A Double Exponential Jump Diffusion Model Approximation and Option Pricing using Fourier Transform

Omar Chowdhury, Avi Agarwal, Rohini Jayanthi

May 16, 2016

Abstract

In our paper we empirically test the double exponential jump diffusion model given by Ramezani and Zeng on the S&P 500, estimate parameters using their density function and then compare them to the parameters estimated by minimizing a weighted square norm of the difference between call prices from a characteristic function-based pricing model and from the Black Scholes model.

1 Introduction

The stock evolution specified by the Black Scholes option pricing framework is a Brownian motion with drift. As discussed in our previous paper, the Black Scholes model fails empirical tests. One indicator that Black Scholes is rejected by data is the presence of nonconstant volatility, or 'volatility smile' across option prices. In the Black Scholes world volatility is the volatility of the underlying asset, and as such, it should be consistent across strikes and time periods.

Researchers have explored many different alternative random processes to better mimic stock returns. Some properties that stock processes exhibit are excess kurtosis, or more chance of outliers, negative skewness of returns, and discrete jumps in returns.

In this paper we use Maximum Likelihood Estimation to estimate the parameters of a Double Exponential Jump Diffusion model from SPX returns using the density function derived by Ramezani and Zeng. As well, we provide code for and discuss implementation issues associated with calibrating DJED parameters from market option prices.

2 Literature Review

The motivation behind the double exponential jump diffusion model developed by S.G.Kou (2002) and Ramezani & Zeng (1998) independently stems from trying to answer two most common empirical phenomena; (1) the asymmetric leptokurtic features - in other words, the return distribution is skewed to the left, and has a higher peak and two heavier tails than those of the normal distribution, and (2) the volatility smile. As we showed in our previous project, Bakshi, Kapadia and Madan (2003) detailed the causal relationship between volatility skew and leptokurtic return distribution using model free risk neutral return density estimation.

A better option pricing mechanism should come from a distribution that permits leptokurtic return features in theory as opposed to the assumptions inherent in Black Scholes methodology. Below is a comparison between DEJD and others.

(1) CEV (constant elasticity of volatility): This is a stochastic volatility model that is used to capture stochastic volatility, a leading feature of stock markets. Analytical solutions for path dependent options and interest rate derivatives are available under this model. However, CEV model does not have the leptokurtic feature. It has a thinner right tail than that of the normal distribution. Under CEV, implied volatility can only be monotone function of the strike price. This makes it not efficient to be used in cases where implied volatility is a convex function of strike.

(2) The Normal Jump-Diffusion: This was the first of jump-diffusion models. The jumps are normally distributed and lead to left-skewed returns, implied volatility smile and analytical solutions for call and put options, and interest rate derivatives. DEJD has the advantage of being able to price path dependent options in closed form. This is because it is able to resolves the "overshoot" problem by knowing the exact distribution of overshoot. It is only possible if the jump sizes are exponential. It also creates independence between "first passage time" and "overshoot" which makes their relation easier to model. This feature is unique to exponential type jumps due to their memoryless property.

(3)Stochastic Volatility Models: DEJD and stochastic volatility models complement each other; stochastic volatility can incorporate dependent structure better, while the DEJD has better analytical tractability, especially for path dependent options and interest rate derivatives. An important phenomenon which we hope our model would permit is the daily return distribution to have more kurtosis than the month return distribution. Das and Foresi (1996) point out, this is consistent with models with jumps, but inconsistent with stochastic volatility models. In stochastic volatility models, the kurtosis decreases as the sampling frequency increases, while in jump models the instantaneous jumps are independent of the sampling frequency.

(4) Affine Jump Diffusion Models: Duffie et al. (2000) propose a general class of affine jump-diffusion models which can incorporate jumps, stochastic volatility, and jumps in volatility. DEJD is a special case of it. Because of the special features of the exponential distribution, DEJD leads to analytical solutions for path-dependent options. The DEJD is a simpler and more tractable alternative to the above.

