Application of Markov Chain Monte Carlo Methods in Financial Modeling

Shenyi Pan

1 Introduction

In modern studies on quantitative finance, many stochastic models are represented in continuous-time form. In these models, the dynamic behavior of the underlying random factors are often specified by certain stochastic differential equations. Empirical investigations of these continuous-time models usually aims to extract information and do inference on the state variables and the parameters from the observed asset prices. That is, we are more interested in $p(\Theta, X|Y)$, which is the conditional distribution of the parameters Θ and the state variables X on the observed empirical prices Y. For example, consider Heston model for stochastic volatility:

$$\frac{dS_t}{S_t} = \mu S_t dt + \sqrt{V_t} dW_t^S,$$
$$dV_t = \kappa(\theta - V_t)dt + \xi\sqrt{V_t} dW_t^V,$$

where W_t^S and W_t^V are two Wiener processes with correlation ρ, μ is is the rate of return of the asset, θ is the long variance (or long run average price variance), κ is the rate at which V_t reverts to θ, and ξ is the volatility of the volatility. In this model, the goal of empirical study is to learn about the volatility state variables $V = \{V_t\}_{t=1}^T$, as well as the parameters $\Theta = \{\rho, \mu, \theta, \kappa, \xi\}$, from the observed prices $Y = \{S_t\}_{t=1}^T$. In other words, we are interested in the joint conditional distribution $p(\Theta, X|Y)$. With the joint distribution, we could further obtain the marginal distributions $p(\Theta|Y)$ and $p(X|Y)$, which allow us to estimate the parameters and state variables.

However, in practice, estimation of $p(\Theta, X|Y)$ in continuous-time models is usually difficult due to the following reasons. Firstly, the data we observe from the market are discrete while the models specify the asset prices continuously with respect to time. It is also possible that the model requires some latent state variables, which is difficult to identify and include in the analysis. Further, $p(\Theta, X|Y)$ could be of high dimension or involve certain non-normal and non-standard terms such as the case of jumps models. Lastly, in certain option pricing models, the parameter even do not have a closed-form solution, which makes it more difficult to do further analysis.

In face of these difficulties, this project will introduce an alternative way, namely using the Markov chain Monte Carlo (MCMC) method to estimate the continuous-time financial models.

2 Bayesian Inference on Financial Modeling

Similar with the setting in the previous section, we use Θ, X, and Y to represent the parameters, state variables, and observed asset prices respectively. Then by Bayes rule, we can decompose $p(\Theta, X|Y)$ into

$$p(\Theta, X|Y) \propto p(Y|X, \Theta)p(X, \Theta) \propto p(Y|X, \Theta)p(X|\Theta)p(\Theta),$$

where $p(Y|X, \Theta)$ is the conditional likelihood function, $p(X|\Theta)$ is the conditional distribution of the state variables on the specified parameters, and $p(\Theta)$ is the prior distribution of the parameters. Assume the time between two observations is 1 and also the Markov properties are satisfied, then we could further decompose $p(Y|X, \Theta)$ and $p(X|\Theta)$ into: $p(Y|X, \Theta) = \prod_{t=1}^{T} p(Y_t|Y_{t-1}, X_{t-1}, \Theta)$, and $p(X|\Theta = \prod_{t=1}^{T} p(X_t|X_{t-1}, \Theta)$.

In order to apply MCMC method to asset pricing models, we must first be able to evaluate the conditional distributions of interest, i.e., $p(Y_t|Y_{t-1}, X_{t-1}, \Theta)$ and $p(X_t|X_{t-1}, \Theta)$. In practice, these conditional distributions of observed prices and state variables are usually characterized by some stochastic differential equations.

2.1 Modeling Observed Pirces

For the observed asset price, its dynamics could either be modeled directly as the solution to a stochastic differential equation, which is commonly used in modeling equity prices or exchange rates, or be modeled as a deterministic function between state variables and parameters, which is more common in modeling options and term structure products.

