

Adaptive Strategies for High Frequency Trading

Erik Anderson and Paul Merolla and Alexis Pribula

Stock prices on electronic exchanges are determined at each tick by a matching algorithm which matches buyers with sellers, who can be thought of as independent agents negotiating over an acceptable purchase or sell price. Exchanges maintain an “order book” which is a list of the bid and ask orders submitted by these independent agents. At each tick, there will be a distribution of bid and ask orders on the order book. Not only will there be a variation in the bid and ask prices, but there will be a variation in the numbers of shares available at each price. For instance at 11:00:00 on February 12, 2008, the order book on the Chicago Mercantile Exchange carried the quotes for the ES contract as shown below:

Bid	Ask	# Bid Shares	# Offer Shares
1362	1362.25	81	79
1361.75	1362.5	610	955
1361.50	1362.75	934	1340
1361.25	1363	815	1786
1361	1363.25	1492	1335

The goal of this project was to explore whether the order book is an important source of information for predicting short-term fluctuations of stock returns. Intuitively, one would expect that when the rate and size of buy orders exceeds that of sell orders, the stock price would have a propensity to drift up. Whether this happens or not ultimately depends on how the agents react and update their trading strategies. Our hope is that by considering the order book, we can better predict agent behaviors and their effect on market dynamics, as opposed to a prediction method that did not consider the book.

There were three phases to our project. First we obtained an adequate historical data source (Section I), from which we extracted metrics based on the order book (Section II). Next, we developed an adaptive filtering algorithm that was designed for short-term prediction (Section III). To evaluate our prediction method, we created a simulated trading environment and back tested a simple trading strategy for market making (Sections IV, V, & VI). We discovered that our prediction technique works well for the most part, however, coordinated market movements (sometimes called market shocks) often resulted in large losses. The final phase of project was to employ a more sophisticated machine learning technique, support vector machines (SVMs), to forecast upcoming shocks and judiciously disable our market making strategy (Sections VII, VIII and IX). Our results suggest that combining short-term prediction with risk management is a promising strategy. We also were able to conclude that the order book does in fact increase the predictive power of algorithm, giving us an additional edge over less sophisticated market making techniques.

I. ORDER BOOK DATA SET

We required an appropriate data set for training and testing our prediction algorithms. Initially we hoped to use the TAQ database, however, it turns out that the TAQ only keeps track of the best bid and the best offer (the so called *inside quote*) across various exchanges. These multiple inside quotes are in many cases *not* equivalent to the order book, since typically, most of the trading for a particular security occurs on one or a few exchanges.

Our ideal data set would be a comprehensive order book, which would include every market and limit order sent to an exchange along with all of the relevant details (e.g., time, price, size, etc.). Such data sets do in fact exist, but currently the cost to gain access to them is prohibitive.

The data that we were able to get access to was the ES contract spanning from February 12, 2008 to March 26, 2008; ES is a mini S&P futures contract traded on the Chicago Mercantile Exchange (CME) that has a cash multiplier of \$50. Unlike the comprehensive book, our data set contains approximately three snapshots a second of the five best bids and offers (Figure 1). There are two points to note from this figure: First, it is clear that the ES can only be traded at 0.25 intervals (representing \$12.5 with the cash multiplier), which is a rule enforced by the exchange. Therefore, price quotes from the order book do *not* provide any additional information (i.e., the queued orders always sit at 0.25 intervals from the inside quote); we verified for our data set that this is almost always the case during normal trading hours. Second, our data set exists in a quote-centered frame. That is, when the inside quote changes, different levels of the queue are either revealed or covered. In the following section, we describe a method to convert from a quote-centered frame to a quote-independent frame, which turns out to be an important detail when extracting information from the book.

