***Predicting price changes with Signed Order Imbalance for equity securities***

**Objective:**

This project aims to improve one period ahead price prediction using order imbalance information from the current time period.

**Background information:**

**Motivation behind the project:**

Securities price prediction is a golden grail. The ability to accurately predict price changes lead to direct profits and revenue generation for the firm. Endless efforts have been devoted into coming up with great models, and many are fruitless while others are more promising. The most successful strategies are usually very secretive and involve advanced mathematical modeling. In this paper, the team dabbled at building one simple model based on signed order imbalances, an idea inspired by Easley, Lopez de Prado and O’Hara’s paper on “Flow Toxicity and Liquidity in a High Frequency World”.

**Definition of SOI:**

**Explanation of the definition:**

defines the total volume of a bucket. Such bucket volume needs to identical throughout the day. volume(i, i-1) defines the total volume in a singular “bin” within each bucket. This term need not to be fixed, although it is possible to fix such a term based on volume (i.e. fixed bin implementation). Alternatively, one can define this term in terms of time; to put it in another way, we aggregate all the volumes within a fixed time interval and define it as volume(i,i-1), where i is the time index associated with each time interval. b(i) is a discrete function only takes value {+1,-1,0}, which indicates the act (Bull/Sell/Neutral) of the corresponding bin.

The classification method is explained in detail in the following part.

**Buy/Sell/Neutral Classification (i.e. b(·) Calculation ):**

Multiple buy/sell/neutral classification schemes are explored:

1. Lee-Ready type classification based solely on trade data. In this algorithm, transactional level prices for trades are used. A trade is classified as a buy trade if the transactional price at time t is higher than the price at t-1, and classified as a sell trade if the transactional price at time t is lower than the price at t-1. In case both transactional prices are identical, we look back one period (i.e. t-2 to t-1 period) and follow the same procedure. The rationale behind this is in case of the tie, we implicitly assume the momentum will continue. If the t-2 to t-1 period return is still 0, then we classify the bin of t-1 to t as neutral, and return 0. (i.e. this particular trade has no weight in calculating signed order imbalance)
2. Nearest quotes with artificial delay based classification.

Let us first explain how we use the nearest quotes to classify trades, then we will explain the rational behind adding a constant artificial delay time to all the quotes.

For each trade, this algorithm looks at the nearest available quote ahead of it (in terms of exchange time). If the transacted price is closer to the bid of the quote, we classify the trade as a sell by setting b to -1 (i.e. we are taking the bid). Alternatively if the transacted price is closer to the ask of the quote, the trade is classified as a buy (or b = 1). If the transacted price is equidistant, we assigned the trade as neutral.

The trades reported to the consolidates tape is usually late than the ), to To remedy for that case, we introduce random artificial delay to the quotes and follow the same algorithm for classification.

Quotes delay technique can and has been applied to the raw data for this project even if we do not use technique 2) for classification.

1. The third classification scheme involves calculating time weighted EMAs of all the quotes, and starting with the nearest EMA quote, we follow technique 2) classification.

**Analysis:**

Before proceeding to building an actual predictive model, it is important to first ascertain our intuition that SOI is well correlated with price returns. The simple linear regression model we used to verify is as follows:

,

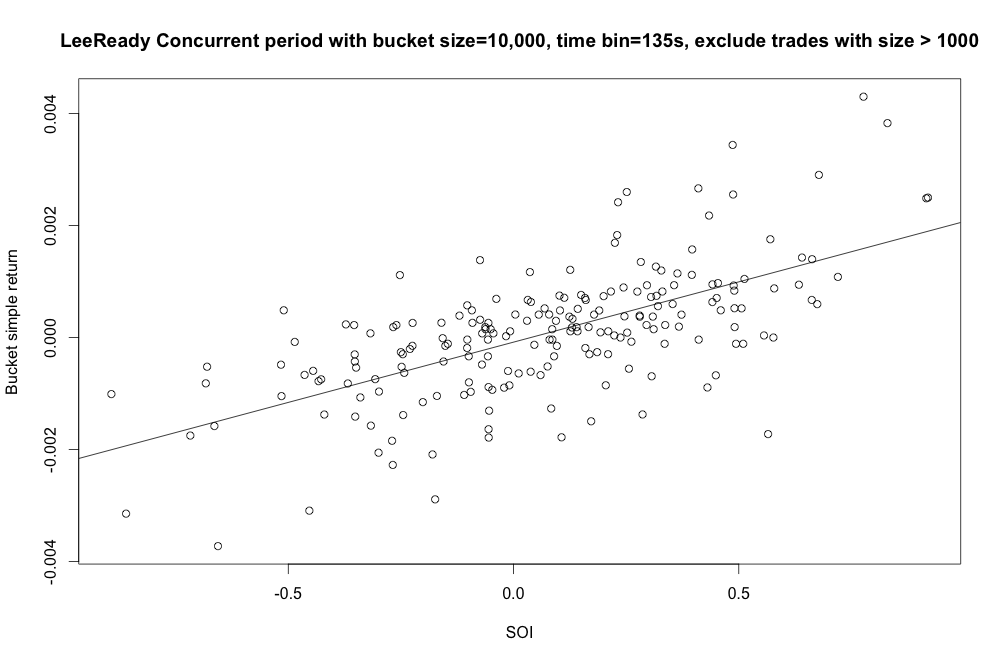
There are several free model parameters we need to determine in order to calculate SOI. The two major ones are bucket size and bin size. The first one is determined by volume, while the second one is determined by time in our implementation, though both can be determined either by volume or time. It is important to calibrate the ideal bucket and bin size, because the regression results are sensitive to the bucket size and bin size.