In the first case, for example, the observed prices $Y = \{S_t\}_{t=1}^{T}$ could be characterized by the following stochastic differential equation:

$$S_{t+1} = S_t + \int_{s=t}^{t+1} \mu_S(S_s, X_s, \Theta)ds + \int_{s=t}^{t+1} \sigma(S_s, X_s, \Theta)dW_s + \sum_{j=N_t}^{N_t+1} \xi_j,$$

where dW_t is the increment from standard Wiener process, $\{N_t\}$ is a point process of jump times with stochastic density λ_t, ξ_j is the jump size. The distribution $p(Y_t|Y_{t-1}, X_{t-1}, \Theta)$ implied by the solution to the stochastic differential equation then generates the conditional likelihood function.

In the second case, the observed asset price is a known function of certain state variables and parameters, namely $S_t = f(X_t, \Theta)$. For example, let $r_t = r(X_t)$ be the stochastic interest rate process, then the price of a European call option is given by

$$f(X_t, \Theta) = E^Q[e^{-\int_{s=t}^{T} r(X_s)ds}(X_T - K)_+|X_t],$$

where K is the strike price of the call option, and Q is the risk-neutral measure.

2.2 Modeling State Variables

Similar with asset price, the state variables $X = \{X_t\}_{t=1}^{T}$ is also modeled in the form of stochastic differential equations. Two of the most common specifications for state variables are diffusion models as well as jump-diffusion models.

In diffusion models, the state variables are modeled by

$$X_{t+1} = X_t + \int_{s=t}^{t+1} \mu(X_s, \Theta) ds + \int_{s=t}^{t+1} \sigma(X_s, \Theta) dW_s$$

where dW_t is the increment from standard Wiener process.

Note that in diffusion models, the sample path is continuous with respect to time. However, it is possible to relax this continuity assumption by introducing a jump term in the stochastic differential equations. By doing this, we could have the jump-diffusion models for the state variables as follows:

$$X_{t+1} = X_t + \int_{s=t}^{t+1} \mu(X_s, \Theta) ds + \int_{s=t}^{t+1} \sigma(X_s, \Theta) dW_s + \sum_{j=N_t}^{N_t+1} \xi_j,$$

which is similar to the jump-diffusion model for asset prices.

2.3 Modeling Parameter Distribution

The third component of the decomposition of the posterior distribution $p(\Theta, X|Y)$ is the prior distribution of the parameters, namely $p(\Theta)$. It represents the non-sample information on the parameters. Usually we use conjugate prior distributions if possible, which provides a convenient way to find conditional posterior distributions that have closed forms and are easy to simulate in practice. For example, if our target distribution follows normal, then we will use another normal distribution as the conjugate prior.

2.4 Time Discretization Technique

Since the conditional distributions $p(Y_t|Y_{t-1}, X_{t-1}, \Theta)$ and $p(X_t|X_{t-1}, \Theta)$ are given as the solution of stochastic differential equations, it is difficult to simulate them due to the lack of a closed form solution. Therefore, in order to simulate the aforementioned continuous-time models, one of the most important techniques used is time discretization.

Again consider a diffusion model for the state variables X with time span Δt. Then the conditional distribution of $X_{t+\Delta t}$ is given by:

$$X_{t+\Delta t} = X_t + \int_{s=t}^{t+\Delta t} \mu(X_s, \Theta) ds + \int_{s=t}^{t+\Delta t} \sigma(X_s, \Theta) dW_s.$$

However, this distribution is difficult to characterize since the distributions of $\int_{s=t}^{t+\Delta t} \mu(X_s, \Theta) ds$ and $\int_{s=t}^{t+\Delta t} \sigma(X_s, \Theta) dW_s$ are unknown. To solve this problem, we could use the Euler scheme to simulate such distribution by assuming sufficiently short time increments Δt. According to Euler scheme, we have that

$$\int_{s=t}^{t+\Delta t} \mu(X_s, \Theta) ds \approx \mu(X_t, \Theta)\Delta t, \text{ and } \int_{s=t}^{t+\Delta t} \sigma(X_s, \Theta) dW_s \approx \sigma(X_t, \Theta)(W_{t+\Delta t} - W_t)$$

Hence, $X_{t+\Delta t} \approx X_t + \mu(X_t, \Theta)\Delta t + \sigma(X_t, \Theta)(W_{t+\Delta t} - W_t)$, which implies that $(X_{t+\Delta t} - X_t|X_t, \Theta) \sim N(\mu(X_t, \Theta)\Delta t, \sigma^2(X_t, \Theta)\Delta t)$.

