High Frequency Final Project: The Hawkes Process

Carroll Huffine

April 26, 2015

1 Introduction

Information is a strong driver of trading activity. One phenomenon that has been increasingly observed in trading, particularly with the advent of algorithmic trading, is trade clustering. Trades are executed by observing other trades in the market rather than based on an asset's external information. Creating models that can explain or fit these patterns in trading will clearly be advantageous and help high frequency traders increase their profits. Therefore studying processes that can help model these phenomena is an active research topic in high frequency trading. The Hawkes process has received attention with regards to its capabilities of modeling trade clustering and I focus on it in this research paper.

The Hawkes process was first introduced in 1971 by Alan Hawkes. The Hawkes process is similar to a Poisson process in that trades arrive randomly over time and are counted using a rate or intensity which is defined as the number of events occurring over a fixed period of time. However, the Hawkes process is an evolution from the Poisson process in that the Hawkes process is self-exciting so that past occurrences have a positive effect on the probability of seeing future occurrences; a Hawkes process depends on the entire event history. Therefore the occurrence of an event at time t and the history of all previous events stimulate the self-exciting aspect of the process and cause more events to occur. Thus the occurrence of events is an endogenous process which helps model trade clustering in the market. The use of the Hawkes process is widespread in many fields. For example, it is used to model the frequency of earthquakes in seismology since one earthquake is often followed by several others and also in neuroscience to model spiking in neural networks. However, it has only recently gained popularity in financial applications.

This paper seeks to give a broad overview of the Hawkes process and some of its applications to finance. I also inspect one area in which I think the Hawkes process is fruitful and lends itself to further research. There are presently no publicly available packages that can model the Hawkes process effectively so I wanted to create a code that would allow us to apply it in actual trading data. (These results are presented in section 4 where I implement a univariate Hawkes process with a constant mean intensity over an entire window.)

2 Poisson Process

The Poisson process is a well-known and documented phenomenon dating back to the 1800's. It models the random arrival of events measured in number of counts per unit of time (λ). In a Poisson process the probability of event arrival increases with the amount of time since the last arrival. Intuitively, this property is illustrated by the increased probability of the bus arriving the longer one waits for it. The following formula describes the Poisson process:

$$\begin{aligned} P[k] &= \frac{e^{-\lambda\tau}(\lambda\tau)^k}{k!} \\ \text{where } \lambda &= \text{intensity} \\ k &= \text{number of events in time interval} \end{aligned}$$

We also have

$$P\big[N(t+\tau) - N(t))\big] = P[k]$$

where k is the number of events during the time interval between t and $t+\tau$, and $N(t)$ is the Poisson process at time t.

While this accounts for the random arrival of trades, the Poisson process does not model the clustering of events. To do this, I need a process that allows for prior events to trigger further events. Thus I employ the Hawkes process.

3 Hawkes Process

The Hawkes process is a point process that evolves from the simple Poisson process. It has the basic framework of the Poisson process in that its main purpose is to count the number of events (or intensity) in a given period of time. However, the advancement of the Hawkes process stems from the fact that it depends on its entire history of events. The past events are directly accounted for in the formula, and the occurrence of events directly impacts the probability of creating new events in the future. This setup allows for endogenous feedback of events that are entirely uninfluenced by outside news and is evident in trading as trade clustering seems to appear frequently. I will begin by considering the simplest form of the Hawkes process in one dimension with a constant mean intensity.

The change in intensity is modeled as the following[4].

$$d\lambda_t = \beta(\lambda_\infty - \lambda_t)dt + \alpha dN_t$$

Solving for λ yields the following equation.

$$\lambda(t) = \mu + \sum_{t_i<t} \alpha e^{-\beta(t-t_i)}$$

$$\begin{aligned}
\text{where } \lambda &= \text{instantaneous conditional intensity} \\
\mu &= \text{background trade intensity} \\
\alpha &= \text{size of the jump} \\
\beta &= \text{exponential intensity of decay}
\end{aligned}$$

This version of the Hawkes process is analytically tractable. In this formulation, none of the parameters change with time, but some more complex versions of the Hawkes process relax this assumption. The μ term is the trade intensity to which the model reverts. It can be viewed as the baseline intensity and what would be expected in the absence of endogenous trade clustering. μ must be greater than or equal to zero. The α and β terms are used to describe the clustering aspect of the arrivals. The standard is that $\alpha < \beta$ so that trade intensity decays faster than it explodes. However, a paper that will be discussed later argues that as α approaches or becomes larger than β, it can indicate criticality and serve as a predictor for a flash crash[8].

