Simulation of the Lévy processes

Numerical Introductory Course

Roman Lykhnenko

Ladislaus von Bortkiewicz Chair of Statistics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de/

Outline

1. Motivation
2. Lévy processes
3. S3 class system in R
4. Compound Poisson process
5. Jump-diffusion process
6. Stable Lévy process
7. Conclusion

Options

The buyer of the **call option** has the right to buy an agreed quantity of a particular commodity (**underlying**) from the seller of the option at a certain time (**expiration date**) for a certain price (**strike price**).
Hence the pay-off, i.e. the value of the call option at expiry, is

$$H_T = \max\{S_T - K, 0\}$$

Option pricing

Value of option may be computed as discounted expectation of the terminal payoff with respect to some martingale measure

$$H_0 = P(0, T)\,\mathrm{E}_Q(H_T).$$

In the **Black-Scholes model**, the risk-neutral dynamics of an asset price is described by the exponential of a Brownian motion with drift:

$$S_t = S_0 \exp(B_t)$$

$$B_t = (r - \sigma^2/2)t + W_t,$$

W_t is a Wiener process

Exponential Levy models

By replacing Brownian motion by a more comprehensive stochastic process:

$$S_t = S_0 \exp(rt + X_t),$$

(X_t is a Lévy process)
a tractable class of risk neutral models generalizing the Black-Scholes model can be obtained.
One needs to simulate trajectories of the Lévy process to compute:

$$H_0 = P(0, T)\,\mathrm{E}_Q(H_T).$$

e.g. Monte Carlo simulation.

Outline

1. Motivation ✓
2. Lévy processes
3. S3 class system in R
4. Compound Poisson process
5. Jump-diffusion process
6. Stable Lévy process
7. Conclusion

Lévy processes

Definition

A stochastic process $X = \{X_t : t \geq 0\}$ *is said to be a Lévy process if it satisfies the following properties:*

(i) $X_0 \overset{\text{a.s.}}{=} 0$;

(ii) *For any* $0 \leq t_1 < t_2 < \cdots < t_n < \infty, X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \ldots, X_{t_n} - X_{t_{n-1}}$ *are independent;*

(iii) $X_t - X_s \overset{\mathscr{L}}{=} X_{t-s}$, *for any* $s < t$;

(iv) $\lim_{h \to 0} P(|X_{t+h} - X_t| > \varepsilon) = 0$, *for any* $\varepsilon > 0$ *and* $t \geq 0$.

Different Lévy processes

1. Compound Poisson
2. Jump diffusion
3. Stable

OOP programming concepts have been used to organize the code(S3 class system in R). Simulation class has been created for each process.

```
> # object of the class compoundPoisson
> instance1 = compoundPoisson(0,20)

> class(instance1)
[1] "compoundPoisson"

# check attributes/fields of the created object
> str(instance1)
List of 4
 $ processValues: num [1:100001] 0 0 0 0 0 0 0 0 0 0 ...
 $ time         : num [1:100001] 0e+00 1e-05 2e-05 3e-05 4e-05 5e-05 6e-05 7e-05 8e-05 9e-05 ...
 $ drift        : num 0
 $ jumpIntensity: num 20
 - attr(*, "class")= chr "compoundPoisson"
```

S3 class system in R

```
# get drift
> instance1$drift
[1] 0

# get jump intensity
> instance1$jumpIntensity
[1] 20

# get all methods of the class compoundPoisson
> methods(class="compoundPoisson")
[1] plot            returnsDensity returnsPlot
```

Definition of Compound Poisson process

A compound Poisson process, parameterised by a rate λ and jump size distribution G, is a process $\{\, Y(t) : t \geq 0 \,\}$ given by

$$Y(t) = \sum_{i=1}^{N(t)} D_i,$$

where $\{\, N(t) : t \geq 0 \,\}$ is a Poisson process with rate λ, and $\{\, D_i : i \geq 1 \,\}$ are independent and identically distributed random variables, with distribution function G, which are also independent of $\{\, N(t) : t \geq 0 \,\}$.