Similar approximation argument could be applied to other continuous-time models. Alternatively, one could also use Milstein scheme to achieve higher order of accuracy in approximation.

3 Overview of Markov Chain Monte Carlo Method

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from a target distribution based on constructing a Markov chain that has the desired distribution as its equilibrium. In our case, the target distribution is $p(\Theta, X|Y)$. According to Hammersley-Clifford theorem, we know that any joint distribution can be characterized by its complete set of conditional distributions. Therefore, $p(X|\Theta, Y)$ and $p(\Theta|X, Y)$ would completely characterize the joint distribution $p(\Theta, X|Y)$.

MCMC provides a method to combine the information in $p(X|\Theta, Y)$ and $p(\Theta|X, Y)$ to characterize the joint distribution. Consider the following algorithm: given initial values $X^{(0)}$ and $\Theta^{(0)}$, iteratively sample $X^{(i)}$ from $p(X|\Theta^{(i-1)}, Y)$ and sample $\Theta^{(i)}$ from $p(\Theta|X^{(i)}, Y)$ for N iterations. This generates a sequence of random variables $\{X^{(i)}, \Theta^{(i)}\}_{i=1}^{N}$. This sequence is then a Markov Chain with properties that the distribution of the chain converges to $p(\Theta, X|Y)$ under certain conditions. If both the conditional distributions of $p(X|\Theta, Y)$ and $p(\Theta|X, Y)$ are known and can be directly sampled from, the algorithm is called a *Gibbs sampler*. Otherwise, if it is difficult to sample from either one of these two conditional distributions, we could instead use the *Metropolis-Hastings algorithm*, which first samples from a proposed distribution and then either accepts this sample with certain probability or retains the old sample from previous iteration.

After the samples have been generated, they can be used for parameter and state variable estimation with the Monte Carlo method. For example, the estimator for Θ_j could be obtained with the posterior sample mean: $\frac{1}{N}\sum_{i=1}^{N}\Theta_j^{(i)}$.

4 Markov Chain Monte Carlo on Financial Models

In this section, we introduce how to apply the MCMC algorithm to some pricing model that are widely used in financial modeling.

4.1 Geometric Brownian Motion

We will start with a simple case, the estimation of a geometric Brownian motion model for the asset price S_t. The price S_t satisfies the following stochastic differential equation:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t,$$

where μ is the rate of return of the asset, and σ is the volatility. In this model, there are no state variables. Therefore, we could drop the X terms. Denote the log-returns by $Y_t = \log \frac{S_t}{S_{t-1}} = \log(S_t) - \log(S_{t-1})$. Then we have a closed form solution for Y_t:

$$Y_t = \left(\mu - \frac{\sigma^2}{2}\right) + \sigma(W_t - W_{t-1}),$$

which further implies that $Y_t \sim N\big(\mu - \frac{\sigma^2}{2}, \sigma^2\big)$.

Our goal is to estimate the parameters based on the observed data Y. In other words, we are interested in $p(\Theta|Y)$, where $\Theta = (\mu, \sigma^2)$, which is fully characterized by $p(\mu|\sigma^2, Y)$ and $p(\sigma^2|\mu, Y)$. Assume we use independent conjugate priors $p(\mu) \sim N(a, A)$ and $p(\sigma^2) \sim IG(b, B)$, where IG denotes the inverse Gamma distribution. Then by Bayes rule, we have that $p(\mu|\sigma^2, Y) \propto p(Y|\mu, \sigma^2)p(\mu)$ and $p(\sigma^2|\mu, Y) \propto p(Y|\mu, \sigma^2)p(\sigma^2)$, where $p(Y|\mu, \sigma^2)$ is the likelihood function of independent normal variables. Calculations show that with the specified prior distributions, we have $p(Y|\mu, \sigma^2) \sim N$ and $p(\sigma^2|\mu, Y) \sim IG$.