The analytical tractability and interpretation make working with this model appealing. It yields results that are intuitive and easy to calculate. Expected intensity is defined as the average intensity for a time period as is represented by the following.

$$\mathbb{E}[\lambda] = \frac{\mu}{1 - \frac{\alpha}{\beta}}$$

Another value of key interest is the branching ratio. This metric describes the endogenous feedback component of the system, and it can be interpreted as the fraction of trades generated endogenously. The branching ratio is given by the following formula:

$$n = \frac{\alpha}{\beta}$$

Another metric used is the expected number of events triggered by one jump at t=0 [4]. The following formula describes this idea:

$$N_{\text{response}} = \frac{\alpha}{\beta - \alpha}$$

The final metric is the half-life [6]. This describes the half-life of the decay for an excited jump and is shown in this formula:

$$t_{\frac{1}{2}} = \frac{\ln(2)}{\beta}$$

This version of the Hawkes process can be fit using numerical optimization via maximum likelihood estimation (MLE). The log-likelihood function for a Hawkes process is given by the following formulas [1].

$$\ln\mathcal{L}(\{t_i\}_{i=1\ldots n}) = -t_n\mu + \frac{\alpha}{\beta}\sum_i^n (e^{-\beta(t_n-t_i)} - 1) + \sum_{i<j}(\ln\mu + \alpha e^{\beta(t_j-t_i)})$$

These can be recursively calculated using the form,

$$\ln\mathcal{L}(\{t_i\}_{i=1\ldots n}) = -t_n\mu + \frac{\alpha}{\beta}\sum_i^n (e^{-\beta(t_n-t_i)} - 1) + \sum(\ln\mu + \alpha\mathrm{R}(i))$$

$$\mathrm{R}(i) = e^{\beta(t_i-t_{i-1})}(1 + R(i-1))$$

$$\mathrm{R}(1) = 0$$

The Hawkes process can be extended to contain background mean trade intensity that varies with time ($\mu(t)$). However, [8] claims that the most simple version is actually the most effective in real-life trading since it does not overfit the data. In addition, the Hawkes process can be modeled in a multivariate setting and the multivariate Hawkes process is discussed in Section 5.

4 Finance Applications

Understanding trade clustering is of great value. However, the use of the Hawkes process is still in its infancy in finance and can potentially cover a large scope of topics. I outline some of the areas that I would like to research further in applying the Hawkes process.

4.1 Algorithmic Speed

One area that is worth investigating is the speed of fitting the Hawkes process. By definition, the process is dependent on all previous time points therefore the algorithm can get bulky and computationally intensive. A package in the R CRAN library (http://www.biostat.jhsph.edu / rpeng/software/index.html) is able to fit the Hawkes process. However, the function may take up to several hours to run depending on the size of data used. Furthermore, the package is incompatible with any version of R newer than version 2.15. Clearly, it is inefficient and not realistic to work with a function that takes hours to model trade clustering. Code was generated fits the Hawkes process in real-time (i.e. seconds) so it can conceivably be used in trading. The code is on OneTick data where I pulled trade data for Apple (AAPL) on December 11, 2014. In this implementation of the Hawkes process on the OneTick data, I was agnostic to whether the trade was a buy or sell and only considered the execution of the trade. The empirical intensity is defined as the as number of executed trades within a given time interval, and the trades are aggregated into 1-minute buckets (#trades/minute). I compare the intensity of the empirical data versus the Hawkes process in Figure 1 and tabulate the key statistics such as the Akaike Information Criterion and the Root Mean Squared Error in Table 1. Overall, my results show that AAPL does not exhibit clustering properties and almost 90% (1-branching ratio) of its trades are generated by exogenous news during the window.

	μ	α	β	$\mathbb{E}(\lambda)$	Branching Ratio	N_{resp}	RMSE	AIC
AAPL	1.001E-9	0.100004	1	1.11E-9	.100004	0.1111	75.4787	-304.1914

Table 1: This table depicts the parameters estimated by MLE for ta 8,000 data point window for AAPL .

