , Ns); % sum
    of the normal jumps between grid points
dW=sqrt(dt).*normrnd(0 , 1 , length(t)-1 , Ns); % brownian
    motion
Rsim=(mu-sigma^2/2)*dt+sigma*dW+Y;

```

```
end
```

B Matlab codes for parameter estimations

```
%Estimate the parameter mu and sigma of BS model
Ns=1; %number of simulation
S=csvread('GOOG.csv',1,5,[1,5,1260,5]);
dt=1/252;
t=linspace(0,(2000-1)*dt,2000)';
W=cumsum([zeros(1,Ns);sqrt(dt).*normrnd(0,1,length(t)
-1,Ns)]);% brownian motion
mu=0.15;sigma=0.19;S0=416;
Ssim=S0*exp((mu-sigma^2/2)*t*ones(1,Ns)+sigma*W); %
Simulated stock price of BS model
Test=diff(log(Ssim),1);
mutest=mean(Test)/dt+var(Test)/(2*dt);
sigmatest=sqrt(var(Test)/dt);
[mu mutest; sigma sigmatest];
S=csvread('GOOG.csv',1,5,[1,5,1260,5]);
R=diff(log(S),1);% the return of the emprical stock
price
muhat=mean(R)/dt+var(R)/(2*dt);
sigmahat=sqrt(var(R)/dt);
[mu muhat; sigma sigmahat];
```

```
% estimation code of MJD model
S=csvread('GOOG.csv',1,5,[1,5,1260,5]); % the emprical
stock price
dt=1/252;
R=diff(log(S),1);
epsilon=0.02; %values to verify jump
jumpindex=find(abs(R)>epsilon);%if true then considered
as jump
lambdahat=length(jumpindex)/((length(S)-1)*dt); % jump
intensity, the number of jumps in per-year
Rjumps=R(jumpindex); % the data of 'jumpindex'
diffusionindex=find(abs(R)<=epsilon); % without jumps,
the diffusion data can be consider as in BS model
```

```

Rdiffusion=R(diffusionindex); % the data of
diffusionindex
sigmahat=std(Rdiffusion)/sqrt(dt);
muhat=[2*mean(Rdiffusion)+(sigmahat^2)*dt]/(2*dt);
sigma_jhat=sqrt((var(Rjumps)-sigmahat^2*dt));
mu_jhat=mean(Rjumps)-(muhat-sigmahat^2/2)*dt;
theta0=[muhat sigmahat lambdahat mu_jhat sigma_jhat]; %
initial value
Logmerton=@(mu,sigma,lambda,mu_j,sigma_j)-sum(log(
logmertonpdf(R,dt,mu,sigma,lambda,mu_j,sigma_j)));
[theta fval]=fminsearch(@(theta)Logmerton(theta(1),
theta(2),theta(3),theta(4),theta(5)),theta0) %
Maximum likelihood method from initial value
Rsim=logmertonrnd(dt,theta(1),theta(2),theta(3),theta
(4),theta(5),t,1); %simulated return
Logmertons=@(mu,sigma,lambda,mu_j,sigma_j)-sum(log(
logmertonpdf(Rsim,dt,mu,sigma,lambda,mu_j,sigma_j)))
;
thetas=fminsearch(@(theta)Logmertons(theta(1),theta(2),
theta(3),theta(4),theta(5)),theta);%maximum
likelihood method from estimated value 'theta'
disp(['mu_ ', num2str([theta0(1) theta(1) thetas(1)])])
disp(['sigma_ ', num2str([theta0(2) theta(2) thetas(2)])
])
disp(['lambda_ ', num2str([theta0(3) theta(3) thetas(3)])
])
disp(['mu_j_ ', num2str([theta0(4) theta(4) thetas(4)])])
disp(['sigma_j_ ', num2str([theta0(5) theta(5) thetas(5)])
])
%moment estimation of empirical log-return
M1sp=mean(R);
M2sp=sum((R-M1sp).^2)/(length(R)-1);%the second central
moment
M3sp=sum((R-M2sp).^3)/(length(R)-1); % the third
central moment
M4sp=sum((R-M3sp).^4)/(length(R)-1); % the fourth
central moment
Beta3sp=M3sp/(M2sp^(1.5));%Skewness coefficient
Beta4sp=M4sp/(M2sp^(2));%Kurtosis coefficient
%moment estimation of estimated parameters by formulas
M1=(theta(1)-theta(2)^2/2+theta(3)*theta(4))*dt;

```

```

M2=(theta(2)^2+theta(3)*(theta(5)^2+theta(4)^2))*dt;
M3=(3*theta(5)^2+theta(4)^2)*theta(4)*theta(3)*dt;
M4=(3*theta(5)^4+6*theta(4)^2*theta(5)^2+theta(4)^4)*
    theta(3)*dt+(3*theta(3)^(2)*(theta(5)^2+theta(4)^2)
    ^2)+6*theta(3)*theta(2)^(2)*(theta(5)^2+theta(4)^2)
    +3*theta(2)^(4))*(dt^2);
Beta3=M3/(M2^(1.5));%Skewness coefficient
Beta4=M4/(M2^(2));%Kurtosis coefficient

function pdflog=logmertonpdf(r,dt,mu,sigma,lambda,mu_j,
    sigma_j) % build a density function of MJD model
if lambda>0
    nterm=100;
    term=zeros(length(r),nterm);% return a matrix
        length(r)*100
    for k=1:nterm
        poisson=(lambda*dt)^k/prod(1:k)*exp(-lambda*dt)
            ;
        normal=1/sqrt(2*pi*(sigma^2*dt+sigma_j^2*k))*
            exp(-(r-((mu-sigma^2/2)*dt+mu_j*k)).^2/(2*(
            sigma^2*dt+sigma_j^2*k)));
        term(:,k) =poisson*normal;
    end
    pdflog=sum([1/sqrt(2*pi*sigma^2*dt)*exp(-(r-(mu-
        sigma^2/2)*dt).^2/(2*sigma^2*dt))*exp(-lambda*
        dt) term],2);
else% lambda <=0
    pdflog=1/sqrt(2*pi*sigma^2)*exp(-(r-(mu-sigma
        ^2/2)*dt).^2/(2*(sigma^2*dt)));
end
end
% build a function of simulated logarithmic return of
MJD model
function Rsim=logmertonrnd(dt,mu,sigma,lambda,mu_j,
    sigma_j,t,Ns)
dN=poissrnd(lambda*dt,length(t)-1,Ns);% number of jump
    between grid point
Y=mu_j*dN+sigma_j*sqrt(dN).*randn(length(t)-1,Ns);% sum
    of the normal jumps between grid points
dW=sqrt(dt).*normrnd(0,1,length(t)-1,Ns);% brownian
    motion

```

```
Rsim=(mu-sigma ^ 2/2)*dt+sigma*dW+Y;  
end
```



Linnæus University
Sweden

Faculty of Technology
SE-391 82 Kalmar | SE-351 95 Växjö
Phone +46 (0)772-28 80 00
teknik@lnu.se