Mid-Price Movement Prediction using Hawkes Processes
Yuanda Chen Major Advisor: Dr. Alec Kercheval
Financial Mathematics, Florida State University

Contribution
We introduce a 4-dimensional Hawkes processes model to simulate the dynamics of Limit Order Book. Each dimension corresponds to the market events that increases/decrease the queue sizes at the best bid/ask. Comparing to the homogeneous Poisson process, the Hawkes process captures the well-observed clustering effect of each single kind of market event. At the meanwhile, it allows us to build-in the cross-excitation among different kinds of market events. Monte Carlo simulation is used to estimate the probability of the next mid-price movement being upward. An algorithmic trading strategy is designed to evaluate the performance.

M-dimensional Hawkes Process
An M-dimensional Hawkes process with exponential kernel has the conditional intensity:

\[ \lambda_m(t | F(t)) = \mu_m + \sum_{n=1}^{M} \sum_{\alpha=1}^{A_{mn}} \int_0^t e^{-\beta_{\alpha} (t - w)} dN_n(w) \]

Given the occurrence times \( \{ t^n_i \} \) for \( i = 1, 2, \ldots, M \),

\[ \lambda_m(t) = \mu_m + \sum_{n=1}^{M} \sum_{\alpha=1}^{A_{mn}} e^{-\beta_{\alpha} (t - t^n_i)} \]

for \( m = 1, 2, \ldots, M \).

Calibration
The model parameters are calibrated by numerically optimizing the likelihood:

\[ \log L \left( \{ t^n_i \}_{n=1}^{M} \right) = \sum_{m=1}^{M} \log L_m \left( \{ t^n_i \}_{n=1}^{M} \right) \]

\[ = \sum_{m=1}^{M} \left[ -\mu_m T - \sum_{n=1}^{M} \sum_{\alpha=1}^{A_{mn}} \left[ 1 - e^{-\beta_{\alpha} (T - t^n_i)} \right] \right] + \sum_{\{ k | t^k < T \}} \log \left[ \mu_m + \sum_{n=1}^{M} \alpha_{mn} R_{mn}(k) \right] \]

with the recursion:

\[ R_{mn}(i) = \sum_{\{ k | t^k < t^n_i \}} e^{-\beta_{\alpha} (t^n_i - t^k)} - e^{-\beta_{\alpha} (t^n_i - t^n_{i-1})} R_{mn}(i-1) + \sum_{\{ t^k_i | t^k_i < t^n_i \}} e^{-\beta_{\alpha} (t^n_i - t^k_i)} \]

\[ R_{mn}(0) = 0 \]

Simulating Hawkes Process

Hawkes v.s. Poisson

Referencing

Simulating Hawkes Process

Hawkes v.s. Poisson

Diffusion Approximation
Let \( K = \{ \alpha_{mn} \} \), \( \bar{\sigma} = \bar{\sigma}(1 - K)^{-1} \mu T, \sigma = \bar{\sigma}(1 - K)^{-1} \) then \( N(T) \sim \bar{\sigma} T \), \( \mu + \bar{\sigma} W \). The queue sizes at best bid/ask can be modeled as:

\[ \{Q^{\mu}(t) = Q_0 + \sum_{n=1}^{M} N^n_i(t) - M N^n_N(t) \}

Assuming no drift terms in \( Q^{\mu} \) and \( Q^{\sigma} \), we have a 2-dimensional correlated Brownian Motion.

Cython Parallel Computing

The Monte Carlo simulations are very time consuming. All experiments are done using 80-core parallel computing in the Research Computing Center of FSU. By modifying Python code to Cython we obtain about x100 speed increase to get near C-performance. The single-core performance of MATLAB/Python/Cython/C is compared and the scalability of Cython and C with MPI is studied for the parallel computing.

Results

This example shows an event-by-event calculation of the probability that the mid-price moves up. The top pane shows the result obtained in 3 ways: blue and green are from Monte Carlo simulation and red from diffusion approximation. From the volumes at best bid/ask shown in the bottom pane, we can see the mid-price doesn’t change during the whole period and it moves down at the end. The blue curve discovers this about 30 events earlier than the other two.

Cython Parallel Computing

The Monte Carlo simulations are very time consuming. All experiments are done using 80-core parallel computing in the Research Computing Center of FSU. By modifying Python code to Cython we obtain about x100 speed increase to get near C-performance. The single-core performance of MATLAB/Python/Cython/C is compared and the scalability of Cython and C with MPI is studied for the parallel computing.

References