

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

A Vignette

*Masashi Okada*

*Okada Algorithm Private Invention Research Laboratory*

*okadaalgorithm@gmail.com*

*Jdmbs version 1.1 2018-01-11*

## Abstract

Black-Scholes model is important to calculate option premiums in the stock market. And variety of jump diffusion models as 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 pricing in the stock market. These models express correlations of companies as a directed graph structure which has a weight of correlational coefficients among companies. And It calculates option premiums together. Then we exhibit monte-carlo algorithms of proposed models. Then we simulate a new model which is proposed in this package.

## 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 models express time-series of a stock price as Geometric Brown Motion in Stochastic Differential Equation. Option premiums are calculated from exercise prices and time duration of options and Geometric Brown Motion under risk-neutral probability. Appearance of Black-Scholes model expanded and grew option markets 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 the 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 model it, Jump Diffusion Model(Cliff and Forsyth 2007) (Shreve 2004) using Poisson Process to express jump phenomenons is proposed. In this paper, We propose Correlational Jumps Model which have correlations of companies in stock prices. A jump phenomenon of one company affects jumps of other correlational companies obeying correlation coefficients among companies. And it can calculate premiums of the companies together. In this paper, a directed graph of correlational companies algorithm is implemented. Then we simulate a proposed model.

## Background

### Black Scholes model

There are several types of options in the stock market. European call option can not exercise 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

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

$E[x]$  expresses 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 drift parameter. It is a trend in the stock price. And  $\sigma$  is volatility.  $r$  is the risk-free interest rate.  $N$  is Gauss distribution.

### Poisson Process

The Poisson Process present random phenomenons happened at any timings. It is widely used to model random points in both time and space. Poisson 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  $S_t$  under a risk-neutral measure  $\mathbb{P}$  to be used for option pricing is given by

$$\begin{aligned} \frac{dS(t+1)}{dS(t)} &= \mu dt + \sigma dW(t) + d\left(\sum_{i=1}^{N(t)} Y_i - 1\right) \\ dJ_t &= S_t d\left(\sum_{i=1}^{N(t)} V_i - 1\right) \end{aligned}$$

Where  $r$  is the risk-free interest rate,  $\sigma$  the volatility,  $\{N(t) : t = 0 \dots\}$  a Poisson process with rate  $\lambda$ ,  $\{W(t) : t = 0 \dots\}$  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

$$\begin{aligned} \begin{pmatrix} \frac{dS_1(t+1)}{dS_1(t)} \\ \frac{dS_2(t+1)}{dS_2(t)} \\ \vdots \\ \frac{dS_n(t+1)}{dS_n(t)} \end{pmatrix} &= \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} dt + \begin{pmatrix} \sigma_1 dW_1(t) \\ \sigma_2 dW_2(t) \\ \vdots \\ \sigma_n dW_n(t) \end{pmatrix} + d \begin{pmatrix} J_{1t} \\ J_{2t} \\ \vdots \\ J_{nt} \end{pmatrix} \\ d \begin{pmatrix} J_{1t} \\ J_{2t} \\ \vdots \\ J_{nt} \end{pmatrix} &= \begin{pmatrix} \sum_{i=1}^{N(t)} (Y_{1i} - 1) * Output_{random_{i1}} \\ \sum_{i=1}^{N(t)} (Y_{2i} - 1) * Output_{random_{i2}} \\ \vdots \\ \sum_{i=1}^{N(t)} (Y_{ni} - 1) * Output_{random_{in}} \end{pmatrix} \end{aligned}$$

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

Where  $random_i$  is a  $n_{th}$  company. And  $U$  is discrete uniform distribution.  $Output_{ij}$  is a correlation coefficients 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 enumeration correlations in a given circulation graph. This program code produces a matrix of correlation coefficients between all pair companies.

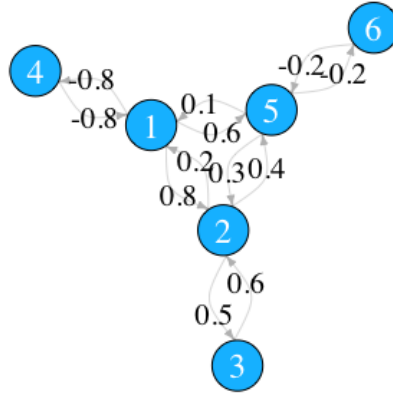


Figure1: The relation of companies

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 `normal_bs`) 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`: 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:

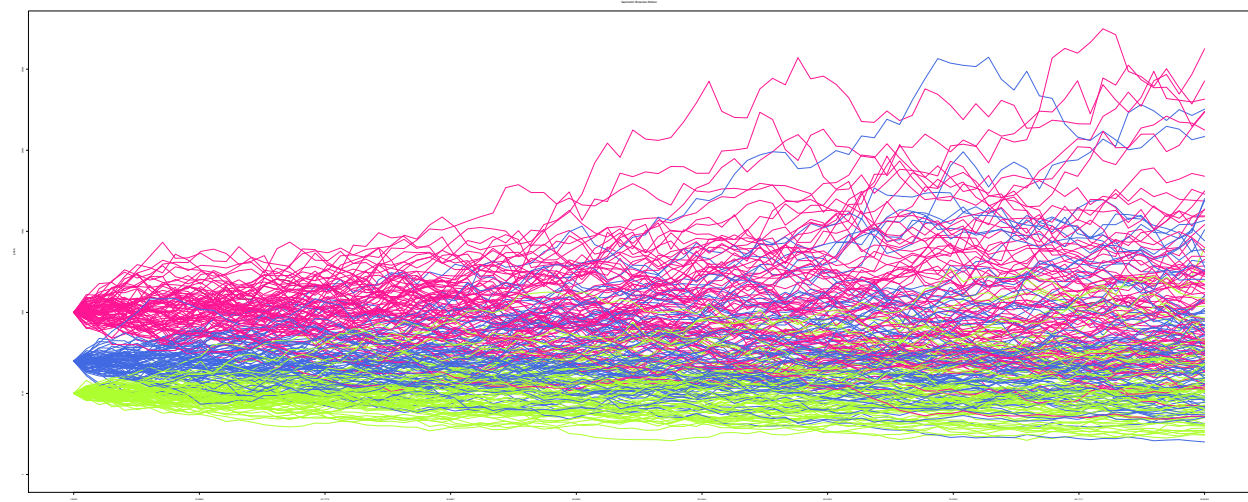
```
> premium <- normal_bs(1, simulation.length=50, monte_carlo=1000,
                        1000, 0.007, 0.03, 3000, "blue")
```

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] 11139.892 8025.443 5662.496
## [1] "Put Option Price:"
## [1] 0.00000 25.11782 0.00000
```

Figure2: Simulation Result of Geometric Brownian Motion

## Conclusions

New algorithm for option prices is described and explained how to use it. 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](mailto: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.
- Shreve, Steven. 2004. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer-Verlag.