For greater generality drift b can be also added

$$Y(t) = bt + \sum_{i=1}^{N(t)} D_i$$

Algorithm for simulation

1. Simulate a random variable N from Poisson distribution with parameter λT. Then N gives the total number of jumps on the interval $[0, T]$
2. Simulate N independent r.v. U_i, uniformely distributed on the interval $[0, T]$. These variables correspond to the jump times.
3. Simulate jump sizes: N independent r.v. Y_i with given law.

   The trajectory is given by

$$Y(t) = bt + \sum_{i=1}^{N} \mathbf{1}_{U_i < t} Y_i$$

```
> plot(instance1, "compoundPoissonDrift0JumpInt20.
  pdf")
```

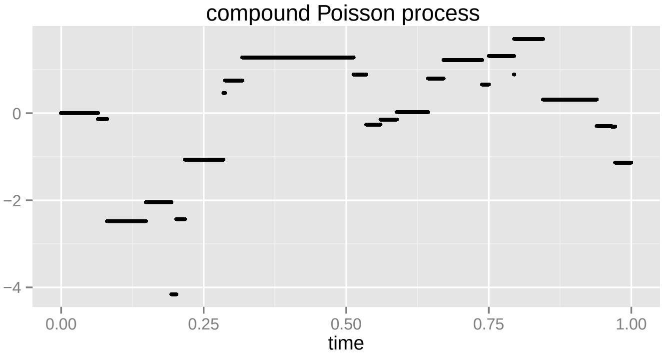

Figure 1: drift is 0, jump intensity is 20

```
> returnsPlot(instance1)
```

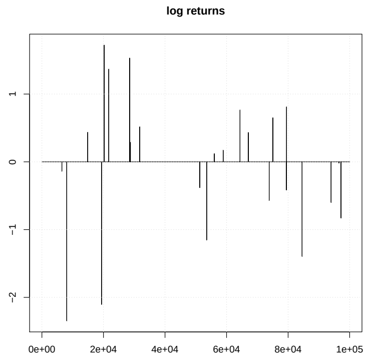

Figure 2: drift is 0, jump intensity is 20

```
> returnsDensity(instance1)
```

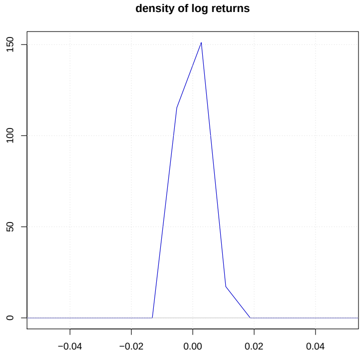

Figure 3: drift is 0, jump intensity is 20

```
> instance1 = compoundPoisson(5,10)
> plot(instance1, "compoundPoissonDrift5JumpInt10.
  pdf")
```

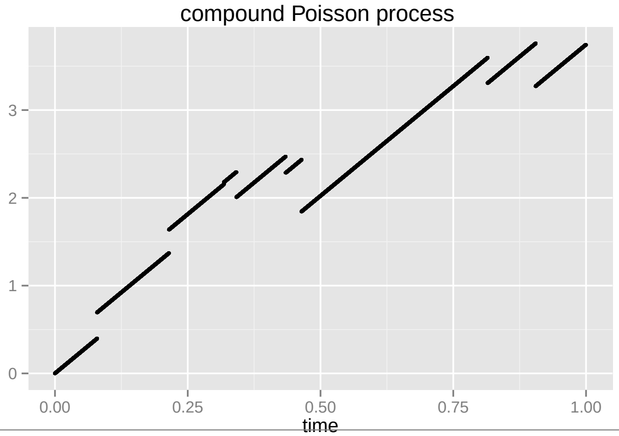

Figure 4: drift is 5, jump intensity is 10

```
> returnsPlot(instance1)
```

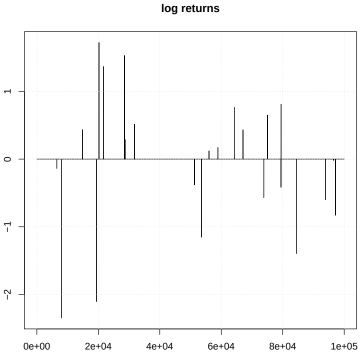

Figure 5: drift is 5, jump intensity is 10

```
> returnsDensity(instance1)
```

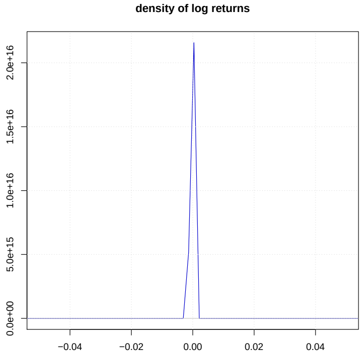

Figure 6: drift is 5, jump intensity is 10

Definition of jump-diffusion process

A Levy process of jump-diffusion type has the following form:

