Jdmbs: An R Package for Monte Carlo Option Pricing Algorithms for Jump Diffusion Models with Correlational Companies

Okada Algorithm
Private Invention
Research Laboratory

Shizuoka City, Japan

Masashi Okada

okadaalgorithm@gmail.com
https://github.com/jirotubuyaki

A Vignette

Jdmbs version 1.4

2020-04-27

Abstract

Option is a one of the financial derivatives and its pricing is an important problem in practice. The process of stock prices are represented as Geometric Brownian motion or jump diffusion processes. In this package, algorithms and visualizations are implemented by Monte Carlo method in order to calculate European option price for three equations by Geometric Brownian motion and jump diffusion processes and furthermore a model that presents jumps among companies affect each other.

Introduction

In the early 1970's, Back and Scholes[1] proposed a method in order to calculate option price. For option pricing, the method to numerically solve Black–Scholes equation that represented as partial differential equation and the method to solve equations directly or monte carlo method are proposed. The processes of stock prices are basically represented as Geometric Brownian motion. But the model can not explain the all aspect of a stock process. Because jump diffusion processes[2] that occur jump phenomena are researched. In this package, algorithms and visualizations are implemented by Monte Carlo method in order to calculate European option price for three equations by Geometric Brownian motion and jump diffusion processes and furthermore a model that presents jumps among companies affect each other. Finally we explain the models and how to use it in this paper.

Background

Geometric Brownian motion

The processes of stock prices are basically represented as Geometric Brownian motion.

$$dS(t) = \alpha(t)S(t)dt + \sigma(t)S(t)dW(t), \tag{1}$$

where $S(t)$ denotes stock price at time t. α is a drift parameter and σ is a volatility. $W(t)$ is Brownian motion. Solution of the equation is given by

$$S(t) = S(0)exp\{\sigma W(t) + (\alpha - \frac{1}{2}\sigma^2)t\}. \tag{2}$$

Price of European option

There are many types of options in the stock market. European call option can not execute until the duration of T is finished, and its strike price is K. Option prices are calculated under a risk-neutral probability. European call option price is given by

$$Price = E[max(S(T) - K, 0)], \tag{3}$$

where $E[x]$ denotes expected value. European put option price is given by

$$Price = E[max(K - S(T), 0)].$$ (4)

Poison Process

The Poisson process presents random phenomena somethings happened in unit time. It is widely used in order to model random points in time. Poison process is given by

$$P(N_t = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!},$$ (5)

where λ is the arrival intensity. k is a number something happen in unit time.

Jump Diffusion Model

Jump diffusion model is added jump phenomena into Geometric Brownian motion. Solution of the equation are given by

$$S(t) = S(0)exp\{\sigma W(t) + (\alpha - \beta\lambda - \frac{1}{2}\sigma^2)t\} \prod_{i=1}^{N(t)} (Y_i + 1),$$ (6)

where $N(t)$ denotes a number of jumps $\{t = 0 \cdots\}$. λ is a number of jumps in unit time, $\{W(t) : t = 0 \cdots\}$ is a Brownian motion. $(Y_i + 1)$ is a jump percentage at jump i

Correlational Jump Diffusion Model

Standard jump diffusion model causes jumps in the one stock market and it does not affect other companies. In correlational Jumps model, one jump among companies affects other stock prices of companies obeying correlation coefficients. Therefore equations are given by

$$S(t) = S(0)exp\{\sigma W(t) + (\alpha - \beta\lambda - \frac{1}{2}\sigma^2)t\} \prod_{i=1}^{N(t)} (Y_i \times \rho_{xy} + 1),$$ (7)

where ρ_{xy} denotes a correlation coefficient between x th company and yth company.