To illustrate why this can be the case, imagine a bucket size of 10 shares, a 100 shares trade would fill all 10 consecutive buckets, and the computed SOI for all the ten periods would be 1. From the surface, it is almost as if we can detect a pattern of autocorrelation of SOI measures, yet the consecutive SOI measures are simply due to splitting the trades. As for time bins, a single time bin aggregates all the trades within the time period, and is consider the smallest unit of processing. If the time bin is set too small, we may be including a lot of ultra-short term noise, if the time bin is too large, we lose a level of granularity, which can be helpful in improving the model.

To help establishing feasibility, we used AMZN stock for 4/23/2013. By using brute computing force, and regressing all the possible bucket size, bin size, and block trade threshold limit within a pre-specified limits, we find the following combinations to be ideal in terms of adjusted R^2.

|  |  |
| --- | --- |
| **Bucket(shares)\_bin(s)\_Exclude(shares)** | **Adj-R^2** |
| 10000 \_ 135 \_ 1000 | 41.30% |
| 10000 \_ 150 \_ 1000 | 41.30% |
| 10000 \_ 120 \_ 1000 | 37.40% |
| 8000 \_ 135 \_ 1000 | 37.10% |
| 9000 \_ 120 \_ 1000 | 36.60% |

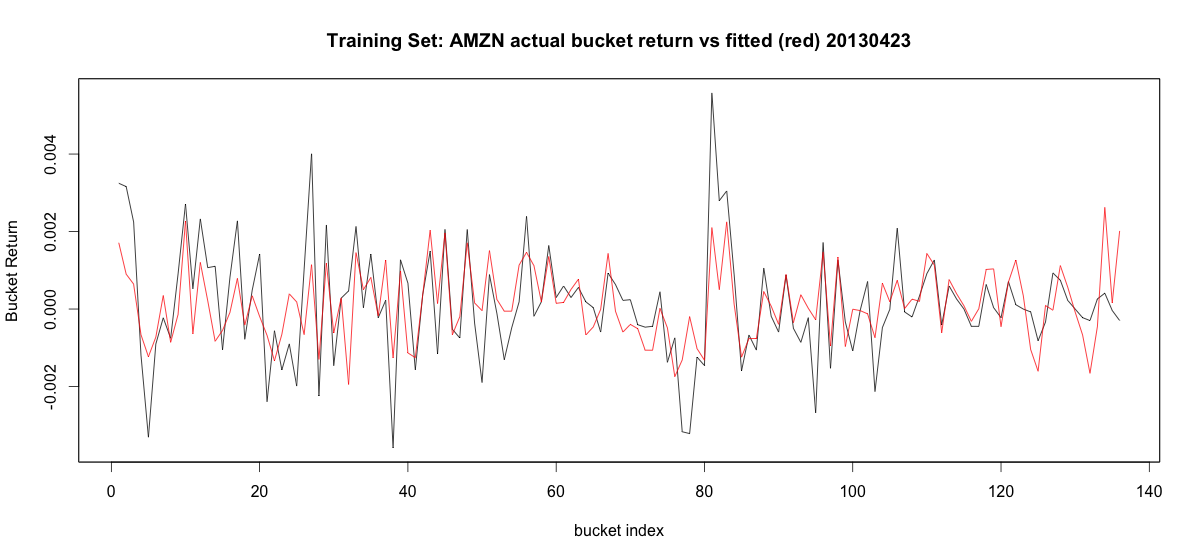
To give the readers an idea how well a R^2 of 41.30% look like, the regression plot is shown below:

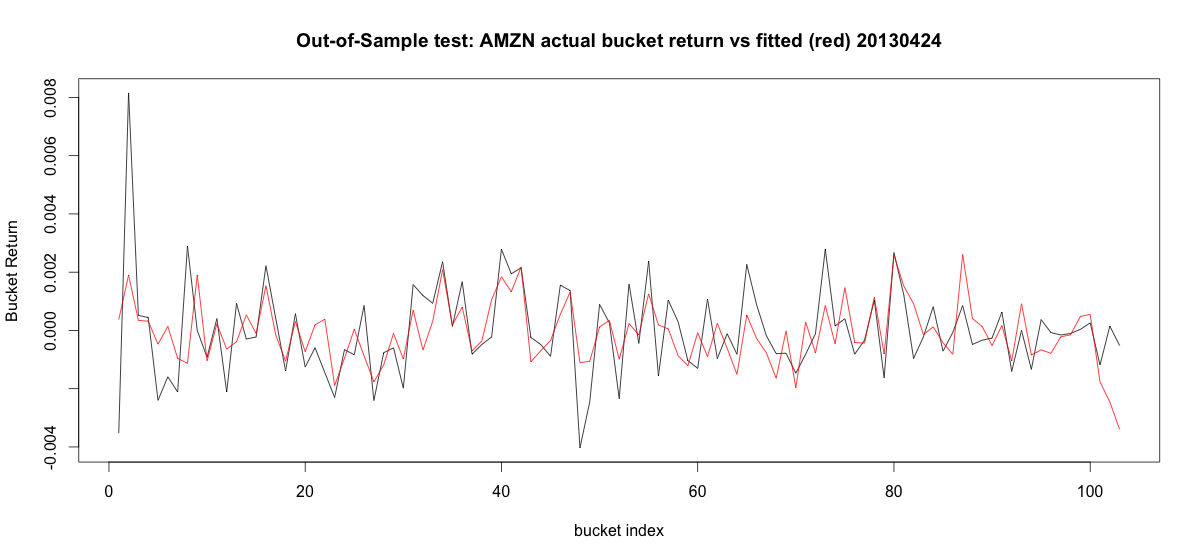


After achieving this step, we moved on to 1) regressing the SOI and the cross term against trade returns and 2) adjusting for the latencies from exchanges, the results improved to:

|  |  |
| --- | --- |
| **Bucket(shares)\_bin(s)\_Exclude(shares)** | **Adj-R^2** |
| 2000 \_ 90 \_ 2000 | 52.18% |
| 15000 \_ 135 \_ 1000 | 50.03% |
| 2000 \_ 105 \_ 2000 | 49.68% |
| 14000 \_ 135 \_ 1000 | 47.79% |
| 15000 \_ 105 \_ 1000 | 47.29% |

We then tested the model robustness across time. Use AMZN 2013/04/23rd data as training data, and 24th data as test set.





Then model captures the movement direction quite well, it correctly predict 94.2% of the price movement direction on 24th. This ratio is calculated as . But as the plots indicated, it is not strong in terms of magnitude depiction.

Next, we want to assess the marginal effect by using different types of classification rules, and the one we are specifically interested is buy/sell classification rule #2, and the weaker R^2 results demonstrated below has shown the Lee-Ready type of classification rule is superior:

|  |  |
| --- | --- |
| **Bucket(shares)\_bin(s)\_Exclude(shares)** | **Adj-R^2** |
| 10000 \_ 60 \_ 3000 | 12.5% |
| 6000 \_ 30 \_ 1000 | 11.4% |
| 6000 \_ 150 \_ 1000 | 9.5% |
| 10000 \_ 60 \_ 5000 | 9.5% |
| 5000 \_ 60 \_ 1000 | 9.5% |

Thus far, we have established the usefulness of using a cross term, the superiority of using delayed exchange time over arrival time, and the validity of SOI’s correlation with price movement.

Thus far, we have used trades’ transactional prices to compute bucket returns. Yet sometimes, practitioners are more interested to predict the quote changes. Not only trade level price changes are noise-prone, one is not guaranteed to be executed at the recorded trade price. So instead, we reformulated the problem, and define mid quote changes in between bucket as price returns.