Kou and wang (2004) demonstrate further that the DEJD model is easy to implement and accurate for analytical approximations of finite-horizon American options. Ramezani and Zeng (2004) fit daily returns for NYSE and NASDAQ firms and daily and monthly returns for the S&P 500 and NASDAQ indexes. They use the MLE with BIC criterion to assess the DEJD alongside LDJ (log normal jump diffusion) and ARCH amongst other models. They find that for individual stocks during 10/96 - 12/98, relative to LJD, DEJD provides a better fit or only 11% of the sampled firms. They then compare DEJD to six popular versions of ARCH using data period 1/1999 - 12/2003. DEJD performs better than LJD and ARCH in this period for majority of stocks. They find that for indexes, the ARCH alternatives dominate, but the DEJD provides a better fit than LJD. Their overall empirical evidence in support of DEJD is mixed.

More evidence on the departure of S&P 500 risk neutral return density from the risk neutral density implied by option prices is given by Ait-Sahalia, Wang and Yared (2000). The paper examines whether the information contained in the cross-sectional option prices and the information contained in the time series of underlying asset values are empirically consistent with each other, i.e., whether option prices are rationally determined, under the maintained hypothesis that the only source of risk in the economy's the stochastic nature of the asset price. The restriction posed by this method is the equality between the cross-sectional and time-series SPD (state price density) which becomes over-identifying, and is therefore testable.

Using empirical Girsanov's change of measure, to identify the risk-neutral density from the observed unadjusted index returns, they design four different tests of the null hypothesis that the S&P 500 options are efficiently priced and reject them.

By adding a jump component to the index dynamics, they are able to partly reconcile the differences between the index and option-implied risk-neutral densities, and propose a peso-problem interpretation of this evidence. The peso problem is a situation where options incorporate a premium for the jump risk in the underlying index that is absent from its recorded time series. They find some evidence in favor of that interpretation. Alternatives of moving away from single factor diffusion and non-stochastic volatility, they find may also help in explaining the observed phenomenon. Rather than questioning the rationality of option market pricing, they try to assess the possible limitations of one-factor diffusion structure.

Ait-Sahalia (2004) gives the required mathematical conditions that need to be satisfied to be able to disentangle a Brownian motion from a jump process. These conditions hold even if, unlike the usual Poisson jumps, the jump process exhibits an infinite number of small jumps in any finite time interval (Cauchy jumps), which ought to be harder to distinguish from Brownian noise, itself made up of many small moves. Although we have not tested these for our model but future work that builds up on our paper may find this useful to implement.

3 Option Pricing with Levy Processes

Rama Cont and Peter Tankov's textbook "Financial Modeling with Jump Processes" is the authoritative text on jump models. Chapters 4 (Building Levy Processes), 8 (Stochastic calculus for jump processes), 11 (Risk-neutral modeling with exponential Levy processes), 13 (Inverse problems and model calibration) were used to study the finer details of the DEJD model and verify the methodologies used in this paper.

3.1 Levy Processes

Kou (2002) and Ramezani Zeng (2006) developed the jump diffusion model independently. The classical Black Scholes Stock Price evolution is as follows:

$$S(t) = S(0)exp(X_t)$$

Using Jensen's theorem to cancel out an extraneous term we have

$$X(t) = (\mu - \frac{\sigma^2}{2})t + \sigma\sqrt{t}W(t)$$

This is known as a geometric brownian motion. The risk neutral measure of a Geometric Brownian Motion stock price is $\mu = r$. Lognormal stock return evolution is very restrictive. As well, this theoretical model is rejected by the data due to the presence of nonconstant volatility across stocks, known as a volatility smile. To remedy this we can use an arbitrary Lévy process $X(t)$ instead of a Brownian motion with drift.

Any Lévy process can be decomposed into a Lévy triplet

$$(a, \sigma^2, \prod)$$

This the Lévy Khintchine representation of the process. a and σ^2 are the parameters of a Brownian motion with drift. $\prod$ is a Lévy measure. It represents the rate of arrival and intensity of a compound Poisson process $\sum_{i=1}^{N(t)} Y_i$. There are certain mathematical requirements on the jump part of the process.

$$\Phi(\omega) = E[e^{i\omega X}] = exp(ai\omega - \frac{1}{2}\sigma^2\omega^2 + \int_{R\backslash\{0\}} e^{i\omega x} - 1i\omega I_{|x|<0]} \prod(dx)$$

Consider the stock process below:

$$S(t) = S(0)exp((\mu - \frac{1}{2}\sigma)t + \sigma Z(t)) \prod_{i=0}^{N(t)} V_i$$

