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

A Vignette

Masashi Okada
Okada Algorithm Private Invention Research Laboratory, Japan
okadaalgorithm@gmail.com

Jdmbs version 1.1 2018-01-15

Abstract

Black-Scholes model is important to calculate option premiums in the stock market. Then, variety of jump diffusion models as the time-series of stock prices are studied. In this paper, we propose a new jumps diffusion model with correlational companies in order to calculate option prices in the stock market. This model denotes correlations of companies as a directed graph structure which has the weights of correlation coefficient among companies and it calculates option premiums together. Then, we exhibit the monte-carlo algorithms of a proposed model. Finally, we simulate a new model which is implemented in this package.

Introduction

In the early 1970's, Black-Scholes (BS) model (Black and Merton 1973) was proposed. This model can calculate an option price as market transactions of derivatives. BS model denotes time-series of a stock price as geometric Brownian motion in Stochastic differential equation. Option premiums are calculated from geometric Brownian motion under a risk-neutral probability. The appearance of BS model expanded and grew option markets at a rapid pace. For the achievement, Scholes and Marton won the novel prize. However BS model does not represent all aspects of characteristics of the real market. Therefore the expansions of BS models are studied and proposed. Especially the time-series of a stock price exhibits phenomenons like price jumps. In response to it, Jump diffusion model (Clift and Forsyth 2007) (Shreve 2004) using Poison process to represent jump phenomenons is researched. In this paper, we propose Correlational jumps model which models the correlations of companies in stock prices. A jump phenomenon of one company affects the jumps of other correlational companies as obeying correlation coefficient and it can calculate the premiums of companies together. In this package, the new model and a directed graph of correlational companies algorithm are implemented. Finally, we explain how to use it and simulate it.

Background

Black Scholes Model

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

$$F = E[max(X(T) - K, 0)]$$

$E[x]$ denotes expected value of x. European put option premium is given by

$$F = E[max(K - X(T), 0)]$$

Black-Scholes model is given by

$$e^{-rT}\{e^{\mu+\frac{\sigma^2}{2}}N(\frac{\mu+\sigma^2-InK}{\sigma}) - KN(\frac{\mu-InK}{\sigma})\}$$

where μ presents a draft parameter. It is a trend in the stock price. σ is a volatility. r is a risk-free interest rate. N is a Gauss distribution.

Poison Process

The Poisson process presents random phenomenons happened at any timings. It is widely used to model random points in both time and space. Poison process is given by

$$P(X(t+s) - X(t) = k) = e^{-\lambda s}\frac{(\lambda s)^k}{k!}$$

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

Mixed-Exponential Jump Diffusion Model

Under the mixed-exponential jump diffusion model (MEM), the dynamics of the asset price St are given by

$$\frac{dS(t+1)}{dS(t)} = \mu dt + \sigma dW(t) + d(\sum_{i=1}^{N(t)} Y_i - 1)$$

$$dJ_t = S_t d(\sum_{i=1}^{N(t)} V_i - 1)$$

where r is the risk-free interest rate, σ is the volatility, $\{N(t) : t = 0 \cdots\}$ a Poisson process with rate λ, $\{W(t) : t = 0 \cdots\}$ is a standard Brownian motion.

Correlational Jumps Model

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

$$\begin{pmatrix} \frac{dS_1(t+1)}{dS_1(t)} \\ \frac{dS_2(t+1)}{dS_2(t)} \\ \vdots \\ \frac{dS_n(t+1)}{dS_n(t)} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} d + \begin{pmatrix} \sigma_1 dW_1 t) \\ \sigma_2 dW_2(t) \\ \vdots \\ \sigma_n dW_n(t) \end{pmatrix} + d \begin{pmatrix} J_{1t} \\ J_{2t} \\ \vdots \\ J_{nt} \end{pmatrix}$$

$$d\begin{pmatrix} J_{1t} \\ J_{2t} \\ \vdots \\ J_{nt} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N(t)}(Y_{1i} - 1) * Output_{randam_i 1} \\ \sum_{i=1}^{N(t)}(Y_{2i} - 1) * Output_{randam_i 2} \\ \vdots \\ \sum_{i=1}^{N(t)}(Y_{ni} - 1) * Output_{randam_i n} \end{pmatrix}$$

$$
\begin{aligned}
random_i &\sim U_i(a,b) \\
a &\in \{\cdots, -2, -1, 0, 1, 2 \cdots\} \\
b &\in \{\cdots, -2, -1, 0, 1, 2 \cdots\}
\end{aligned}
$$

where $random_i$ is a n_{th} company and U is a discrete uniform distribution. $Output_{ij}$ is a correlation coefficient from company i to company j.