Therefore, the MCMC algorithm for Geometric Brownian motion is a two-step Gibbs sampler. Given current values $\mu^{(i)}$ and $(\sigma^2)^{(i)}$, the MCMC algorithm sequentially samples $\mu^{(i+1)} \sim p\big(\mu|(\sigma^2)^{(i)}, Y\big) \sim N$ and $(\sigma^2)^{(i+1)} \sim p\big(\sigma^2|\mu^{(i+1)}, Y\big) \sim IG$.

4.2 Black-Scholes Model for Option Pricing

Our second example is to apply MCMC to the estimation of Black-Scholes model for a call option. The full model is

$$\begin{aligned}
&S_{t+1} = S_t + \int_{s=t}^{t+1} \mu S_s dt + \int_{s=t}^{t+1} \sigma S_s dW_s, \\
&C_t = BS(\sigma, S_t) + \epsilon_t = S_t \Phi(d) - e^{r(T-t)} K \Phi(d - \sigma(T-t)) + \epsilon_t, \epsilon_t \sim N(0, \sigma_c^2), \\
&d = \frac{\log(S_t/K) + (r + \frac{1}{2}\sigma^)(T-t)}{\sigma\sqrt{T-t}},
\end{aligned}$$

where S_t is the equity price and S_t follows a geometric Brownian motion, C_t is the option price, K is the strike price, $\Phi(\cdot)$ is the cumulative distribution function of a standard normal random variable. We also include the term ϵ_t to model the bid-ask spread in the real market. Again denote the log-returns by $Y_t = \log \frac{S_t}{S_{t-1}} = \log(S_t) - \log(S_{t-1})$.

Our goal is to estimate the parameters based on the observed equity prices S, log-returns Y, and option prices C. In other words, we are interested in $p(\Theta|S, C, Y)$, where $\Theta = (\mu, \sigma^2)$, which is fully characterized by $p(\mu|\sigma^2, S, C, Y)$ and $p(\sigma^2|\mu, S, C, Y)$. Since given Y, μ is independent from S and C, we have that $p(\mu|\sigma^2, S, C, Y) = p(\mu|\sigma^2, Y)$. Similar with the case of geometric Brownian motion, $p(\mu|\sigma^2, Y)$ is normal if we assume the prior distribution of μ is also normal.

To estimate the volatility parameter σ^2, by Bayes rule we can decompose $p(\sigma^2|\mu, S, C, Y)$ into $p(\sigma^2|\mu, S, C) \propto p(C|\sigma^2, S)p(Y|\mu, \sigma^2)p(\sigma^2)$. However, it might be difficult to directly sample from $p(\sigma^2|\mu, S, C, Y)$ since the posterior distribution does not have a closed form. To see this, note that $p(C|\sigma^2, S) \propto \prod_{t=1}^T p(C_t|\sigma^2, S_t) \prod_{t=1}^T exp\big(-\frac{(C_t - BS(\sigma, S_t))^2}{2\sigma_c^2}\big)$, where $BS(\sigma, S_t)$ non-linear in σ^2 and S_t.

Therefore, we could apply a Metorpolis-Hastings algorithm to simulate and estimate σ^2. Suppose that we propose the transition kernel $q(\sigma^2) = p(\sigma^2|\mu, Y) \propto p(Y|\mu, \sigma^2)p(\sigma^2)$. Similar to the previous case of geometric Brownian motion, if we assume the prior distribution of σ^2 is inverse Gamma, then $q(\sigma^2)$ also follows inverse Gamma. Hence, given the current value $(\sigma^2)^{(i)}$, the Metropolis-Hastings algorithm to update σ^2 is:

Step 1: Sample $(\sigma^2)^{(i+1)}$ from $q(\sigma^2)\big) \sim IG$.

Step 2: Accept $(\sigma^2)^{(i+1)}$ with probability $\min\left(1, \frac{p\left((\sigma^2)^{(i+1)}|\mu,S,C,Y\right)q\left((\sigma^2)^{(i)}\right)}{p\left((\sigma^2)^{(i)}|\mu,S,C,Y\right)q\left((\sigma^2)^{(i+1)}\right)}\right)$.

5 Simulation studies on Merton's Jump-Diffusion Model

In the previous section, the application of MCMC on two selected pricing models, geometric Brownian motion and Black-Scholes model, is introduced. In this section, we will introduce how to use MCMC on another pricing model, Merton's Jump-Diffusion model. Results from simulation studies as well as real data analysis will also be provided.