Where V_i are arbitrary iid random variables and $N(t)$ is a Poisson distributed random variable with $N(t) \sim Pois(\lambda)$. The logarithmic stock returns are:

$$log(S(t)) = log(S(t)) - log(S(t-1))$$

and as such

$$log(S(t)) = X(t-(t-1))$$

3.2 The Double Exponential Jump Diffusion stock process

Start with the log return process:

$$X(t) = (\mu - \frac{1}{2}\sigma^2)t + \sigma Z(t) + \sum_{i=0}^{N(t)} log(V_t)$$

THe double jump exponential model formulated by Ramzani and Zeng can be formulated two ways: as the sum of a poisson distributed number of iid asymmetric exponential distributions or as two sums of two iid exponential distributions. This is a special case of the Lévy process above with the compound Poisson process specified. In the Kou formulation:

$$log(V) \sim \begin{cases} \xi^+ & \text{with probability } p \\ \xi^- & \text{with probability } 1-p \end{cases}$$

$$\xi^+ \sim exp(\eta_u)$$
$$\xi^- \sim exp(\eta_d)$$

$$f_{log(V)}(v) = p * \eta_u e^{-eta^u v} 1_{v \geq 0} + (1-p) * \eta_d e^{\eta_d v}$$

Where $p \geq 0$ is the probability of an up jump.

Ramezani and Zeng start instead by explicitly defining the distributions of up and down jumps.

$$V^u \sim \text{Pareto}(\eta_u)$$

$$V^d \sim \text{Beta}(\eta_d, 1)$$

So that

$$S(t) = S(0)exp[(\mu - \frac{1}{2}\sigma^2)t + \sigma Z(t)] \prod_{j=u,d} V^j(N^j(\lambda^j t))$$

$$\prod_{j=u,d} V^j(N^j(\lambda^j t)) = \begin{cases} 1 & \text{if } N(\lambda^j t) = 0 \\ \prod_{j=1}^{N(\lambda^j t)} V^j(N^j(\lambda^j t)) & \text{if } N(\lambda^j t) = 1, 2, 3, ... \end{cases}$$

setting $Y = ln(V)$,

$$X(t) = (\mu - \frac{1}{2}\sigma^2)s + \sigma Z(s) + \sum_{i=0}^{N^u(s)} Y_s^u + \sum_{i=0}^{N^d(s)} Y_s^d$$

The Kou formulation does not allow for a density function that supports maximum likelihood estimation. However, the Kou formation has an easily verified characteristic function. Ramezani and Zeng demonstrate that the models are connected by a small change of variables and subsequently refer to the two interchangeably.

Note that the logarithm of a Pareto distribution and the logarithm of -1 times a Beta distribution are both exponential distributions.

$$f_Y(y) = p\eta_u e^{-\eta y} I_{y \geq 0} + (1-p)\eta_d e^{\eta_d y} I_{y<0}$$

This is identical to the distribution of jumps in the Kou model if $p = \frac{\lambda_u}{\lambda}$, $\lambda = \lambda_u + \lambda_d$

Double Jump Exponentially distributed stock returns exhibit excess kurtosis and can exhibit skewness if $\eta_u \neq \eta_d$ and/or $\lambda_u \neq \lambda_d$.

3.3 Density function of the DEJD

Ramezani and Zeng estimate the return process' parameters via Maximum Likelihood Estimation. To use MLE we need to have the full specification of the density function of the DEJD. Ramezani and Zeng chose Pareto and Beta jumps specifically to allow for analytical derivation of the the probability density of the returns.

The derivation is as follows.

Let $N_s^u = m$ and $N_s^d = n$ be the number of up and down jumps during a time interval of length s. In one period we could have

$$\begin{cases} m = 0, n = 0 \\ m = 0, n \geq 1 \\ m \geq 1, n = 0 \\ m \geq 1, n \geq 1 \end{cases}$$

eg, no jumps, only one kind of jump, or both types of jumps.