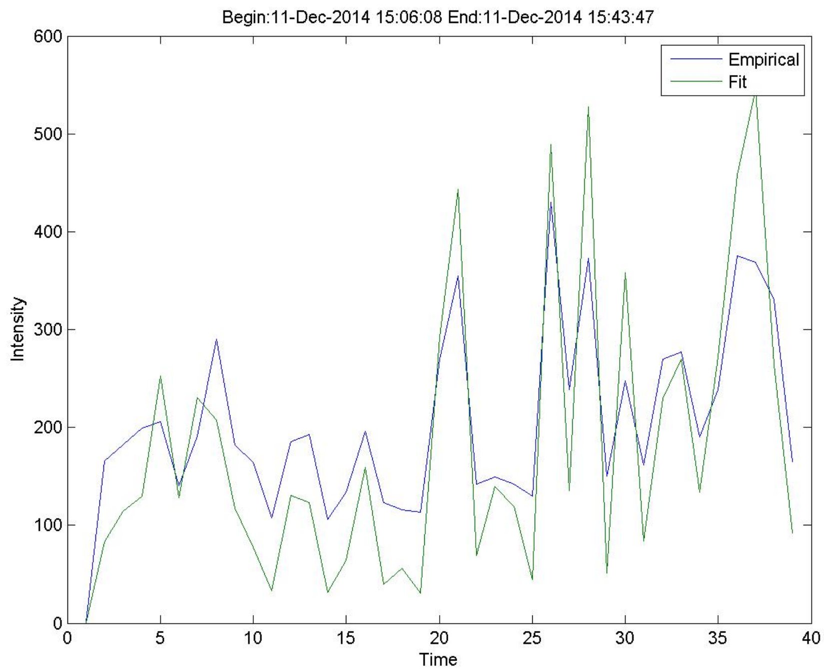

Figure 1: Empirical versus fit intensity of AAPL for 8,000 datapoint window (approximately 40 minutes of trading).

Another way to increase the speed of the procedure of fitting the Hawkes process involves using a different method for estimating its parameters, Generalized Method of Moments (GMM). In the paper by Fonseca and Zaatour[4], the authors claim that the Hawkes process can be solved “instantaneously” using GMM, but the results are not as accurate as MLE. On the other hand, implementing GMM on a low-level programming language would allow for immediate information to be used in high frequency trading. The authors claim that finding the moments for the one-dimensional case is feasible, but the multivariate solution is tedious to implement.

The parameters of the Hawkes process using GMM are calibrated by finding the moments of the difference in jump size $N_{t+\tau} - N_t$ and the autocorrelation structure of the Hawkes process. Deriving the results takes a few pages of calibration, but the moments are presented as the following,

$$\mathbb{E}[N_{t+\tau} - N_t] = \frac{\lambda_\infty \beta \tau}{\alpha - \beta} + e^{t(\alpha-\beta)} \frac{(-\lambda_\infty \beta + e^{(\alpha-\beta)\tau}\lambda_\infty \beta - \alpha\lambda_0 + e^{(\alpha-\beta)\tau}\alpha\lambda_0 - e^{(\alpha-\beta)\tau}\beta\lambda_0)}{(\alpha-\beta)^2}$$

When taking the first moment to $t \to \infty$ for a stationary regime, I find the following,

$$\mathbb{E}[N_{t+\tau} - N_t] = \frac{\lambda_\infty}{1 - \alpha/\beta} \times \tau$$

The paper then defines $\Lambda = \frac{\lambda_\infty}{1-\alpha/\beta}$. Then the second moment is derived to take the following form when considering the long-term stationary form with $t \to \infty$,

$$\begin{aligned}\mathbb{E}[(N_{t+\tau} - N_t)^2] &= \tau\Lambda \\ &+ 2\beta\lambda_\infty\Lambda \times \Bigg[-\frac{\tau^2}{2(\alpha-\beta)} - (\alpha-\beta)^2\tau + (\alpha-\beta)^{-3}(e^{(\alpha-\beta)\tau} - 1) \Bigg] \\ &- \Bigg[-\frac{\tau^2}{(\alpha-\beta)} + (\alpha-\beta)^{-2}(e^{(\alpha-\beta)\tau} - 1) \Bigg]\end{aligned}$$