Let S_t be the asset price at time t. Then according to Merton's jump-diffusion model, the dynamics of S_t is specified as:

$$d\log S_t = \mu\ dt + \sigma\ dW_t + JdN_t,$$

where dW_t is the increment from standard Wiener process, N_t follows a Poisson process with intensity λ, the jump size $J \sim N(k, s^2)$. Given the observed asset prices $S = \{S_1, \ldots, S_n\}$, this model requires the estimation of parameters μ, σ, λ, k, and s. For a fixed Δt, applying time discretization, we have that

$$\Delta \log S_t = \mu\ \Delta t + \sigma\ \Delta W_t + J\Delta N_t.$$

Assume Δt is sufficiently small, ΔN_t is either 1 with probability $\lambda\Delta t$, or 0 with probability $1 - \lambda\Delta t$.

Further assume there are 250 trading days per year. Hence $n = 250$ and $\Delta t = \frac{1}{250}$. We could then generate the sample paths for S_t by using:

$$\log S_{t+1} - \log S_t \sim \begin{cases} N(\mu\Delta t, \sigma^2\Delta t), \text{ if } U > \lambda\Delta t \\ N(\mu\Delta t + k, \sigma^2\Delta t + s^2), \text{ if } U \le \lambda\Delta t \end{cases},$$

where $U \sim Unif(0,1)$. Assume $S_0 = 20000$, $\mu = 0.01$, $\sigma = 0.2$, $\lambda = 2$, $k = 0$, and $s = 0.1$, five simulated sample paths are shown in Figure 1.

Denote the log-returns by $Y_t = \log \frac{S_t}{S_{t-1}} = \log(S_t) - \log(S_{t-1})$. According to the model setting, there are five parameters in this model. Therefore, we need to develop five groups of conjugate prior and posterior for these parameters.

Keep other parameters fixed, the likelihood function of μ is proportional to a normal density. Therefore, a normal distribution $N(m, \tau^2)$ is a suitable conjugate prior for μ. The posterior distribution can be obtained as another normal distribution. For σ^2, a suitable conjugate prior is $IG(\alpha, \beta)$, then the posterior distribution is another inverse Gamma distribution. For λ, a suitable conjugate prior is $Beta(a, b)$, then the posterior distribution is $beta(a + N, b + n - N)/\Delta t$, where N is the number of jumps from Y_1 to Y_n. Similarly, for k, a suitable conjugate prior is $N(m_J, \tau_J^2)$, then the posterior distribution is $N\left(\frac{\tau_J^2 \sum_{i=1}^N J_i/N + m_J s^2/N}{\tau_J^2 + s^2/N}, \frac{\tau_J^2 s^2}{N\tau_J^2 + s^2}\right)$. Finally, for s^2, if we use $IG(\alpha_J, \beta_J)$ as the conjugate prior, the posterior distribution is then $IG\left(\alpha_J + \frac{N}{2}, \beta_J + \frac{\sum_{i=1}^N (Y_i - k)^2}{2}\right)$.

The above prior and posterior distributions are conditional on the state variables J_i and ΔN_i. This makes the Gibbs sampler procedure more complicated since only Y_i are