As stated before the logarithms of the Pareto and Beta distribution are both distributed exponentially.

$$ln(V^u) = Y^u \sim exp(\eta_u)$$

$$-ln(V^d) = -Y^d \sim (\eta_d)$$

As well, the sum of iid exponential distributions is a Gamma distribution

$$Y_i \sim exp(\theta)$$

$$X = \sum_{i=1}^{n} Y_i \sim \Gamma(n, \theta)$$

Let $U = \sum_{i=1}^{N_s^u} Y_i^u > 0$, and $D = \sum_{i=1}^{N_s^d} Y_i^d > 0$ and T = U +D.

$$r(s) = (\mu - 0.5\sigma^2)s + \sigma Z(s) + U + D$$

For $N_s^u = m \geq 1$, the conditional density $U|m \sim \Gamma(m, \eta_u)$ similarly for D, $-D|n \sim \Gamma(n, \eta_d)$. Therefore

$$f_{U|m}(U) = \frac{\eta_u^m}{(m-1)!} U^{m-1} e^{-\eta_u U}$$

and for $N_s^d \geq 1$

$$f_{D|n}(D) = \frac{\eta_d^n}{(n-1)!} (-D)^{n-1} e^{\eta_d D}$$

The distribution of $T|m, n \geq 0$ can be computed using convolution.

$$f_{T|m,n}(t) = \int_{-\infty}^{\infty} f_D(x) f_U(t-x) dx = \frac{\eta_u^m \eta_d^n}{(n-1)!} \int_{-\infty}^{0^t} (-x)^{n-1} (t-x)^{m-1} e^{(\eta_u + \eta_d)x} dx$$

Now we determine the four conditional densities. If m = n =0

$$f_{r(s)|0,0}(r) = \frac{1}{\sqrt{2\pi} * \sigma} e^{-\frac{1}{2\sigma^2 s}(r-\mu s+.5\sigma^2 s)^2}$$

When m = 0 and $n \geq 1$, we find the joint distribution of the sum of independent random variables $\Gamma(n, \eta_d)$ and $N((\mu - \frac{1}{2}\sigma^2)s, \sigma^2 s)$

$$f_{r(s)|0,n} = \frac{\eta_d^n}{(n-1)!\sqrt{2\pi s}\sigma} \int_{-\infty}^{0} (-D)^{n-1} e^{\eta_d x - \frac{1}{2\sigma^2 s}(r-x-\mu s+0.5\sigma^2 s)^2} dx$$

and similarly

$$f_{r(s)|m,0} = \frac{\eta_u^m}{(m-1)!\sqrt{2\pi s}\sigma} \int_{0}^{\infty} (x)^{m-1} e^{-\eta_u x - \frac{1}{2\sigma^2 s}(r-x-\mu s+0.5\sigma^2 s)^2} dx$$

$$f_{r(s)|m,n}(r) = \frac{\eta_u^m \eta_d^n}{(m-1)!(n-1)!\sqrt{2\pi s}\sigma} \int_{-\infty}^{\infty} (\int_{-\infty}^{0\wedge t} (-x)^{n-1}(t-x)^{m-1} e^{(\eta_u+\eta_d)x} dx)$$

$$* e^{-\eta_u t} e^{\frac{1}{2\sigma^2 s}(r-t-\mu s+0.5\sigma^2 s)^2} dt$$

The unconditional density of $s = 1$ period returns can be expressed as a Poisson weighted sum of the above four conditional densities. Let $P(n, \lambda) = \frac{e^{-\lambda}\lambda^n}{n!}$

$$f(r) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} P(n,\lambda_d)P(m,\lambda_u)f_{n,m}(r) = e^{-(\lambda_u+\lambda_d)} f_{0,0}(r) + e^{-\lambda_u} \sum_{n=1}^{\infty} P(n,\lambda_d) f_{0,n}(r)$$

$$+ e^{-\lambda_d} \sum_{m=1}^{\infty} P(m,\lambda_u) f_{m,0}(r) + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} P(m,\lambda_u)P(n,\lambda_d) f_{m,n}(r)$$

4 Fourier Transform methods for option pricing

We follow the Carr-Madan method to find the price of a European call with strike price K. Let X(t) be a Lévy process.

let $k = log(K)$ and let $s = log(S(T))$. s has risk neutral density $q_T(k)$

$$C_T(k) = \int_{k}^{\infty} (e^s - e^k) q_T(s) ds$$

$C_T(k)$ is not square integrable $lim_{k \to -\infty} C_T(k) = S_0$, so we define with $\alpha > 0$

$$c_T(k) \equiv exp(\alpha k) C_T(k)$$

The Fourier transform of $c_T(k)$ is

$$\psi_T(\omega) = \int_{-\infty}^{\infty} e^{i\omega k} c_T(k) dk$$

and the inverse Fourier transform and step to remove the α term is is:

$$C_T(k) = \frac{exp(-\alpha k)}{\pi} \int_0^{\infty} e^{-i\omega k} \psi_T(\omega) d\omega$$

In words, we found the price of the call option in terms of the reverse Fourier transform of the characteristic function of the call option.

