Preliminary Project Update

Nicholas Head

January 11, 2015

Problem Background

A long-standing problem in the field of market microstructure relates to the information asymmetry that occurs between two parties wishing to buy or sell financial assets. The issue arises when one trader may potentially be in the possession of privileged information about the underlying asset, while the counterparty may only have access to public news and may end up buying an asset with a lower expected value than the efficient price would suggest. This scenario is otherwise known as insider-trading.

The research to date has dealt with estimating a value called the Probability of Information-based Trading (PIN). This is a crucial area of research particularly for market makers, as they need to set a bid-ask spread (their profit) that will take into account the probability of them buying an asset with a lower expected value. A higher spread is designed to compensate the market maker for this eventuality.

Traditional models for PIN are based off measuring the number of buy and sell trades taking place in the market over a period of time, and using this as a proxy to estimate PIN parameters. The trade direction (buy or sell) is not generally made available in market data and therefore must be considered a latent variable also to be estimated. The mechanism for this is typically done by the matching of trade and quote data.

PIN itself is typically modelled with the sequential trade model first put forward by Easley and O'Hara [1]. The model assumes the presence of both informed and uninformed traders, along with an equally uninformed market maker. In this setup information events occur with probability α, and the news is either good news (with probability $1 - \delta$) or bad news (with probability δ). The market maker's job is to set prices and execute orders as they arrive. The informed traders will buy a stock for which the news is good and will sell otherwise, on the same day the information event occurs. The rate of informed trading is μ and the rates of uninformed buy and sell orders are ϵ_b and ϵ_s respectively. This model is illustrated in Figure 1.

[1] http://www.fsa.ulaval.ca/personnel/PHGRE5/files/Easley_OHara_1987.pdf

The model further assumes the number of buy and sell trades to be independent of one another, to follow Poisson processes for a particular trading day and to be independent across trading days. This leads to a likelihood function with a mixture of Poisson processes

that can then be used to obtain the PIN as the fraction of informed trades to the total number of trades:

$$PIN = \frac{\alpha\mu}{\alpha\mu + \epsilon_b + \epsilon_s}$$

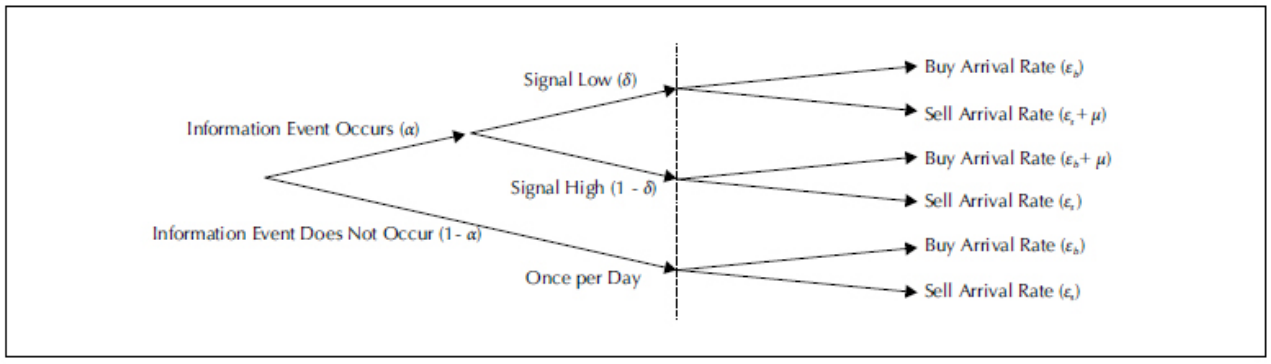

Figure 1 *Tree Diagram of the Trading Process*

Source: Adapted from Easley, Hvidkjaer, and O'Hara (2002).

There have been a variety of criticisms of this model (for instance by Benos and Jochec[2]) that have speculated that the model does not intuitively conform to the theoretical economic realities. For instance one way of testing the efficacy of the PIN statistic is by observing a relationship between it and the bid-ask spread (at times when the PIN is higher, this should lead to a higher spread as market makers attempt to compensate for the increased risk of trading with informed traders). The issue with this approach however is that the positive relationship could actually be caused by something else other than PIN e.g. inventory effects. Furthermore the assumptions related to the information events are unrealistic in the respect that the news events are not necessarily independent across days, nor is the one-event-per-day assumption realistic.

[2] http://business.illinois.edu/finance/phd/pdf/5299.pdf

Perhaps more importantly is that the PIN model does not take trade volume into account. It is conceivable that when a trader has inside information, the are more likely to increase the size of their trades to take more profit from their information. The traditional PIN model doesn't take this into account at all.