Correlational Companies Algorithm

In order to calculate correlation coefficients between all pair companies, all paths must be enumerated in a graph structure. We propose algorithm for enumerating correlations in a given circulation graph. This program code produces a matrix of correlation coefficients between all pair companies.
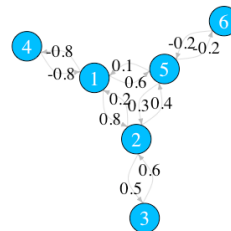

Figure 1: The correlation of companies

This package includes a Perl program in order to calculate the correlations of companies. Please change connect_companies parameters and use like below. output data is data.csv.

```
> perl path.pl
```

Table 1: Result of the correlation coefficients of the companies

	1	2	3	4	5	6
1	1	0.98	0.49	-0.8	0.92	-0.184
2	0.24	1	0.5	-0.192	0.52	-0.104
3	0.144	0.6	1	-0.1152	0.312	-0.0624
4	-0.8	-0.784	-0.392	1	-0.736	0.1472
5	0.16	0.38	0.19	-0.128	1	-0.2
6	-0.032	-0.076	-0.038	-0.0256	-0.2	1

Installation

Jdmbs is available through GitHub (https://github.com/jirotubuyaki/Jdmbs) or CRAN (https://CRAN.R-project.org/package=Jdmbs). If download from Github you can use devtools by the commands:

```
> library(devtools)
> install_github("jirotubuyaki/Jdmbs")
```

Once the packages are installed, it needs to be made accessible to the current R session by the commands:

```
> library(Jdmbs)
```

For online help facilities or the details of a particular command (such as the function normal_bs) you can type:

```
> help(package="Jdmbs")
```

Methods

This package has three methods.
This is a normal model by Monte Carlo:

```
> price <- normal_bs(day=180, monte_carlo=1000,
                     start_price, mu, sigma, K, plot=TRUE
                     )
```

Jump diffusion model by Monte Carlo:

```
> price <- jdm_bs(day=180, monte_carlo=1000,
                  start_price, mu,sigma, lambda, K, plot=TRUE
                  )
```

This is a proposed method by Monte Carlo. companies_matrix must be required:

```
> price <- jdm_new_bs(companies_matrix, day=180, monte_carlo=1000,
                      start_price, mu,sigma, lambda, K, plot=TRUE
                      )
```

Let arguments be:
- companies_data: a matrix of a correlation coefficient of companies
- day: an integer of a time duration of simulation.
- monte_carlo: an integer of an iteration number for monte carlo.
- start_price: a vector of company's initial stock prices.
- mu: a vector of drift parameters of geometric Brownian motion.
- sigma: a vector of volatility parameters of geometric Brownian motion.
- lambda: an integer of how many times jump in unit time.
- K: a vector of option strike prices.
- plot: a logical type of whether plot a result or not.

Let return be:

- price of a list of (call_price, put_price)

Simulation

It is a normal model by Monte Carlo:

```
> price <-  normal_bs(
                    day=100, monte_carlo=10,
                    start_price=c(300,500,850),
                    mu=c(0.1,0.2,0.05),sigma=c(0.05,0.1,0.09),
                    K=c(600,700,1200),
                    plot=TRUE
                    )
```

Jump diffusion by Monte Carlo:

```
> price <- jdm_bs(day=100, monte_carlo=10,
                    start_price=c(5500,6500,8000),
                    mu=c(0.1,0.2,0.05),sigma=c(0.11,0.115,0.1),
                    lambda=2,
                    K=c(6000,7000,12000),
                    plot=TRUE
                    )
```