Next we find the characteristic function of the call option in terms of the characteristic function of the logarithmic returns.

$$\begin{aligned}
\psi_T(\omega) &= \int_{-\infty}^{\infty} e^{i\omega k} \int_k^{\infty} e^{\alpha k} e^{-rT} * (e^s - e^k) q_T(s) ds dk \\
&= \int_{-\infty}^{\infty} e^{rT} q_T(s) \int_{-\infty}^{s} (e^{s+\alpha k} - e^{(1-\alpha)k}) e^{i\omega k} dk ds \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s) (\frac{e^{(\alpha+1+i\omega)s}}{\alpha + \omega} - \frac{e^{(\alpha+1+i\omega)s}}{\alpha + 1 + i\omega}) ds \\
&= \frac{e^{-rT} \phi(\omega - (\alpha+1)i)}{\alpha^2 + \alpha - \omega^2 + i(2\alpha+1)\omega}
\end{aligned}$$

We then solve for the option price at time T with strike $exp(k)$ by substituting the above equation into the reverse fourier transform specified above it.

Care must be taken in choosing the decay parameter α. Observe that for $\psi(0)$ to be finite, we need that $\Phi(-(\alpha+1)i)$ must be finite. Since stock returns are the exponent of the return distribution, we can express this condition as:

$$E[S_T^{\alpha+1}] < \infty$$

Which can be solved analytically. Carr and Madan recommend that

$$\alpha = \frac{1}{4} \text{Upper Bound on } \alpha$$

The Kou Model has characteristic function

$$e^{i\omega log(S_0) \Psi(T*\omega)}$$

$$\Psi(\omega) = \frac{1}{2}\sigma^2\omega^2 + i\mu\omega + i\omega\lambda(\frac{p}{\eta_u - i\omega} - \frac{1-p}{\eta_d + i\omega})$$

By the change of variables formula connecting the Kou formulation to the Ramezani Zeng formulation above, we set $\lambda = \lambda_u + \lambda_d$ and $p = \frac{\lambda_u}{\lambda}$ in the above.

4.1 Risk Neutral Density

Option prices are computed from the risk neutral probability measure. In this case the drift of the process should be the risk free interest rate. As such we specify a new drift under a new measure

For a jump diffusion process as specified above,

$$\mu' = \lambda(m-1) - r = 0$$

Where m is the mean of the Y_i jump intensities. For the Ramezani Zeng DJED we derived the following via independence between the three random variables, and inclusion of the dividend yield δ:

$$\mu' = r - \delta + \frac{\lambda_u}{\eta_u - 1} + \frac{\lambda_d}{\eta_d + 1}$$

We replace this value of $\mu' for \mu$ in the last line of previous section.

5 Discussion of Implementation

Let $f(r|\theta)$ be the density function specified above. Let theta be $(\mu, \sigma\eta_u, \eta_d, \lambda_u, \lambda_d)^T$.

Maximum Likelihood Estimation of a vector of parameters can be phrased as:

$$\text{minimize} - \sum_{i=1}^{N} log(f(r|\theta))$$

subject to:

$$A\theta \leq b$$

The constraint equation specifies the bounds of each parameter and can also be expressed as

$$(\sigma > 0, \eta_u > 1, \eta_d > 0, \lambda_d > 0, \lambda_u > 0)$$

The problem defined above is both nonconvex and highly nonlinear as it contains among other nonlinear terms summations and oscillating integrals. Specifying a gradient would also be very difficult, and the paper authors do not specify one. Therefore we examined nonlinear optimization tools in Matlab. The authors of the paper discretized the integrals via Gaussian quadrature and empirically tested upper and lower bounds for the integrals and summations. They found that integrals with lower bounds of negative infinity could be truncated at -2 with six digit accuracy. As well they set the following termination condition on the infinite summations. Let

$$S_n = \sum_{i=1}^{n} X_i$$

. Terminate the summation when:

$$2 * |X_{n+1}| \leq FTOL(|S_n| + |S_{n+1}|)$$

They chose $FTOL = 10^{-10}$ because it affords at least 8 digit accuracy.