$$X(t) = bt + cW_t + \sum_{i=1}^{N(t)} Y_i,$$

where $\{ N(t) : t \geq 0 \}$ is a Poisson process counting the jumps of X, and Y_i are jump sizes(i.i.d. variables). It is especially important to specify the distribution of jump sizes. In the **Merton model**, jumps in the log-price X_t are assumed to have a Gaussian distribution. In the **Kou model**, the distribution of jump sizes is asymetric exponential.

Algorithm for simulation

Simulation of $(X_1, \ldots, X_n)$ fot n fixed times $t_1, \ldots, t_n$

1. Simulate n independent centered Gaussian random variables G_i with variance $(t_i - t_{i-1})\sigma^2$
2. Simulate the compound Poisson part The discretized trajectory is given by

$$X(t_i) = bt_i + \sum_{i=1}^{N} G_k + \sum_{i=1}^{N} \mathbf{1}_{U_i<t} Y_i$$

```
> instance1 = jumpDiffusion(0,20)
> str(instance1)
List of 6
 $ time              : num [1:10001] 0e+00 1e-04 2e
   -04 3e-04 4e-04 5e-04 6e-04 7e-04 8e-04 9e-04 ...
 $ jump_diffusion    : num [1:10001] 0.03503 0.01868
    0.0011 -0.00819 -0.03587 ...
 $ gaussian_component: num [1:10001] 0.03503 0.01868
    0.0011 -0.00819 -0.03587 ...
 $ jump_component    : num [1:10001] 0 0 0 0 0 0 0 0
    0 0 ...
 $ drift             : num 0
 $ jump_intensity    : num 20
 - attr(*, "class")= chr "jumpDiffusion"
```

```
> plot(instance1, "jumpDiffDrift0JumpInt20.pdf")
```

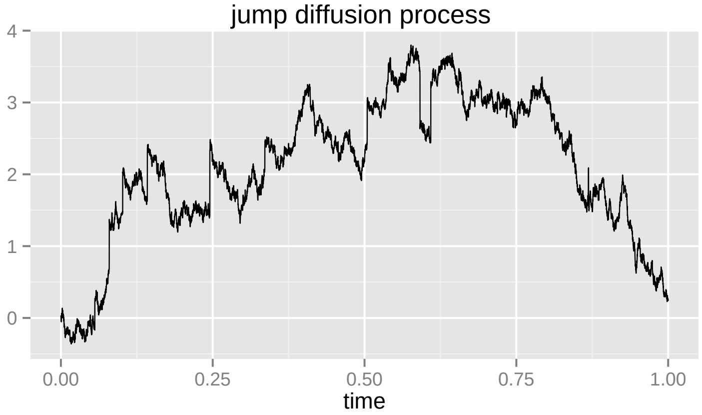

Figure 7: drift is 0, jump intensity is 20

```
> gaussianComponentPlot(instance1)
```

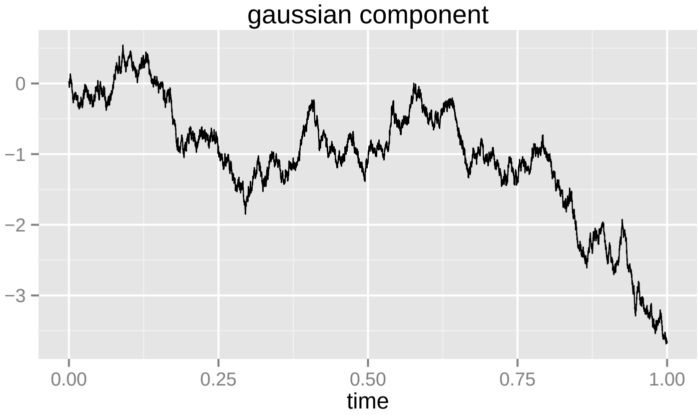

Figure 8: drift is 0, jump intensity is 20

```
> jumpComponentPlot(instance1)
```