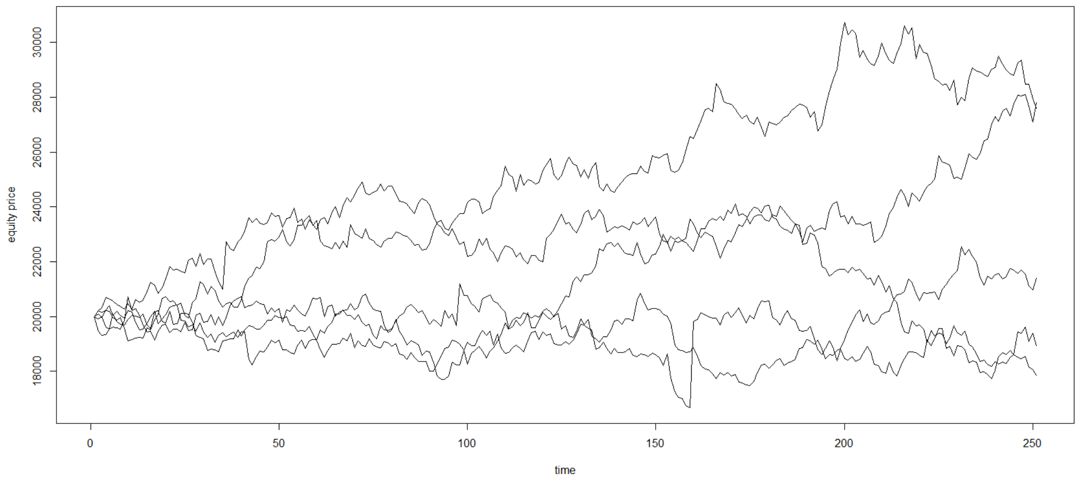

Figure 1: Five sample paths of Merton's jump-diffusion model

observed. As a result, at each time t_i, J_i and ΔN_i must be simulated conditional on the observed values of Y_i. We know that $(Y_i|\Delta N_i = 0) \sim N(\mu\Delta t, \sigma^2\Delta t)$, while $(Y_i|\Delta N_i = 1) \sim N(\mu\Delta t + k, \sigma^2\Delta t + s^2)$. Therefore, by Bayes rule, we have that $P(\Delta N = 1|Y_i) = \frac{P(Y_i|\Delta N_i=1)\lambda\Delta t}{P(Y_i|\Delta N_i=1)\lambda\Delta t + P(Y_i|\Delta N_i=0)(1-\lambda\Delta t)}$. Then $P(\Delta N = 0|Y_i) = 1 - P(\Delta N = 1|Y_i)$.

Additionally, the jump size J_i is needed only when $\Delta N_i = 1$. The conditional distribution of J_i given Y_i is then $(J_i|Y_i) \sim N\big(\frac{(Y_i-\mu\Delta t)/\sigma^2\Delta t + k/s^2}{1/\sigma^2\Delta t + 1/s^2}, \frac{1}{1/\sigma^2\Delta t + 1/s^2}\big)$.

In order to start the Gibbs sampler algorithm, we need to choose the initial values for the parameters, namely $\mu^{(0)}$, $\sigma^{(0)}$, $\lambda^{(0)}$, $k^{(0)}$, and $s^{(0)}$. We may also need the initial values $J^{(0)}$ and $\Delta N^{(0)}$, which could be simulated with the initial parameters. The complete Gibbs sampler algorithm is as follows, where all the posterior distributions have been specified in the previous paragraphs:

Step 1: Sample $\mu^{(j+1)}$ from $p(\mu|(\sigma^2)^{(j)}, \lambda^{(j)}, k^{(j)}, (s^2)^{(j)}, Y_i)$.

Step 2: Sample $(\sigma^2)^{(j+1)}$ from $p(\sigma^2|\mu^{(j+1)}, \lambda^{(j)}, k^{(j)}, (s^2)^{(j)}, Y_i)$.

Step 3: Sample $\lambda^{(j+1)}$ from $p(\lambda|\mu^{(j+1)}, (\sigma^2)^{(j+1)}, k^{(j)}, (s^2)^{(j)}, Y_i)$.

Step 4: Sample $k^{(j+1)}$ from $p(k|\mu^{(j+1)}, (\sigma^2)^{(j+1)}, \lambda^{(j+1)}, (s^2)^{(j)}, Y_i)$.

Step 5: Sample $(s^2)^{(j+1)}$ from $p(s^2|\mu^{(j+1)}, (\sigma^2)^{(j+1)}, \lambda^{(j+1)}, k^{(j+1)}, Y_i)$.

Step 6: Sample $(\Delta N_i)^{(j+1)}$ from $p(\Delta N_j|\mu^{(j+1)}, (\sigma^2)^{(j+1)}, \lambda^{(j+1)}, k^{(j+1)}, (s^2)^{(j+1)}, Y_i)$ for $i = 1, \ldots, n$.