furthermore, the third moment with $t \to \infty$ is shown to be,

$$\begin{aligned}\mathbb{E}[(N_{t+\tau} - N_t)^3] &= \frac{1}{2(\alpha-\beta)^6}\lambda_{infty}\beta \\ &\Bigg[-e^{2(\alpha-\beta)\tau}\alpha^2(2\alpha - 3\beta)(\alpha-\beta) \\ &2e^{(\alpha-\beta)\tau}\alpha(\alpha^3 - 4\alpha^2\beta + 6\beta^3 + 3(\lambda_\infty + \alpha)(\alpha - 2\beta)(\alpha-\beta)\beta\tau) \\ &+ \beta\Bigg(3\alpha(\alpha^2 - \alpha\beta - 4\beta^2) \\ &+ 2(\alpha+\beta)(3\lambda_\infty\alpha(\alpha - 2\beta) + \beta^2(2\alpha+\beta))\tau \\ &+ 6\lambda_\infty(\alpha-\beta)^2\beta^2\tau^2 + 2\lambda_\infty^2\beta(\alpha+\beta)^3\tau^3 \Bigg)\Bigg]\end{aligned}$$

Finally, the paper defines the autocorrelation structure which will be left out of this report other than acknowledging it exists. The parameter estimation is given as,

$$\hat{\theta} = \text{argmin}\Bigg\{ (M - f(\theta))^\top W (M - f(\theta)) \Bigg\}$$

Where M are the empirically estimated moments and $f(\theta)$ are the theoretical moments. Clearly even for the one-dimensional case, these equations are complex. However, it is entirely tractable and possible to implement in code. Additionally, since autocorrelation is applied in the calibration, it is possible to investigate forecasting which will be examined in section 4.3.

4.2 Branching Ratio

One main goal in modeling market data is prediction and correlation since this will help us develop trading strategies. The two papers by [8] and [6] claim that when fitting a Hawkes process, the branching ratio is key in predicting movements and important events in the stock markets.

In "Analysis of Order Clustering Using High Frequency Data: A Point Process Approach," [6] shows that there is correlation between intra-day high frequency trading and the intra-day dynamics of the branching ratio. In this paper, the authors are interested in measuring quotes per trade with the branching ratio. Their results are inconclusive since it was difficult for them to find a good proxy for high frequency trading data because the data set they chose to use did not have enough observations to suit their research needs. However, they claim that their research is a starting point for this topic since they were able to show that positive correlation exists between the branching ratio and high frequency trading and further testing should be continued in the future.

Another strategy addressing the Branching Ratio is in the paper by Filimonov and Sornette [8]. In this paper, the authors claim that endogenous trading information associated with the S&P 500 could have predicted flash crashes in the stock market. The authors state that the branching ratio behaves like the "criticality" of a nuclear reactor and assert that this occurs when the branching ratio approaches 1. Furthermore, the authors believe that approaching this point can predict a crisis.

To investigate, they backtest E-mini futures. The authors show the change in the branching ratio over time, for example: from 1998-2000 the branching ratio is 0.3, from 2000-2002, it is elevated to 0.6 and by 2006, just before the crisis, it reached 0.7 and approaching 0.8 in late 2006. The authors interpret these results as showing that 70-80% of trades are generated by endogenous causes rather than new exogenous information. They also claim that the flash crash in 2010 was effectively predicted by the Hawkes process as evidenced by their calibration which shows that the branching ratio increased from 72% to 95% so that all but 5% of trading was triggered endogenously. Statistically significant increases in branching ratio can help predict these crashes. From my perspective though, I would argue that correlation does not prove causation and that the flash crash could have possibly caused the influx of trading.

4.3 Forecasting

Another key outcome of implementing the Hawkes process is to develop a trading strategy based on forecasting. This could be accomplished by discovering some autoregressive structure within the trade arrivals, and since a Hawkes process includes all previous data in the estimation at each time step, it would be the perfect candidate. Source [4] makes the claim that this is very possible and defines the following AR(1) formula to describe a Hawkes process.

$$y_t = a_1 y_{t-1} + \epsilon_t$$

Here, y_t is defined as $y_t = \frac{1}{\tau}(N_{t_i} - N_{t_{i-1}})$ and $\tau = t_i - t_{i-1}$. The AR parameters for the Hawkes process as the following.

$$\mathbb{E}_{t-1}[y_t^2] = a_1 \mathbb{E}_{t-1}[y_t y_{t-1}] + \mathbb{E}_{t-1}[y_1 u_t]$$

it follows that

$$a_1 = \frac{\mathrm{Cov}(\tau, 0)}{V(\tau)} = \mathrm{Acf}(\tau, 0)$$

The same reasoning can be extended to an arbitrary AR(p) process, but for brevity, these equations will be left to the reader if desired. They can be found in [4] on page 19.