Correlational Companies Algorithm

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

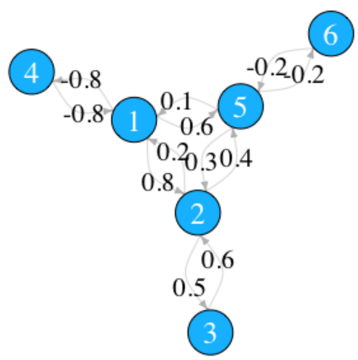


Figure 1. The relation of companies

This package includes a Perl program in order to calculate a 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). 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 for monte carlo:

```
> premium <- normal_bs(companies, simulation.length=180, monte_carlo=1000,
                       start_price, mu, sigma, K, color)
```

Jump diffusion model for monte carlo:

```
> premium <- jdm_bs(companies, simulation.length=180, monte_carlo=1000,
                    start_price, mu,sigma, event_times,jump, K, color)
```

This is a proposed method for monte carlo. companies_data must be required:

```
> premium <- jdm_new_bs(companies_data, companies, simulation.length=180,
                        monte_carlo=1000, start_price, mu,sigma,
                        event_times, jump, K, color)
```

Let's arguments be:

- companies_data : a matrix of a correlation coefficient of companies
- companies : an integer of a company number in order to simulate.
- simulation .length : an integer of a duration of simulation.
- monte_carlo : an integer of iterations of monte carlo.
- start_price : a vector of company's initial stock prices.
- mu : a vector of parameter of geometric Brownian motion.
- sigma : a vector of parameters of geometric Brownian motion.
- event_times : an integer of how many times jump in a unit time.
- jump : a vector of jump parameters.
- K : a vector of option execution prices.

- color : a vector of colors in a plot.

Let's return be:

- premiums of a list of (call_premium, put_premium)

Example

It is normal model for monte carlo:

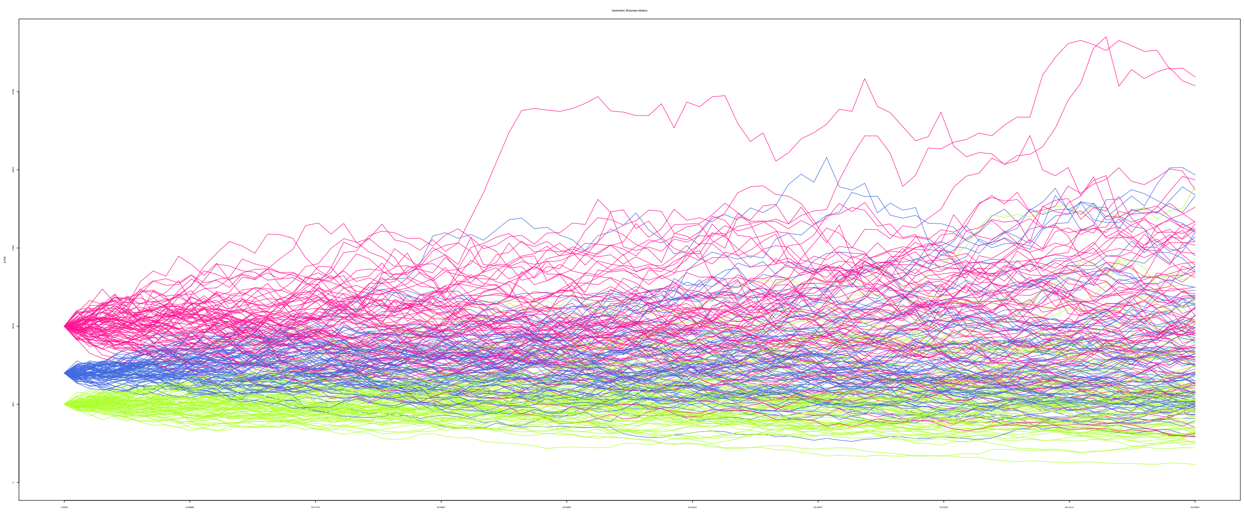
```
> premium <- normal_bs(1, simulation.length=50, monte_carlo=1000,
                      1000, 0.007, 0.03, 3000, "blue")
```

Jump diffusion for monte carlo:

```
> premium <- jdm_bs(3 ,simulation.length=100,monte_carlo=80,
                   c(1000,500,500), c(0.005, 0.025, 0.01),
                   c(0.08,0.04,0.06), 3, c(0.1,0.1,0.1),
                   c(2500,3000,1500), c("red","blue","green"))
```

It is a proposed method for monte carlo. data.csv must be required:

```
> premium <- jdm_new_bs(matrix(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9),
                        nrow=3, ncol=3), 3, simulation.length=100,
                        monte_carlo=80, c(1000,500,500),
                        c(0.005, 0.025, 0.01),c(0.08,0.04,0.06), 3,
                        c(0.1,0.1,0.1), c(2500,3000,1500),
                        c("red","blue","green")
                        )
```



```
## [1] "Call Option Price:"
## [1] 10750.430  8050.604  5643.559
## [1] "Put Option Price:"
## [1]  0.00000 37.03408 14.45809
```

Figure 2. Simulation result of geometric Brownian motion

Conclusions

New algorithm for option prices was described and explained how to use it. This package can produce option prices with 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

Black, Scholes, and Merton. 1973. "The Pricing of Options and Corporate Liabilities" 3 Issue 3: 637–54.

Clift, Simon S, and Peter A Forsyth. 2007. "Numerical Solution of Two Asset Jump Diffusion Models for Option Valuation," 1–44.

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