Step 7: Sample $(J_i)^{(j+1)}$ from $p(J_i|\mu^{(j+1)}, (\sigma^2)^{(j+1)}, \lambda^{(j+1)}, k^{(j+1)}, (s^2)^{(j+1)}, Y_i)$ at time t_i where $\Delta N_i = 1$.

Repeat the above steps for M times. The estimator for the five parameters could then be obtained by taking the posterior sample mean.

If we randomly choose one of our simulated paths and carry out the Gibbs sampler algorithm with the number of iterations $M = 10000$, we could obtain the estimators for each parameter is $\hat{\mu} = 0.01494569$, $\hat{\sigma} = 0.1431233$, $\hat{\lambda} = 1.959425$, $\hat{k} = 0.05154602$, and $\hat{s} = 0.1019208$, which are close to the true parameters that we used to generate the path.

The same algorithm could also be applied to the analysis of some real data sets from the financial market. For example, Figure 2 shows the path of the Hang Seng Index, which is the major stock market index in Hong Kong, in all the trading days in 2014.

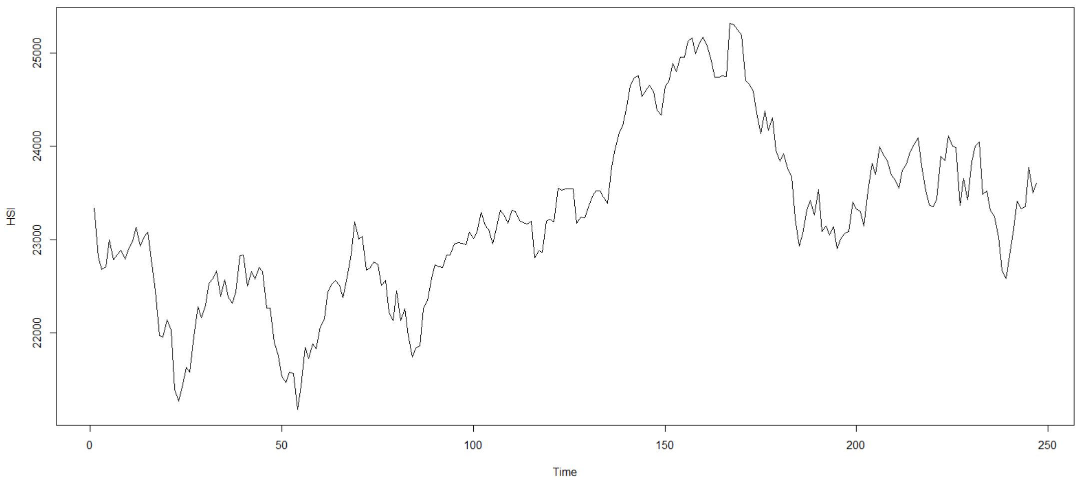

Figure 2: Hang Seng Index in the year of 2014

If we assume that the Hang Seng Index also follows Merton's jump-diffusion model, we can estimate the parameters with the same Gibbs sampler algorithm specified previously. Carrying out the Gibbs sampler with the number of iterations $M = 10000$, we could obtain the estimators for the model parameters are $\hat{\mu} = 0.008668952$, $\hat{\sigma} = 0.199980486$, $\hat{\lambda} = 1.641497666$, $\hat{k} = 0.510247303$, and $\hat{s} = 0.300448979$.

6 Conclusion

In summary, this project discusses about how to apply the Markov Chain Monte Carlo (MCMC) method to the analysis of equity pricing models. Firstly, Bayesian inference on such continuous-time financial models, as well as some important techniques such as time discretization that are used to simulate processes driven by stochastic differential equations, is introduced. Next, a brief overview of the MCMC algorithm is provided, which is further supplemented by how to apply it to models such as geometric Brownian motion and Black-Scholes model. Finally, some simulation as well as real case studies are performed, which applies the MCMC to Merton's jump-diffusion model. The result indicates that the MCMC algorithm is accurate in estimating the model parameters, and the same algorithm could also be applied to the analysis of some real-world data sets.