Extending the trade arrivals and clustering to an AR framework would be ideal for developing trading strategies. Generating algorithms (or proving this is not a valid strategy) should be the next logical extension of the research performed in this paper. One other area is splitting bid and ask arrivals and finding the clustering of each separately. There is information in the market not currently being utilized, and it could generate positive returns since traders are not capturing this information for profit.

5 Multivariate Hawkes Process

The univariate Hawkes process can also be extended to a multivariate version. If I let $c_1, c_2, ..., c_n$ define an event cascade, so that c_1 triggers the event c_2 which then triggers the event c_3 and so forth, then the general Hawkes process is a point process that is described by its intensity, λ at a given dimension and point in time:

$$\lambda_s(t) = \mu_s + \sum_{s=1}^{s=M} \sum_{t<t'} \alpha_{s',s} \kappa(t' - t)$$

As in the univariate case, μ_s represents the background rate of intensity, but in the multivariate case it is the trade intensity in dimension s. M is the number of states, $\alpha_{s',s}$ denotes the influence of state s' on state s and κ is a time-decaying kernel which in my case is modeled by the function $e^{-\beta_s(t'-t)}$ where β controls the magnitude of the decay. As the time from the occurrence of the event increases, this function tends towards 0. In contrast to the univariate case, $\alpha_{s',s}$ is a matrix showing how the events in dimension s' affect events in dimension s. More formally, I may refer to $\alpha_{s',s}$ as the branching matrix and interpret the coefficients of this matrix as the mean expected number of descendants of a given point event [12]. So, for example if I have a bivariate Hawkes process with branching matrix α given by:

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

then $a_{1,2}$ tells us the expected number of event 1 which results from event 2. Also, in the case where non-diagonal entries are $0s$ the branching matrix reduces to the univariate Hawkes process where the process is self-exciting only.

5.1 Estimating parameters

Similar to the univariate case, estimates of the intensity and branching are found using MLE, so that for a sequence of event cascades over time, the maximization is set up as:

$$\mathcal{L} = \sum_{i=1}^{N} log\lambda_{m_n}(t_n) - \sum_{m=1}^{M} \int_0^T \lambda_m ds$$

(by the formulation in Li and Zha [14]) here M denotes the number of dimensions, t_n is the nth event in a cascade, m_n shows the dimension where the nth event occurred. However, as the number of dimensions increases in a problem, the number of parameters that need to be estimated also increases by a factor of $O(M^2)$ where M is the original number of dimensions. This means that estimating the parameters through MLE becomes more computationally challenging and unclear. Furthermore, estimation of parameters may be increasingly complicated as the length of the rolling window over which they are estimated increases. This is due once again to limitations in computing power when estimating the maximum likelihood coefficients [10].

5.2 Applications Towards Finance

The multidimensional Hawkes process easily lends itself to the dynamics of high frequency trading and to modeling market, limit and cancellation orders at different price levels and sides of the order book [10]. I can adapt these specifications to the multivariate Hawkes process by allowing M, which I previously defined generically as "states", to denote the number of order types (i.e. market or limit orders) which has direct applications towards modeling the interactions between buy, sell and cancellation orders as well as their effects on prices and order frequencies.