The single integrals specified in f(r) are relatively well behaved and can be computed with matlab's numerical integration function $integrate(f, lowerbound, upperbound)$. However, the integral

$$f_{r(s)|m,n}(r) = \frac{\eta_u^m \eta_d^n}{(m-1)!(n-1)!\sqrt{2\pi s}\sigma} \int_{-\infty}^{\infty} (\int_{-\infty}^{0\wedge t} (-x)^{n-1}(t-x)^{m-1} e^{(\eta_u+\eta_d)x} dx)$$

$$*e^{-\eta_u t} e^{\frac{1}{2\sigma^2 s}(r-t-\mu s+0.5\sigma^2 s)^2} dt$$

Has the integrating variable of the outside integral as the bound of the inner integral. The authors were unable to find an the existing numerical integration function in MATLAB and Python that could numerically solve this relation. As such we computed the outside integral by discrete summations and then used built in quadrature functions to solve the inner integral.

5.1 Optimization functionality in MATLAB and Python

We originally wrote our optimization in MATLAB. MATLAB's nonlinear optimization function $fmincon$ can take constraints and bounds as input. The optimizations it could use relevant to our purposes were the default $'interior-point'$ (specifically the Quasi-Newton method $BFGS$,The Broyden-Fletcher-Goldfarb-Shanno algorithm, and its limited memory approximation $'L-BFGS'$) and $'sqp'$ (Sequential Quadratic Programming).

Both take as their basic idea the Newton-Raphson method that iterates through points with the objective of finding the point where the function gradient becomes a zero vector. The Newton-Raphson method finds the root of a function g(x) ie g(x) = 0 by iterating through points $x_1, x_2, ..., x*$ in the order

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

To find the minimum of a function f(x), we follow the Newton Method on the function's gradient f'(x).

However, Newton-Raphson requires explicit computation of the gradient and Hessian matrix (first and second derivative) of f(x) at each x in the sequence and it converges on local minima. SQP and BFGS are both non-gradient methods. BFGS is Quasi Newton method that computes an approximation of the gradient from $s_n = x_{n+1} - x_n$ and the difference between the gradients at x_n and x_{n+1} $y_n = g_{n+1} - g_n$. The intuition behind the BFGS Hessian approximation step is that it chooses a Hessian to minimize the norm of the difference of the estimated hessian at the current point and the fixed estimate of the previous Hessian, subject to a tangency relation $H_{n+1}^{-1} y_n = s_n$ and that H^{-1} is symmetric (as second derivatives are symmetric). There is an analytic solution to this relation.

Our MATLAB implementation of the Maximum Likelihood Estimator took several hours to complete one iteration of the algorithm for our SPX dataset. Part of the problem was that MATLAB as a high level environment has significant overhead for function calls and other operations. Individual calls to our density function and its subfunctions (the conditional densities specified in Section 2) took significant amounts of time. By our calculations it would have taken several weeks for the algorithm to complete. Therefore we rewrote our optimization in Python.

The numpy module in Python contains well written nonlinear optimization functions and numerical integration functions with underlying routines written in FORTRAN. Numpy contains implementations of the nonlinear algorithms specified above. As well it has available the Powell Method, which is the algorithm Ramezani and Zeng used in their implementation.

5.2 Numerical Integration for call Prices

The Carr Madan method requires integration over the frequency domain. As we decided to numerically integrate rather than use fast fourier transforms, we needed to use a python module capable of integration in the frequency domain. After some research we found the Python mpmath module for arbitrary floating point precision arithmetic of real and imaginary numbers (http://mpmath.org/). It is a default module in sympy and Sage, an open source mathematics library.

6 Experiments

We conducted several experiments. First we compared the parameter estimates given by feeding the optimization algorithm different starting values. Second, we priced options using these MLE parameter estimates after converting our stock process drift to the risk neutral measure. Third, we attempted to estimate the parameters implied by the market options prices. We justify this theoretically by observing that the existence of the volatility smile rejects the Black Scholes model, so it is valid to calibrate from these prices as the market is using an unknown model (or mix or lack of models) to converge to market prices. This experiment (where we compare the posterior estimates of down jumps and volatility against the predictive implied estimates) would have given us a sense of whether or not options are overpriced in their implied parameters.

