# Jdmbs: An R package for A Monte Carlo Option Pricing Algorithm for Jump Diffusion Model with Correlation Companies

A Vignette

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

*Jdmbs version 1.0 2017-01-31*

## Abstract

Black-Scholes model is important to calculate option premium in stock market. And variety of jump diffusion model as time-series of stock price are studied. In this paper, we propose new jumps diffusion model with correlational companies in order to calculate option pricing in a stock market. This model express correlational of companies as directed graph structure have a weight of correlational coefficients among companies. And calculate option premiums together. Then we exhibit monte- carlo algorithm of proposed model. After simulate this model comparing with standard jump model and change parameters, we discuss effectiveness of it.

## Introduction

In the early 1970's, Black-Scholes model(Black and Merton 1973) is proposed. This model can calculate an option price as market transactions of derivatives. Black-Scholes model express time-series of stock price as geometric brown motion in stochastic differential equation. Option premium is calculated from exercise price and time duration of option and geometric brown motion under risk-neutral probability. Appearance of Black-Scholes model expanded and grew option market at a rapid pace. For the achievement, Scholes and Marton won the novel prize. But BS model does not represent all aspects of characteristics of a real market. And expansion of BS model is studied and proposed variety of models. Especially time-series of a stock price exhibits phenomenons like a price jump. And in order to modeling it, Jump Diffusion Model(Clift and Forsyth 2007) using Poison Process to express jump phenomenons is proposed. In this paper, I propose Correlational Jumps Model which have correlation of companies in stock price. A jump phenomenon of one company affect jumps of other correlational companies obeying correlation coefficients among companies. And it can calculate premiums of the companies together. In chapter 3, a directed graph of correlational companies algorithm explain. Then in chapter 4, we simulate a proposed mode and explain its algorithm.

## Background

### Black Scholes model

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

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

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

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

Black-Scholes model is given by

where $\mu$ present a draft parameter. it is a trend int the stock price. And $\sigma$ is volatility. $r$ is is the risk-free interest rate.$N$ is gauss distribution.

### Poison Process

The Poisson Process present random phenomenons happened as time sequence. It is widely used to model random points in time and space. Poison process is given by

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

where $\lambda$ is the arrival intensity. $k$ is a number something happen.

### The Mixed-Exponential Jump Diffusion Model

Under the mixed-exponential jump diffusion model (MEM), the dynamics of the asset price St under a risk-neutral measure1 P to be used for option pricing is 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, $\sigma$ the volatility, $\{N(t) : t = 0 \cdots\}$ a Poisson process with rate $\lambda$, $\{W(t) : t = 0 \cdots\}$ a standard Brownian motion.

## Correlational Jumps Model

Standard Jump Diffusion model occurs jump in one stock market and it does not affect other companies. In correlational Jumps model one jump among companies affects other stock price 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 discrete uniform distribution. $Output_{ij}$ is a correlation coefficients from $i$ company to \$ j \$. it is from result of algorithm 1.

## Correlation Companies Algorithm

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

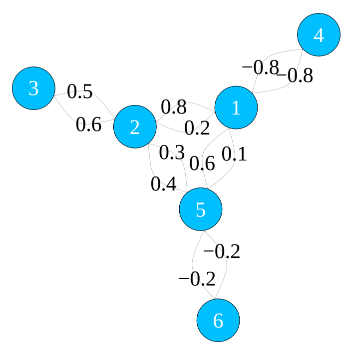



Figure1: The relation of companies

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 for correlation coefficients of 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

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 hmds) you can type:

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

## Method

This package has three method. And it is excused by: It is normal model for monte carlo:

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

Jump Diffusion for monte carlo:

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

It is a proposed method for monte carlo. data.csv 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 args be

- companies_data is a correlation coefficients of companies in "data.csv" file.
- companies is a j of simulate companies.
- simulation.length is a duration of simulation.
- monte_carlo is a iteration j of monte carlo.
- start_price is a vector of initial price of j stock prices.
- mu is a vector of parameters of geometric brown motions.
- sigma is a voctor of parameters of geometric brown motions.
- event_times is somethings happen how many times in Unit time.
- jump is a vector of jump parameter.
- K is a vector of option execution prices.
- color is a vector of colors in plot.

Let's return be

- premium of a list with (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")
```

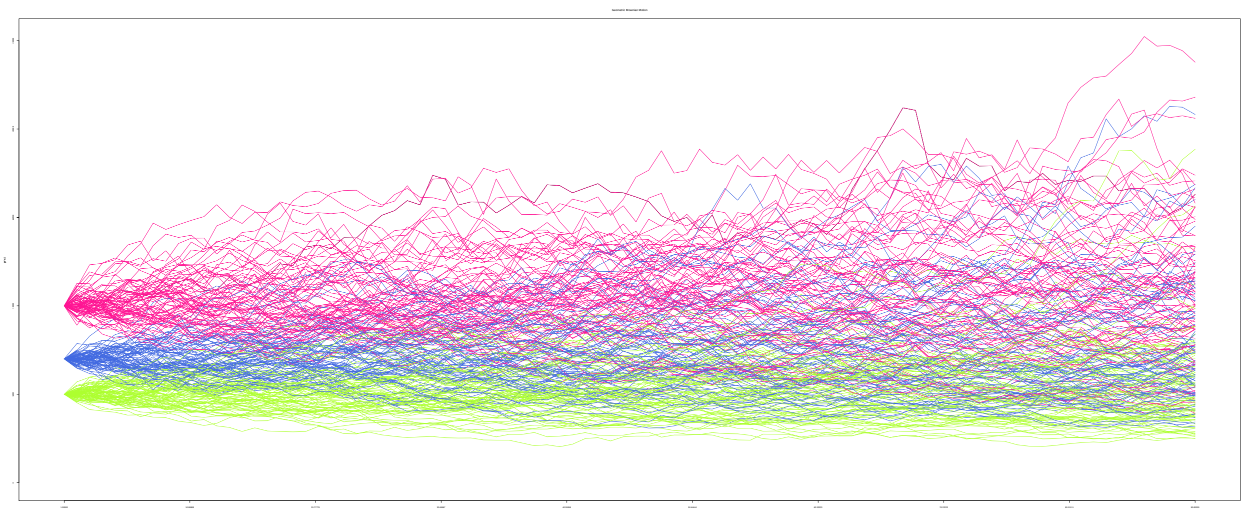
Jump Diffusion for monte carlo:

```
> 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] 10940.768  7786.305  5749.784
## [1] "Put Option Price:"
## [1] 0 0 0
```

Figure2: Simulation Result of Geometric Brownian Motion

## Conclusions

New algorithm for option price is described and explain how to use. This package can produce a option price 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 lots of 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.