As another example of a financial application, Bacry and Muzy [15] use a four-dimensional Hawkes process to model the impact of market order arrivals on prices by taking into account market microstructure elements such as random movements in prices, high frequency trade mean-reversion, discrete price grids and correlations at different times [15]. Researchers have also been successful in extending the multivariate Hawkes model and adapting it to capture novel market elements. For example, Fauth and Tudor hypothesize that larger volume of trades have a bigger impact on trades than smaller volumes since larger volumes have the ability to trigger other trades thus "exciting the market" [11]. The authors therefore extend the multivariate Hawkes process to include a multiplicative marked intensity term, g in their modeling of the impact of trade volume on intensity and prices:

$$\lambda_s(t) = \mu_s + \sum_{s=1}^{s=M} \sum_{t<t'} \alpha_{s',s} \kappa(t - t') g_s(v)$$

Here, v is the mark which is defined as a value assigned to each point in order to add more information and models the volume at each point [11]. The function g captures the effect of volume on intensity so that "the intensity increases not only with respect to the arrival time of events but also with respect to the mark value." Fauth and Bacry apply their model to foreign exchange data and find that their model is consistent with empirical observations and conclude by suggesting further research using the multivariate Hawkes process with applications in risk management and Value at Risk. Thus, overall it appears that the multivariate Hawkes process can become a useful tool in analyzing and modeling high frequency data.

6 Conclusion

In conclusion, this report has discussed some valuable aspects of applying the Hawkes process to finance. Though this application is still in its infancy, the results appear quite promising. The one-dimensional case is shown to be a powerful and also analytically tractable method that can help a trader gain additional information about the market. I also outline the areas which would benefit most from the application of a Hawkes process: faster calibration, use of the branching ratio, and forecasting. In addition, I expanded the one-dimensional setting into a multivariate scope which allows us to model many co-existing pieces of a high frequency trader's information set. This includes the simultaneous modeling of both bids and asks while also considering market orders as well as limit orders. The results indicate that if applied correctly, the Hawkes process can yield valuable information. Since the application of the Hawkes process to finance is still in its infancy, many of the resources are not of the highest quality, and it is sometime difficult to interpret the validity of the presented outcome. However, ultimately the Hawkes process is a rapidly developing concept that could prove to be a useful tool for high frequency traders.