7 Appendix (Codes for Section 5)

```
## Simulate the sample paths
mu=.01
sigma=.2
lambda=2
k=0
s=.1
m=5
n=250
dt=1/n
S=matrix(log(20000),m,n+1)
for (i in 1:n) {
jump=ifelse(runif(m)<lambda*dt,1,0)
jumpsize=jump*rnorm(m,k,s)
S[,i+1]=S[,i]+rnorm(m,mu*dt,sigma*sqrt(dt))+jumpsize
}
plot(exp(S[1,]),type='l',xlab='time', ylab='equity price', ylim=c(min(exp(S)
),max(exp(S))))
for (i in 2:m) {
points(exp(S[i,]), type='l')
}

## Gibbs sampler
Y=diff(log(S[2,]))
# Set the initial values
mu=0
sigma=.1
lambda=1
k=0.1
s=.5
jump=c(rep(1,n))
jumpsize=Y/2
# Assign the prior
mean.mu=mean(Y)/dt/n
var.mu=1
alpha.sigma=1
beta.sigma=.01
alpha.lambda=1
beta.lambda=.01
mean.k=.5
var.k=.5
alpha.s=1
beta.s=.01
```

```
M=10000
est.mu=0
est.sigma=0
est.lambda=0
est.k=0
est.s=0
for (i in 1:M){
# Calculate posterior
v.mu=1/(1/(sigma^2/n/dt)+1/var.mu)
m.mu=((sum(Y-jumpsize)/n/dt)/(sigma^2/n/dt)+mean.mu/var.mu)/(1/(sigma^2/
n/dt)+1/var.mu)
mu=rnorm(1,m.mu,sqrt(v.mu))

a.sigma=n/2+alpha.sigma
b.sigma=beta.sigma+sum((Y-mu*dt-jumpsize)^2)/2
sigma=sqrt(rinvgamma(1,a.sigma,b.sigma))/sqrt(dt)

J=jumpsize[jump==1]
j=sum(jump)
a.lambda=j+alpha.lambda
b.lambda=n-j+beta.lambda
lambda=rbeta(1,a.lambda,b.lambda)/dt

if (j>1){
v.k=1/(1/(s^2/j)+1/var.k)
m.k=(mean(J)/(s^2/j)+mean.k/var.k)/(1/(s^2/j)+1/var.k)
k=rnorm(1,m.k,sqrt(v.k))

a.s=j/2+alpha.s
b.s=beta.s+sum((J-k)^2)/2
s=sqrt(rinvgamma(1,a.s,b.s))
}

pjump=1/(1+(1-lambda*dt)/(lambda*dt)*sqrt((sigma^2*dt+s^2)/(sigma^2*dt))*
exp(-(Y-mu*dt)^2/(sigma^2*dt)/2+(Y-mu*dt-k)^2/(sigma^2*dt+s^2)/2))
jump=ifelse(runif(n)<pjump,1,0)
var.jump=1/(1/s^2+1/sigma^2/dt)
mean.jump=((Y-mu*dt)/sigma^2/dt+k/s^2)*var.jump
jumpsize=jump*(rnorm(n)*sqrt(var.jump)+mean.jump)
est.mu=est.mu+mu/M
est.sigma=est.sigma+sigma/M
est.lambda=est.lambda+lambda/M
est.k=est.k+k/M
est.s=est.s+s/M
```

```
}
c(est.mu, est.sigma, est.lambda, est.k, est.s)

## Hang Seng Index
dat=read.csv("HSI.csv")
HSI=rev(dat[,5])
plot.ts(HSI)
Y=diff(log(HSI))
n=length(Y)
dt=1/n
# Then use the same code as in the simulation case
```

References

Chan, N. H. and Wong, H. Y. (2013). *Handbook of financial risk management : simulations and case studies*. Wiley handbooks in financial engineering and econometrics. Hoboken, NJ : Wiley.

Creal, D. (2012). A survey of sequential monte carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296.

Johannes, M. and Polson, N. (2009). Markov chain monte carlo. In *Handbook of Financial Time Series*, pages 1001–1013. Springer.

Murphy, J. (2010). Bayesian methods for high frequency financial time series analysis. *PhD First Year Report, Cambridge University Department of Engineering*.

Pollard, M. and Smith, T. (2007). Markov chain monte carlo analysis of option pricing models. Technical report, Working paper.

Tsay, R. S. (2005). *Analysis of financial time series*, volume 543. John Wiley & Sons.