Next, we assess the We first calculated EMA of all the quotes, and regress the exponential moving average of mid quotes against SOI multiplied by the variance of trade prices within a bucket:

The EMA calculation is defined as:

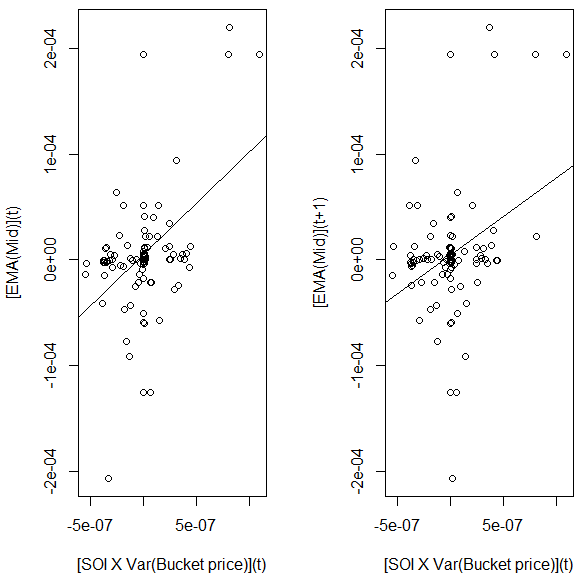
In addition, we have predicted one period ahead R(t) using the same model:

This model is useful in predicting the one period ahead mid-quote. We also introduce one additional twist by artificially delaying the exchange time on the quotes.

We tested this model using five different stocks for five consecutive days. As the stock Below are the output we obtained for one stock AGN, they left column head contains the parameters specific to each regression run, namely the quotes delay, and the EMA decay

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **4/23/2013** |  | **4/24/2013** |  | **4/25/2013** |  |
| *Concurrent Regression* | | | | | |
| Key (delay(s)\_decay) | Adj R2 | Keys | Adj R2 | Keys | Adj R2 |
| 0.025\_1.5 | 33.83% | 0\_0.5 | 20.58% | 0.055\_0.5 | 24.58% |
| 0.03\_1.5 | 33.80% | 0.1\_0.5 | 18.39% | 0.075\_0.5 | 24.28% |
| 0.025\_1.25 | 33.70% | 0.075\_0.5 | 18.18% | 0.08\_0.5 | 24.18% |
| 0.03\_1.25 | 33.70% | 0.09\_0.5 | 18.08% | 0.09\_0.5 | 24.12% |
| 0.03\_1 | 33.19% | 0.085\_0.5 | 18.08% | 0.085\_0.5 | 24.12% |
| *One Step Ahead prediction* | | | | | |
| Key (delay(s)\_decay) | Adj R2 | Keys | Adj R2 | Keys | Adj R2 |
| 0.025\_1.5 | 42.25% | 0\_0.5 | 14.54% | 0.055\_0.5 | 4.64% |
| 0.03\_1.5 | 42.12% | 0\_0.75 | 13.95% | 0.005\_0.5 | 4.56% |
| 0.025\_1.25 | 40.88% | 0\_1 | 13.02% | 0.09\_0.5 | 4.49% |
| 0.03\_1.25 | 40.76% | 0\_1.25 | 11.34% | 0.075\_0.5 | 4.44% |
| 0.025\_1 | 38.90% | 0\_1.5 | 9.25% | 0.05\_0.5 | 4.39% |

|  |  |  |  |
| --- | --- | --- | --- |
| **4/26/2013** |  | **4/29/2013** |  |
| *Concurrent Regression* | | | |
| Key | Adj R2 | Key | Adj R2 |
| 0.06\_0.5 | 14.66% | 0.045\_0.5 | 43.07% |
| 0.06\_0.75 | 13.98% | 0.085\_0.5 | 42.20% |
| 0\_0.5 | 11.72% | 0.07\_0.5 | 42.00% |
| 0.06\_1 | 11.54% | 0.065\_0.5 | 41.77% |
| 0.085\_0.5 | 9.70% | 0.06\_0.5 | 41.74% |
| *One Step Ahead prediction* | | | |
| Key | Adj R2 | Key | Adj R2 |
| 0.1\_0.5 | 7.25% | 0.005\_0.5 | 1.30% |
| 0.07\_0.5 | 7.19% | 0\_0.5 | 0.80% |
| 0.095\_0.5 | 7.18% | 0.075\_0.5 | 0.51% |
| 0.09\_0.5 | 7.14% | 0\_1.5 | 0.45% |
| 0.075\_0.5 | 7.08% | 0\_1.25 | 0.24% |



Above are two sample plots correspond to the cells high-lighted in cyan.