A Main Function

```
%-------------------------------------
% Fit a Hawkes Process using MLE
% Trey, Katherine, Philippine, Leonore
%-------------------------------------

close all, clear all, clc;
warning off;
tic %used for timing the algorith

%% Declare User Input
numTrades = 8000; % sets the window length
saveImages = true; % set to true to save an image each iteration
file = dir('*.xlsx');
all_trades = importdata(file.name);
times = all_trades.data(:,4); % should be set to the column containing
    unix timestamps

%% Set variables
T = numel(times);
% Chop of times that don't match with size of trade window
chopOffTimes = rem(T,numTrades);
T = T-chopOffTimes;
times = times(chopOffTimes+1:end);
totIterations = T/numTrades; % must be an integer

% Zero out matrices
AIC = zeros(totIterations,1);
BIC = zeros(totIterations,1);
fullRMSE = zeros(totIterations,1);
mu =zeros(totIterations,1);
alpha = zeros(totIterations,1);
beta = zeros(totIterations,1);
expectedIntensity = zeros(totIterations,1);
branchingRatio = zeros(totIterations,1);
Nresponse = zeros(totIterations,1);
halfLife = zeros(totIterations,1);

%% Begin Iteration data points
beginTimes = 1;
endTimes = numTrades;
for iterationTrack = 1:totIterations
%% Initialize current loop varaibles
set = 1:numTrades;
timesNow = times(beginTimes:endTimes);
beginTimes = beginTimes + numTrades;
endTimes = endTimes + numTrades;

%% Introduce noise to matching time stamps
[~,uniqueStamps] = unique(timesNow);
```

```matlab
idxNonunique = ~ismember(set,uniqueStamps);
for i = 1:numTrades
    if idxNonunique(i)
        timesNow(i) = timesNow(i) + rand;
    end
end
timesNow = sort(timesNow);

%% Call the minimization function
% set initial guess and reference function
mu0 = .1; % making mu >= or beta seems to work the best
alpha0 = .1; % keeping this about 1/10 of beta seems the best
beta0 = 1;
parameters = [mu0; alpha0; beta0];
func = @(parameters) HawkesMLE(parameters,timesNow);

%[fitParameters,logLikelihood,EXITFLAG] = fmincon(func,parameters,[-1 0
    0],0,[],[],[],[],[],optimset('MaxFunEvals',100000,'TolFun',1e-8,'
    TolX',1e-8));
[fitParameters,logLikelihood,EXITFLAG] = fminunc(func,parameters,
    optimset('MaxFunEvals',100000,'TolFun',1e-8,'TolX',1e-8));

mu(iterationTrack) = fitParameters(1);
alpha(iterationTrack) = fitParameters(2);
beta(iterationTrack) = fitParameters(3);

% fmincon/fminunc appears to work the best

% Calculate the fitted conditional intensities
tempIntensity = zeros(numTrades,1);
for t = 2:numTrades
    tempIntensity(t) = exp(-beta(iterationTrack) * (timesNow(t) -
        timesNow(t-1)))*(1+tempIntensity(t-1));
end

% Calculate results
conditionalIntensity = mu(iterationTrack) + alpha(iterationTrack)*
    tempIntensity;
expectedIntensity(iterationTrack,1) = mu(iterationTrack)/(1-alpha(
    iterationTrack)/beta(iterationTrack));
branchingRatio(iterationTrack,1) = alpha(iterationTrack)/beta(
    iterationTrack);
Nresponse(iterationTrack,1) = alpha(iterationTrack)/(beta(
    iterationTrack)-alpha(iterationTrack));
halfLife(iterationTrack,1) = log(2)./ beta(iterationTrack);

%% Bin the data
% Add one for each empirical count
% Sum the conditional intensities within each window
empiricalBins = [];
fitBins = [];
```

```
binTimes = [];
binTimes(1) = timesNow(1);
timeTrack = timesNow(1);
empiricalBins(1) = 0;
fitBins(1) = 0;
n = 1;
for t = 1:numTrades
   if timesNow(t) < timeTrack
       empiricalBins(n) = empiricalBins(n) + 1;
       fitBins(n) = fitBins(n) + conditionalIntensity(t);
   else
       while timesNow(t) > timeTrack
       n = n + 1;
       timeTrack = timeTrack + 60;
       empiricalBins(n) = 0;
       binTimes(n) = timeTrack;
       fitBins(n) = 0;
       end
       empiricalBins(n) = 1;
       fitBins(n) = fitBins(n) + conditionalIntensity(t);
   end
end
numBins = 1:numel(empiricalBins);

%% Results
timeBegin{iterationTrack,1} = datestr(datenum([1970 1 1 0 0 timesNow(1)
   ]));
timeEnd{iterationTrack,1} = datestr(datenum([1970 1 1 0 0 timesNow(end)
   ]));
dateVector = [];
dateVector = zeros(numel(binTimes),1);
for i = 1:numel(binTimes)
    dateVector(i) = datenum([1970 1 1 0 0 binTimes(i)]);
end

setTitle = strcat('Begin:', timeBegin{iterationTrack},' End:', timeEnd{
   iterationTrack});

img(iterationTrack) = figure(iterationTrack);
%plot(dateVector,empiricalBins,dateVector,fitBins); % using actual
   dates caused distorition of the axes
plot(1:numel(dateVector),empiricalBins,1:numel(dateVector),fitBins)
title(setTitle)
xlabel('Time')
ylabel('Intensity')
legend('Empirical','Fit')
%datetick

[thisRMSE,thisAIC,thisBIC] = hawkesFitStatistics(empiricalBins,fitBins,
   logLikelihood);
```

```
fullRMSE(iterationTrack,1) = thisRMSE;
AIC(iterationTrack,1) = thisAIC;
BIC(iterationTrack,1) = thisBIC;

%% Save plot to image
if saveImages
    [~, finalMonth] = month(timeEnd);
    finalYear = year(timeEnd(iterationTrack));
    finalYear = num2str(finalYear);
    iter = num2str(iterationTrack);
    imageName = strcat('Image',iter,'_',finalMonth(iterationTrack,:),
        finalYear,'.jpg');
    saveas(img(iterationTrack),imageName);
end
close all;
end

%% Output results
T = table(timeBegin,timeEnd,fullRMSE,AIC,BIC,mu,alpha,beta,
    expectedIntensity,branchingRatio,Nresponse,halfLife);
writetable(T,'Hawkes Output.csv');
figure
plot(fullRMSE);
title('RMSE Plot')
xlabel('Data Point')
ylabel('RMSE')

algoTime = toc %algoTime is the time in seconds for the entire script
```