Many extensions to the basic PIN model have been proposed over the past 25 years to account for all of its perceived shortcomings. Most recent of which is by the original authors themselves who propose a model called Volume-Synchorised PIN (VPIN). The authors claim that this indicator would have predicted the May 2010 Flash Crash[3].

[3] http://insight.kellogg.northwestern.edu/article/the_trouble_with_vpi

A further extension that I would like to investigate is to see whether the model can be reworked to cater for intraday data.

Statistical Techniques

Most of the parameter estimation I have seen in the literature relies on maximum likelihood . After I have reproduced the results from some of the more seminal work, my intention is to re-implement the models under the Bayesian framework using MCMC. My reasoning for this is simply that the market-makers are inherently Bayesian in the first place. They propose a model for the likelihood (e.g. poisson mixture) and some prior distribution for parameters to estimate. After observing some data (e.g. trading activity in the market) they use the old posterior as a new prior when predicting new values. This them leads them to develop better and better values for their bid-ask spreads taking into account the PIN given the newly available data.

Data

I have managed to successfully get access to NYSE Trade and Quote (TAQ) data from University of Pennsylvania, Wharton Research Data Services. This is a comprehensive historical database of all equities traded not just on the NYSE but other US exchanges such as Nasdaq and Amex.

The datasets consists of both trade and quote data in the following formats:

	SYMBOL	EX	PRICE	SIZE
2014-03-04 09:30:01.427	HZO	N	14.2500	915
2014-03-04 09:30:01.430	HZO	N	14.2500	1116
2014-03-04 09:31:23.746	HZO	D	14.3920	200
2014-03-04 09:31:23.746	HZO	D	14.3960	100
2014-03-04 09:31:23.746	HZO	D	14.3980	100
2014-03-04 09:31:23.763	HZO	D	14.4000	100

Table 1: Trade Data Sample for Symbol HZO

	SYMBOL	EX	BID	BIDSIZ	OFR	OFRSIZ
2014-03-04 04:00:00.266	HZO	P	0.00	0	0.00	0
2014-03-04 07:39:12.963	HZO	P	0.00	0	18.29	18
2014-03-04 07:39:12.969	HZO	P	0.00	0	16.99	10
2014-03-04 07:39:12.971	HZO	P	10.26	18	16.99	10
2014-03-04 07:39:12.973	HZO	T	10.77	1	0.00	0
2014-03-04 07:39:12.974	HZO	P	11.52	18	16.99	10

Table 2: Quote Data Sample for Symbol HZO

The resolution I have attained access to goes down to millisecond which is both a benefit and a boon. The high frequency data provides

an unprecendented view on the microstructure dynamics however also means the shear volume of the data may prove to be computationally very challenging to transmit, store and calculate. As I will outline below, I have made some strides in this area already.

Work so far

Once I managed to get access to the TAQ data, the next step was getting it into a format where it would be amenable to processing and analysis. I managed to find an R package from Corenlissen et al called 'highfrequency'[4] which is specifically designed for that task. One stumbling block I did find however was that the package wasn't set up to handle the millisecond resolution that is now available in TAQ data. For this reason I have had to make several modifications to the highfrquency package in order to make it work. Additionally there were other logic changes such as column ordering and naming that needed to be updated.

[4] http://highfrequency.herokuapp.com/index.html

Once the data had been loaded successfully the next task was running some trade cleanup routines to remove zero prices, limit the observations to a single exchange, filtering by sale condition, and merging any records with the same timestamp

```
tdata = tradesCleanup(tdataraw = tdata, exchanges = "D",
    report = FALSE)
```

This was then followed by quote cleanup which was responsible for removing any errant values with abnormally large spreads and outliers where the mid-quote deviated by more than 25 median absolute deviations from a rolling centred median.

```
qdata = quotesCleanup(qdataraw = qdata, exchanges = "T",
    report = FALSE, maxi = 25)
tdataAfterFinalCleanup = tradesCleanupFinal(qdata = qdata,
    tdata = tdata)
```

Once the trade and quote data was cleaned up the trade and quote data must be matched up using the technique as outlined in Vergote 2005[5]

[5] http://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=1503965

```
tqdata = matchTradesQuotes(tdataAfterFinalCleanup,
    qdata)
```

At this point we are finally ready to plot the data and conduct some initial analysis:

```
plot.xts(tqdata[, "PRICE"], type = "bars", main = "Trades and Spreads for HZO")
lines(tqdata[, "OFR"], col = "red")
lines(tqdata[, "BID"], col = "green")
```