6.1 Least squares calibration regularized by regularized by Euclidean distance

Given a prior exponential-Levy model Po with characteristics $(a, \sigma^2, \prod)$; find a parameter vector θ which minimizes

$$\tau(\theta) = \sum_{i=1}^{N} \omega_i |C^\theta(T_i, K_i) - C_i|^2 + \alpha H(\theta)$$

In Cont & Tankov, they choose H to be relative entropy or Kullback Leibler distance $H(\theta) = \mathcal{E}(Q_\theta|P_o)$ of the the pricing measure Q_θ with respect to the prior model P_o. α is the regularization parameter which along with H penalizes the squared error for deviating away from P_o. In our implementation we choose, the euclidean distance between the prior and calibrated parameter vectors.

6.2 Choice of regularization parameter

The above functional consists of two parts: the distance functional, which is convex (at least locally) in its argument θ and the quadratic pricing error which measures the precision of calibration.

The coefficient α, called the regularization parameter defines the relative importance of the two terms: it characterizes the trade-off between prior knowledge of the Levy measure and the information contained in option prices. If Q_θ is large enough, the convexity properties of the entropy functional stabilize the solution and when $\alpha \xrightarrow{0}$, we recover the nonlinear least squares criterion. Therefore the correct choice of α is important: it cannot be fixed in advance but

its "optimal" value depends on the data at hand and the level of error present in it.
One way to achieve this trade-off is by using the Morozov discrepancy principle. Assume that bid and ask price data are available. This means that we can measure the error on inputs by

$$\epsilon_0 = ||C^{bid} - C^{ask}|| \equiv \sqrt{\sum_i \omega_i |C_i^{bid} - C_i^{ask}|^2}$$

Since the "true" price lies somewhere between the bid price and the ask price and not exactly in the middle, it is useless to calibrate the mid-market prices exactly: we only need to calibrate them with the precision ϵ_0. On the other hand, by increasing a we improve the stability, so the best possible stability for this precision is achieved by

$$\alpha^* = sup\{\alpha : ||C^{\alpha(\theta)} - C^{mm}|| \leq \epsilon_o\}$$

,

where $\theta(\alpha)$ denotes the parameter vector found with a given value of α and C^{mm} is the vector of mid-market prices. In practice $||C^{\alpha(\theta)} - C^{mm}||$ is an increasing function of α so the solution can be found by Newton's or dichotomy method with a few low-precision runs of the minimization routine.

6.3 Numerical implementation

The following is the calibration algorithm:

- Choose the prior. We choose our prior by fitting the DEJD model historical daily returns of S&P returns from January 2012 to December 2015.
- Compute the optimal regularization parameter α^* to achieve trade-off between precision and stability using the a posteriori method described above:

  $$\epsilon^2(\alpha^*) = \sum_{i=1}^{N} \omega_i |C_i^{bid} - C_i^{ask}|^2 \approx \epsilon_0^2$$

  The optimal α^* is found by running the gradient descent method (BFGS) several times with low precision.
- Solve variational problem for $\tau(\nu)$ with α^* by gradient-based method (BFGS) with high precision.

  We implemented the above algorithm on Intel Core i7 4770S, 3.1GHz quad-core with hyper-threading and turbo boost to 3.9GHz, and 16GB of DDR3 RAM. The program had been running for almost 20 hours without results at the time of submission.

We give equal weight to each option price, $\omega_i = \frac{1}{N}$ Where N is equal to the number of price observations.

7 Results

Our MLE code gives us the following parameters for the DEJD model; σ is 0.21, λ_u is 0.05, λ_d is 0.09, η_u is 29.99, η_d is 19.37, μ is 0.34 (although this μ is irrelevant for option pricing). The values of λ, the jump arrival rates suggest that an up jump happens on average, every 20 years and a down jump every 10 years. This does not look very realistic but the MLE parameters for diffusion and drift look reasonable.

We price options on S&P 500 index for 2012 using the uncalibrated MLE parameters above. The overall average absolute difference in pricing is \$14. It is \$10 for short term options and \$18 for long term options. Short term options are those with 9-60 days of expiry and those with 60-120 days are long term options are defined in our previous project. Next we look at the pricing difference for calls and puts. The average absolute difference for calls is \$17 which is the same value without taking absolute. We deduce that our DEJD model priced almost all the calls higher than the market. This repeats itself with puts as well with the average value being \$11. The prices implied by the DEJD model hence, in our case shift the entire implied volatility curve upwards. The volatility curve does not pivot or shift down, which were possible cases as reviewed in our previous paper.