B MLE Function

```
% Function to fit the Hawkes parameters with MLE

function logLikelihood = HawkesMLE(parameters,times)
% Function used to in optmization to fit the Hawkes parameters.  Use
    the
% negative of the minimization to find the maximum.
% Input:
%   parameters - The intial guess for mu, alpha, and beta being fitted
%   times - A list of trade arrival times
% Output:
%   logLikelihood - LL function used for fitting

% Equations:
% LL = -T(end)*mu + alpha/beta * sum(exp(-beta*(t(end)-ti)-1) + ...
%       sum(log(mu+alpha*R(i))
% R(i) = exp(-beta*(t_(i) - t_(i-1))) * (1+R(i-1))
% R(1) = 0
% firstSum = alpha/beta * sum(exp(-beta*(t(end)-ti)-1)
% secondSum = sum(log(mu+alpha*R(i))

mu = parameters(1);
alpha = parameters(2);
beta = parameters(3);
T = size(times,1);

% find firstSum
timeDifference = times(T) - times;
timeExponential = exp(-beta*timeDifference)-1;
firstSum = alpha/beta * sum(timeExponential);

% find secondSum
R = zeros(T,1);
for timeCounter = 2:T
    R(timeCounter) = exp(-beta * (times(timeCounter) - ...
        times(timeCounter-1)))*(1+R(timeCounter-1));
end
secondSum = sum( log(mu + alpha*R) );

logLikelihood = -(-mu*times(T) + firstSum + secondSum);
```

References

[1] Stephen Crowley, 2013 Jan 30, *Point Process Models for Multivariate High-Frequency Irregularly Spaced Data*.

[2] Angelos Dassios and Hongbio Zhao, 2013 July 10, *Exact simulation of Hawkes process with exponentially decaying intensity*. "Electronic Communications in Probability" Vol 18, no.62, p. 1-13.

[3] Ione Muni Toke, 2011 Feb 4, *An Introduction to Hawkes Processes with Applications to Finance*. BNP Paribas Chair of Quantitative Finance.

[4] Jose Da Fonseca and Riadh Zaatour, 2013 Aug 4, *Hawkes Process : Fast Calibration, Application to Trade Clustering and Diffusive Limit*. Auckland University of Technology. Grande Voie des Vignes.

[5] Thomas Liniger, 2009, *Multivariate Hawkes Processes*. Swiss Federal Institute of Technology Zurich.

[6] F. Lorenzen, 2012 Aug, *Analysis of Order Clustering Using High Frequency Data: A Point Process Approach*. Tilburg School of Economics and Management.

[7] Alvaro Cartea, Sebastian Jaimungal, Jason Ricci, 2014 April15, *Buy Low Sell High: a High Frequency Trading Perspective*. Forthcoming in SIAM Journal of Financial Mathematics.

[8] Vladimir Filimonov and Didier Sornette, 2012 April 18, *Quantifying reflexivity in financial markets: towards a prediction of flash crashes*. University of Geneva.

[9] Chistopher Stover, *Hawkes Process*. From MathWorld – A Wolfram Web Resource, created by Eric W. Weisstein. "http://mathworld.wolfram.com/HawkesProcess.html".

[10] Rajat Goyal, Mayank Gupta, Sahand Sasha Karimi, March 2014, *Analysing Order Flow Dependence in Markov Chain Limit Order Book Model*. Unpublished: 2014 MFE AFP Finalists

[11] Alexis Fauth and Ciprian A. Tudor, November 20, 2012 *Modeling First Line Of An Order Book With Multivariate Marked Point Processes*. Sorbonne University

[12] Paul Embrechts, Thomas Liniger, Lu Lin, 7 March, 2011, *Multivariate Hawkes Processes: An Application to Financial Data*. Applied Probability Trust

[13] Ioane Muni Toke and Fabrizio Pomponio, June 14, 2012, *Modelling Trades-Through in a Limit Order Book Using Hawkes Processes* Economics E-journal, Volume 6, 2012-22

[14] Liangda Li and Hongyuan Zha, 2014 *Learning Parametric Models for Social Infectivity in Multi-Dimensional Hawkes Processes* Proceedings of the Twenty-Eight AAAI Conference on Artificial Intelligence

[15] Emmanuel Bacry and Jean-Francois Muzy, 2014 *Hawkes Model for Price and Trades High-Frequency Dynamics* Quantitative Finance, Vol. 14, Issue 7