We also did one out of sample run by first estimating the delay and decay parameters by averaging these parameter values from 4/23 to 4/26. Then we obtained the 4/29 concurrent and one period ahead predictions using the estimated model parameters (set delay = 0.035s, and decay = 0.75). Lastly we compare the R^2s with in sample calibrated parameters for 4/29. Below are the differences in performance:

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Out of Sample Concurrent R2 | Key | Best In Sample Concurrent R2 |
| 0.035\_0.75 | 35.54% | 0.035\_0.75 | 43.07% |
| Key | Out of Sample Prediction R2 |  | Best In Sample Prediction R2 |
| 0.045\_0.5 | 0.00% | 0.005\_0.5 | 1.30% |

Even though the decay and delay parameters are not very stable from day to day, using an average value of these parameters to construct out of sample regression yields satisfactory results for this particular example.

Below are the results we have obtained using the same methodology for AMZN stocks.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **4/23/2013** |  | **4/24/2013** |  | **4/25/2013** |  |
| *Concurrent Regression* | | | | | |
| Key (delay(s)\_decay) | Adj R2 | Key | Adj R2 | Key | Adj R2 |
| 0.045\_0.75 | 29.91% | 0.03\_0.75 | 6.50% | 0.005\_1.5 | 4.20% |
| 0.045\_0.5 | 29.43% | 0.035\_0.75 | 6.49% | 0.005\_1.25 | 4.11% |
| 0.045\_1 | 26.65% | 0.06\_0.5 | 6.46% | 0.005\_1 | 3.69% |
| 0.045\_1.25 | 14.90% | 0.055\_0.5 | 6.45% | 0.005\_0.75 | 2.86% |
| 0.005\_0.75 | 3.70% | 0.05\_0.5 | 6.38% | 0.095\_1.5 | 2.79% |
| *One Step Ahead Prediction* | | | | | |
| Key | Adj R2 | Key | Adj R2 | Key | Adj R2 |
| 0.085\_1.5 | 19.34% | 0.055\_0.5 | 4.49% | 0.005\_1.25 | 8.22% |
| 0.085\_1.25 | 19.32% | 0.05\_0.5 | 4.46% | 0.005\_1.5 | 8.02% |
| 0.085\_1 | 18.19% | 0.06\_0.5 | 4.44% | 0.005\_1 | 7.82% |
| 0.085\_0.75 | 16.00% | 0.065\_0.5 | 2.54% | 0.095\_0.75 | 7.28% |
| 0.085\_0.5 | 12.65% | 0.1\_0.5 | 2.14% | 0.1\_0.75 | 7.20% |

|  |  |  |  |
| --- | --- | --- | --- |
| **4/26/2013** |  | **4/29/2013** |  |
| *Concurrent Regression* | | | |
| Key | Adj R2 | Key | Adj R2 |
| 0\_0.5 | 20.73% | 0.04\_0.75 | 16.67% |
| 0\_0.75 | 19.48% | 0.045\_0.75 | 16.48% |
| 0\_1 | 16.31% | 0.02\_1 | 16.39% |
| 0\_1.25 | 13.14% | 0.04\_1 | 15.89% |
| 0.055\_0.75 | 11.40% | 0.045\_1 | 15.62% |
| *One Step Ahead Prediction* | | | |
| Key | Adj R2 | Key | Adj R2 |
| 0\_0.5 | 9.07% | 0.04\_0.75 | 19.54% |
| 0.05\_0.5 | 8.99% | 0.045\_0.75 | 19.35% |
| 0.05\_0.75 | 8.35% | 0.04\_1 | 18.69% |
| 0.09\_0.5 | 7.51% | 0.045\_1 | 18.42% |
| 0.09\_0.75 | 7.42% | 0.04\_0.5 | 17.71% |

We see the results are particularly good for 4/23/2013, which is the date when White House’s twitter account was hacked. As a result, we expect SOI in generally to work better when the market is relatively more volatile and filled with more impactful news.

We believe in this intuition, because as more market participants are aware of potential impactful news, their behaviors became more heavily driven by the market psychology, or their interpretation of the market psychology and reaction. Consequently, an algorithm that looks solely at the buy/sell imbalance sentiments becomes more effective. Though without more carefully designed experiments and quantitative data to back up this hypothesis, this observation only remains as an interesting hypothesis. If time permits, we plan to assess how our model behaves during different market environments.

**<TO JIA: IF POSSIBLE, please expand to include the other 3 stocks, you can put them in the Appendix, make sure the data is consistent with the observation above, or else change the conclusion>**

**<TO JIA: EXPAND on The following>**

Note a potentially alternate form of this model defines R(t) as follows:

Lastly, we want to assess the effectiveness of buy/sell/neutral classification rule #3:

**<TO JIA: All these data are in the zip file I emailed you, there is a folder called EMA\_BUY\_SELL, please do some analysis similar to the ones above>**