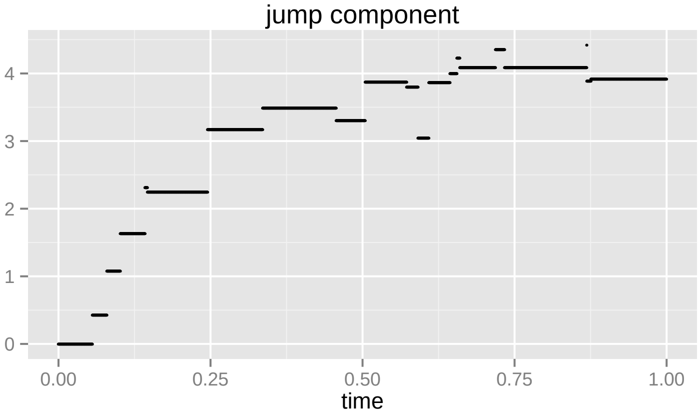

Figure 9: drift is 0, jump intensity is 20

```
> # point plot
> plot(instance1, "jumpDiffDrift0JumpInt20.pdf")
```

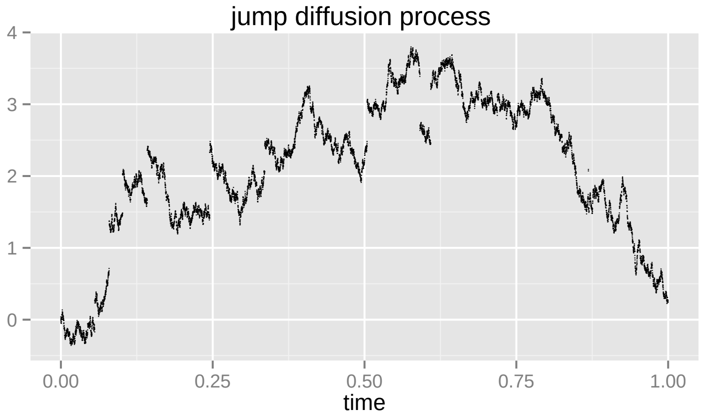

Figure 10: drift is 0, jump intensity is 20

```
> returnsDensity(instance1)
```

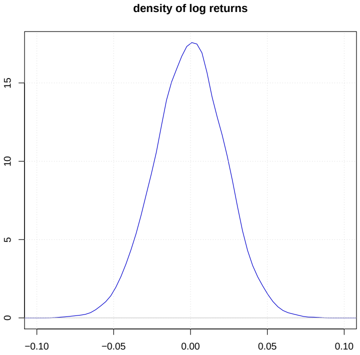

Figure 11: drift is 0, jump intensity is 20

```
1 > returnsPlot(instance1)
```

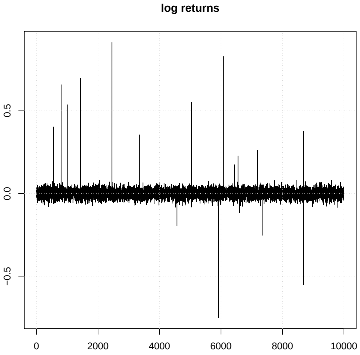

Figure 12: drift is 0, jump intensity is 20

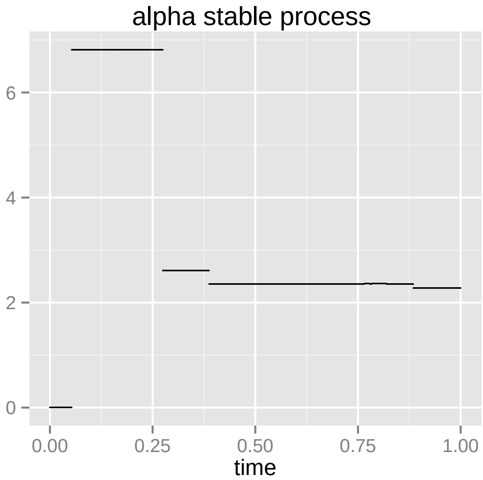

Figure 13: α is 0.5

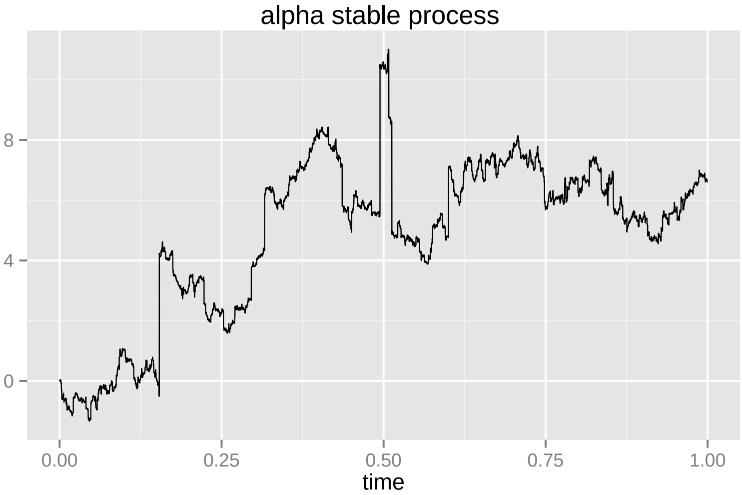

Figure 14: α is 1.9

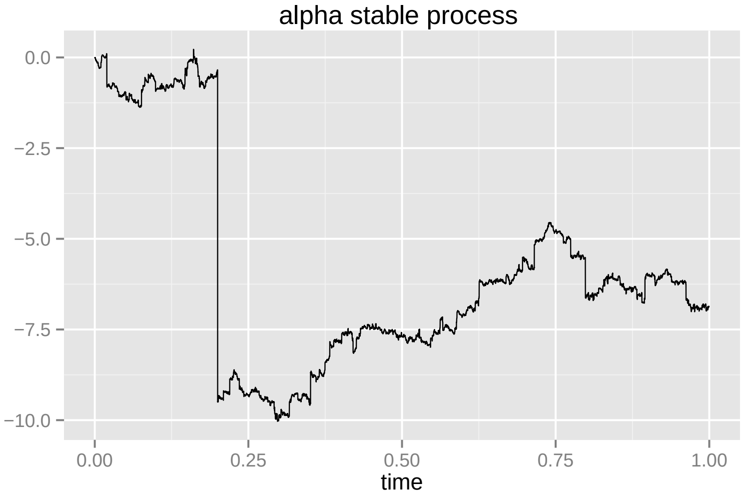

Figure 15: α is 1.7

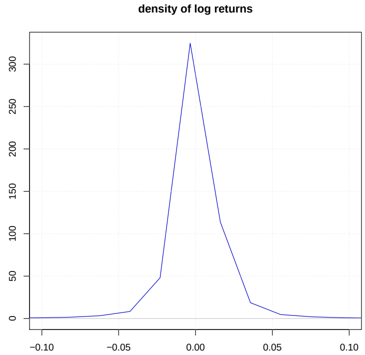

Figure 16: α is 1.7

Properties

- When α is small, the process has very fat tails, and the trajectory is dominated by big jumps.
- When α is large, the behavior is determined by small jumps
- Stable processes and many other Lévy processes can be approximated by a combination of compound Poisson process and Brownian motion.

Summary

Need to simulate stochastic processes to model dynamics of the underlying asset (perform Monte-Carlo to price derivatives)
Simulation class was created for

- Compound Poisson process
- Jump-diffusion process
- Stable Lévy process

References

- Rama Cont, Peter Tankov
  Financial modelling with jump processes
  CRC Press UK, 2004

- Borak, S., Härdle, W. K. and López-Cabrera, B.
  Statistics of Financial Markets: Exercises and Solutions
  Springer-Verlag, Heidelberg, 2010

- Franke, J., Härdle, W. K. and Hafner, C. M.
  Statistics of Financial Markets: An Introduction
  3rd Edition, Springer-Verlag, Heidelberg, 2011