It is a correlation method by Monte Carlo. corr_matrix must be required:

```
> corr_matrix <- matrix(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9),
                    nrow=3, ncol=3
                    )

> price <- jdm_new_bs(corr_matrix,
                    day=100, monte_carlo=10,
                    start_price=c(5500,6500,8000),
                    mu=c(0.1,0.2,0.05),sigma=c(0.11,0.115,0.1),
                    lambda=2,
                    K=c(6000,7000,12000),
                    plot=TRUE
                    )
```
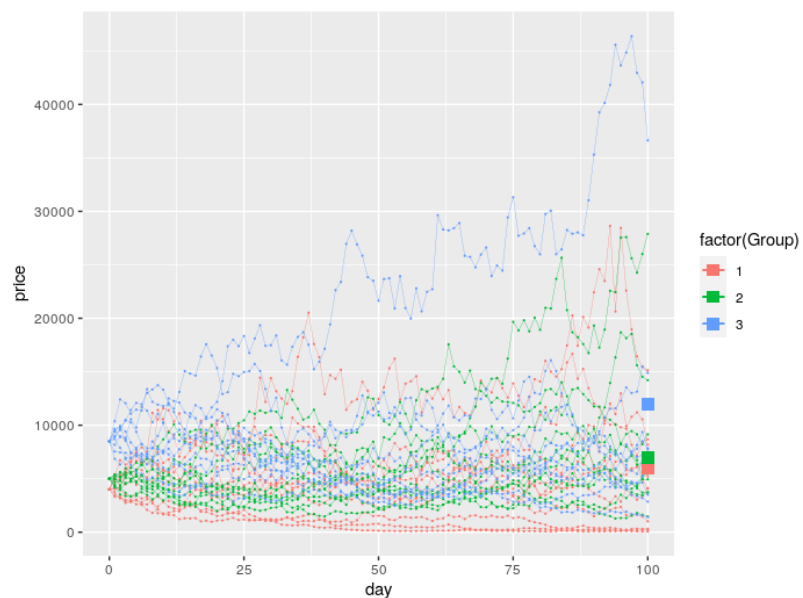

Figure 2: three stock prices: $day = 100, monte\_carlo = 10, start\_price = c(300, 500, 850), mu = c(0.1, 0.2, 0.05), sigma = c(0.05, 0.1, 0.09), K = c(600, 700, 1200)$. Square points are strike prices.
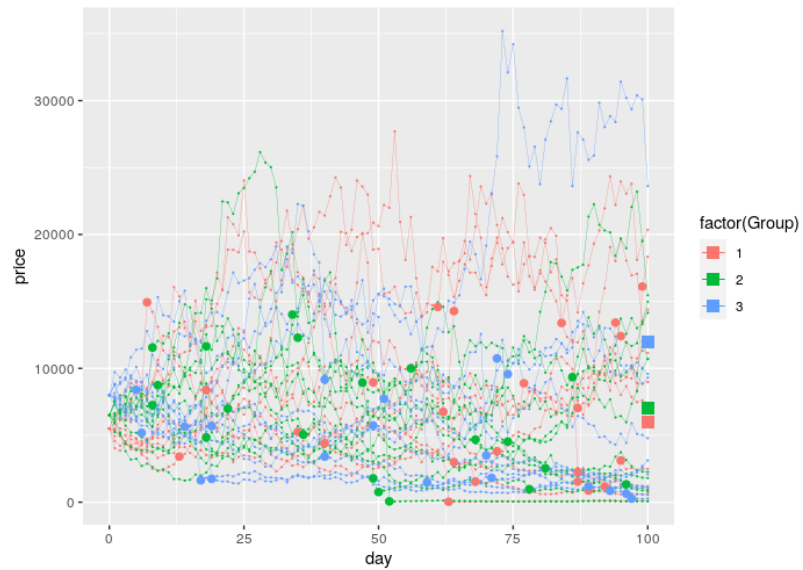
4

Figure 3: three stock prices: $day = 100, monte\_carlo = 10, start\_price = c(5500, 6500, 8000), mu = c(0.1, 0.2, 0.05), sigma = c(0.11, 0.115, 0.1), lambda = 2, K = c(6000, 7000, 12000)$. Square points are strike prices. Big round points are jump points.

Conclusions

Algorithms for option prices were described and explained how to use it. This package can produces option prices of related companies. And several improvements are planed. Please send suggestions and report bugs to okadaalgorithm@gmail.com.

Acknowledgments

This activity would not have been possible without the support of my family and friends. To my family, thank you for much encouragement for me and inspiring me to follow my dreams. I am especially grateful to my parents, who supported me all aspects.

References

[1] Scholes Black and Merton. The pricing of options and corporate liabilities. Journal of Political Economy, 3 Issue 3:637-654, 1973.

[2] Simon S Clift and Peter A Forsyth. Numerical Solution of Two Asset Jump Diffusion Models for Option Valuation. Applied Numerical Mathematics, pages 1-44, 2007.

[3] Steven Shreve. Stochastic Calculus for Finance II: Continuous-Time Models. Springer-Verlag, 2004.