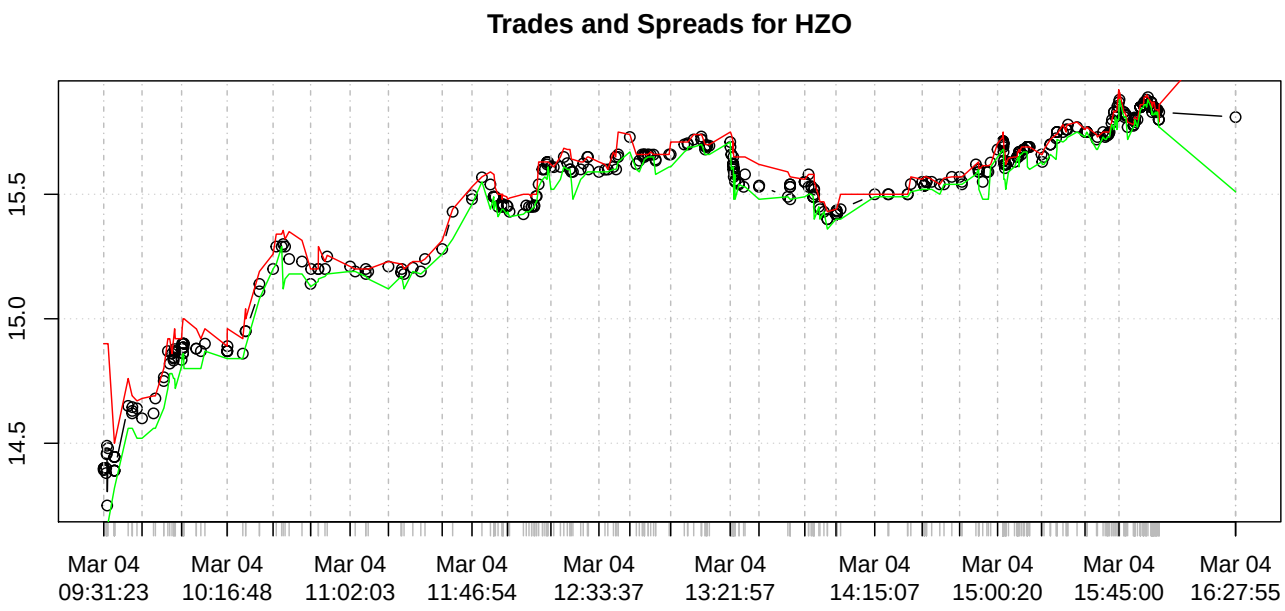

Figure 1: Trades and Spreads for HZO - Cleansed

The highfrequency package also provides some liquidity measures which will become necessary later on

The inferred trade direction (D_t) is a vector which has values 1 or (-1) if the inferred trade direction is buy or sell respectively. This is based off the Lee and Ready 1991 approach[6].

[6] http://www.acsu.buffalo.edu/~keechung/MGF743/Readings/Inferring%20trade%20direction%20from%20intraday%20data.pdf

```
tradeDirection = getTradeDirection(tqdata)
```

The effective spread can also be calculated as:

$$\text{effective spread}_t = 2 * D_t * (\text{PRICE}_t - \frac{(\text{BID}_t + \text{OFR}_t)}{2}),$$

```
effectiveSpread = tqLiquidity(tqdata, type = "es")
```

At this point we are ready to set up the inputs to our likelihood calculation, specifically we need the number of buy and sell orders

```
tradeDirection = matrix(tradeDirection)
buy_side = which(tradeDirection > 0)
num_buy_side = length(buy_side)
num_sell_side = length(tradeDirection) - num_buy_side
ntrades = cbind(num_buy_side, num_sell_side)
```

And finally we can run our MLE procedure using the likelihood function provided by the R 'PIN' package[7]

[7] http://journal.r-project.org/archive/2013-1/zagaglia.pdf

```
devtools::install_github("cran/PIN")
init_params = cbind(0.15, 0.05, 0.5, 0.5)
```

```
param_optim = optim(init_params, PIN::pin_likelihood,
    gr = NULL, ntrades)
```

Which gives us our model's parameter estimates and hence our PIN value

$$PIN = \frac{\alpha\mu}{\alpha\mu + \epsilon_b + \epsilon_s}$$

```
epsi <- param_optim$par[1]
miu <- param_optim$par[2]
alph <- param_optim$par[3]
delt <- param_optim$par[4]

pin <- (alph * miu)/(alph * miu + 2 * epsi)
pin
```

```
## [1] 0.02620556
```

By this result we can see there is a very low probability (2.6%) of the presence of informed traders.