The risk free rate is very close to zero throughout the entire period. Assuming that it is zero, we calculate the average profit made by an investor for the entire period. This would be average of $[S(T) - K]^{+} - C(0)$ for calls and $[K - S(T)]^{+} - P(0)$ for puts. The average profit if the investor pays the market price is \$-8 and \$-22 if they pay DEJD prices calculated using uncalibrated MLE parameters. For calls, these values are \$-2 and \$-19 respectively and for puts the values are \$-13 and \$-25. We see that the uncalibrated DEJD performs poorly. The efficiency of the calibrated model can be tested similarly using faster

8 Conclusions

Our conclusion is that the parameter estimates from Maximum Likelihood Estimation are inaccurate but the mean and variance seem relatively realistic. As well, the option prices we computed from these parameters were on average higher than market last prices. We did find that the probability of downward jumps was higher than the probability of up jumps, which indicates skewed returns.

A qualitative conclusion we have come to is that model calibration without complete markets (where multiple stock evolution parameters must be estimated from calibration) is computationally intensive and requires significant thought about implementation. We were required to compute integrals that were difficult to code with commonly available functions. As well we had to make choices about appropriate parameters (α for example) that required significant interpretation of theoretical work.

This has given us an appreciation for the immense amount of thought and effort that has gone into financial software (Numerix, Fincad) that can calibrate options from DJED and other Lévy Processes.

9 Future Work

In future work we would like to implement a faster calibration algorithm than the method outlined above. As well, it may be good to seek out more powerful computers on which to run our complex, large optimizations. This would have given us the opportunity to compare the calibrated parameters to the MLE estimates ex-post-facto for a given time period.

Some other models we would like to estimate parameters for would be Stochastic Volatility and Stochastic Volatility with Jumps (The Bates Model). These models may better fit the data. Many stock return models have been proposed in academic literature. Some include other interesting properties such as time inhomogeneity (a discretely monitored Brownian Motion that models discrete trading), stochastic processes in higher order moments (variance of variance is a stochastic process) and stochastic interest rates.

Calibration of such options would require derivation of much more complicated characteristic functions and risk neutral measures. They might also require more advanced numerical solution techniques (such as Monte Carlo simulation based parameter estimation) to calibrate tractably. The details abstracted by academic papers in implementing and option pricing with these models would provide ample material for further work.

Finally we could measure the returns of a trading strategy where we compute option prices with an alternate model and buy and sell path dependent and other exotic options based off the differences between the market price and the price given by our model.

References

Yacine Aït-Sahalia and Jean Jacod. Testing for Jumps in a Discretely Observed Process. *The Annals of Statistics*, 37(1):184–222, 2009. ISSN 00905364. doi: 10.1214/07-AOS568.

Yacine Aït-Sahalia, Yubo Wang, and Francis Yared. Do option markets correctly price the probabilities of movement of the underlying asset? *Journal of Econometrics*, 102(1):67–110, 2001. ISSN 03044076. doi: 10.1016/S0304-4076(00)00091-9.

Peter Carr and Dilip B Madan. Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2(1990):1–18, 1999.

Rama Cont and Peter Tankov. *Financial Modelling with Jump Processes*. CRC Press LLC, 2004.

Aswath Damodaran. Option Pricing Basics.

Darrell Duffie, Jun Pan, and Kenneth Singleton. Transform Analysis and Asset Pricing for Affine Jump-Diffusions. *Econometrica*, 68(6):1343–1376, 2000.

S. G. Kou. A Jump-Diffusion Model for Option Pricing. *Management Science*, 48(8):1086–1101, 2002. ISSN 0025-1909. doi: 10.1287/mnsc.48.8.1086.166.

Cyrus Ramezani and Yong Zeng. *An Empirical Assessment of the Double Exponential Jump-Diffusion Process*. PhD thesis, 2004.

Peter Tankov and Ekaterina Voltchkova. Jump-diffusion models: a practitioner's guide. *Banque et Marchés*, pages 1–24, 2009.