An Introduction to Jump Processes
Merton & Kou Jump Diffusion

Note: *This work is intended for informative and educational purposes only.*

1. Introduction

Geometric Brownian Motion is often the most commonly used process to model stock prices. Its natural extension from traditional (otherwise known as Arithmetic) Brownian Motion to bound the possible values of the process below by zero is clearly real world applicable. But for all the strengths of Geometric Brownian Motion, including its simplicity, it still lacks some typically observed properties from the real world, namely jumps.

While most would agree that stock prices move (at least to some degree) randomly. Large shocks to the markets, either positive or negative, can make a stock jump by some considerable amount. For instance, consider the impact on companies, and digital assets such as Bitcoin, whenever Elon Musk tweets about them. In the most recent case, the price of Twitter jumped over 5% upon announcement Musk was looking to take it private.

So is there a way to incorporate this idea of jumps into our models? Is there such a process that actually includes these random jumps? Well, as it so happens, there is! In this final issue we will consider two well-known examples of jump processes: Merton Jump Diffusion (yes, that is the same Marton from Black-Scholes Merton) and Kou Jump Diffusion.

2. Merton Jump Diffusion

Instead of using the Stochastic Differential Equation representation of our process, we will use equivalent integrated solution,

$$X_t = (\mu_S - \frac{1}{2}\sigma^2)t + \sigma_S W_t + \sum_{i=1}^{N(t)} Z_i. \tag{1}$$

The first part of (1) is simply the integrated from our Geometric Brownian Motion, where we have that X_t is our log-price variable $\log(S_t/S_0)$. However, the second part will likely be new, so let's examine each part.

The Size of the Jumps

In the Merton Jump Diffusion process, the size of the jumps, given by Z_i are assumed to be normally distributed. This means we also need to specify the parameters associated with the size of the jumps, μ_J and σ_J, where the subscript J indicates the jump normal distribution.

Non-Compound Poisson Process

The upper limit of the sum, $N(t)$ is itself a random variable and takes the form of a Poisson Process. Recall that a Poisson Process considers the number of arrivals within a certain time period, with the rate λ dictating the frequency. So how do we adapt this to stock jumps? Well if we set the parameter to an appropriate value, we can essentially bound the values the Poisson Process can take. For instance, by taking the value $\lambda = 0.5$ we will either not jump (the Poisson Process will take value 0) or jump (the Poisson Process will take value 1) a large proportion of the time.

It is worth noting that there is still finite probability that the Poisson Process takes values greater than 1, hence, it is the modeller's choice what to do with these. They could either be capped at 1, or take their original value and simply represent a *very* large jump.

Implementation

I have included a MATLAB implementation of the Merton Jump Diffusion in my *Stochastic Processes* GitHub repository. Feel free to take a look, play with it, and produce the graphs to see the impact of adding this jump component.

3. Kou Jump Diffusion

The Kou Jump Diffusion process follows much the same lines as its Merton cousin. Again we will work with the integrated representation of the solution,

$$X_t = (\mu_S - \frac{1}{2}\sigma^2)t + \sigma_S W_t + \sum_{i=1}^{N(t)} Z_i. \tag{2}$$

Where we have that X_t is our log-price variable $\log(S_t/S_0)$.

The Size of the Jumps

The main difference, however, is the distribution of the jumps. While Merton considered a normal distribution, which is well liked in the industry due to its nice mathematical properties—it does not truly represent empirical stock returns. In reality, there is a higher chance of large losses than those predicted by the normal distribution, in what is known as the 'fat tails problem'.

As a result, Kou introduced the use of the double exponential distribution for the jump parameters. This differs from the traditional Laplace distribution as the two exponential (positive and negative) *need not be the same.* This means that we can incorporate into the process the idea of asymmetric jumps, since jumps downwards tend to be larger than those upwards.

The double exponential random variable takes the following form. Let $\eta_1, \eta_2 > 0$ and $p \in [0, 1]$ be the probability of an upwards jump.

$$p\eta_1 e^{-\eta_1 x}\mathbf{1}[x \geq 0] + (1-p)\eta_2 e^{\eta_2 x}\mathbf{1}[x < 0]. \tag{3}$$

Where $\mathbf{1}[\textit{condition}]$ is the indicator function.

Non-Compound Poisson Process

The Poisson Process $N(t)$ in the upper limit of the sum remains the same as in the Merton case.

Implementation

I have included a MATLAB implementation of the Kou Jump Diffusion in my *Stochastic Processes* GitHub repository. Once again, feel free to take a look, play with it, and produce the graphs to see the impact of adding this jump component.

4. Discussion

Jump processes are an excellent way of including some of the more empirical features of financial markets into modelling. What is worth noting though, particularly in an option pricing environment, is that the introduction of the jumps increases the overall volatility of the stock simulations thereby increasing the price of both calls and puts.

A more detailed discussion of Merton's process can be found in most quantitative finance books, however, for Kou's process, I recommend his original paper—it is well structured and written in a slightly less mathematically formal way than others to make it that bit more accessible.