II. EXTRACTING RELEVANT PARAMETERS FROM THE ORDER BOOK

Here, we describe two quantities that we extracted from the order book that are related to supply and demand: The first quantity that we tracked was the cumulative sum of the orders in the bid and offer queues. In essence, this sum reflects how

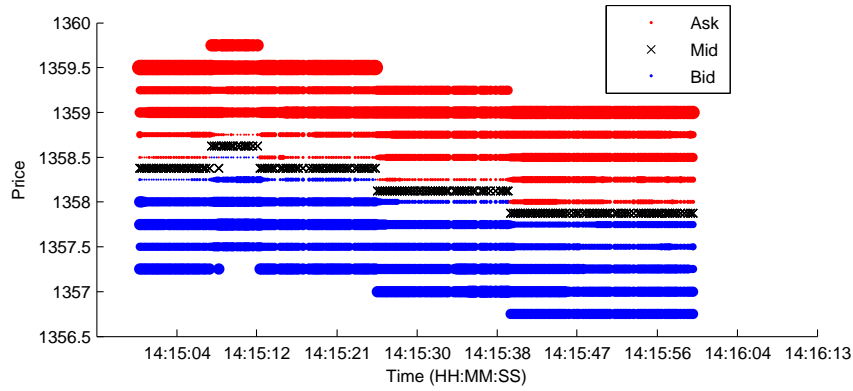


Fig. 1. One minute of the order book for the ES contract showing the five best bids (blue) and asks (red) on February 12, 2008; data was obtained from the open source jBookTrader website [1]. Market snapshots arrive asynchronously and are time stamped with a millisecond accuracy (rounded to seconds in the figure for clarity). Order volumes, which range from 3 to 1008 in this example, are indicated by the size of each dot.

many shares are required to move a security by a particular amount (often referred to in the literature as the price impact [2]). One helpful analogy is to think of these queued orders as barriers that confine a security’s price within a particular range. For example, when the barrier on the buy side is greater than the one on the sell side (more buy orders in the queue than sell orders), we might expect the price to increase over the next few seconds. Of course whether this happens ultimately depends on how these barriers evolve in time (i.e., how agents react and update their trading strategies), however we expect that knowing the barrier heights will help us to better predict market movements.

In addition to the cumulative sum, we also estimated the rates at which orders flow in to and out of the market. To continue with our analogy, these rates reflect how barriers are changing from moment to moment, which could be important for detecting how agents react in certain market conditions. In fact, a recent paper suggests that agents modeled as sending orders with a fixed probability (Poisson statistics) can explain a wide range of observed market behaviors on average [3]. Based on this result, we expect that tracking how these rates continuously change might capture market behaviors on a shorter time scale.

A subtle but critical issue is that when the market shifts (i.e., there is a new inside quote), our measurements become corrupted because we are using a quote-centered order book. For example, consider when the price changed from 1358.25 to 1358.5 right after 14:15:04 in Figure 1. Here the cumulative sum for the bid queue suddenly decreases from 3517 to 2690 simply because a part of the queue has now become covered (conversely, the cumulative sum of the ask queue will increase because part of the queue is now revealed). Even though there has not been a fundamental change in supply and demand, our cumulative sum metric shifts by 23%! There are a number of ways to correct for this superfluous jump — our approach was to simply subtract out the expected change due to a market shift, estimated from previous observations. Specifically, we tracked how each level of the queue changes for the previous 200 market shifts (Figure 2), and subtracted out the expected amount when a shift was detected. Returning to our previous example, we can now see that right after 14:15:04 in Figure 1, the cumulative sum only changed from 3517 to 3325 after correcting for the expected change (now only a 5% change). In an similar manner, we also adjusted our rate calculations by keeping track of the direction of the shift, and then computing the rate across neighboring price levels.

III. ADAPTIVE MINIMUM MEAN SQUARED ERROR PREDICTION OF BEST BID/ASK PRICES

In this section, we discuss using adaptive minimum mean squared error estimation (MMSE) techniques to predict the best bid or ask prices M ticks into the future. We may estimate the best ask (\hat{A}) and the best bid (\hat{B}) using

$$\begin{aligned} \hat{A}(k+M) &= \mathbf{x}^T(k) \mathbf{c}_a(k) \\ \hat{B}(k+M) &= \mathbf{x}^T(k) \mathbf{c}_b(k). \end{aligned}$$

where $\mathbf{x}^T(k)$ is a $1 \times P$ vector of parameters from the order book. The time-varying filter coefficients ($\mathbf{c}_a, \mathbf{c}_b$) may be determined by training for the best fit over the L prior data sets ($-(L-1) \leq k \leq 0$). We use a cost function which is a sum of squared errors

$$C = \sum_{m=k-L+1}^k (y(m) - \hat{y}(m))^2$$

where y and \hat{y} are the observed and predicted values respectively.

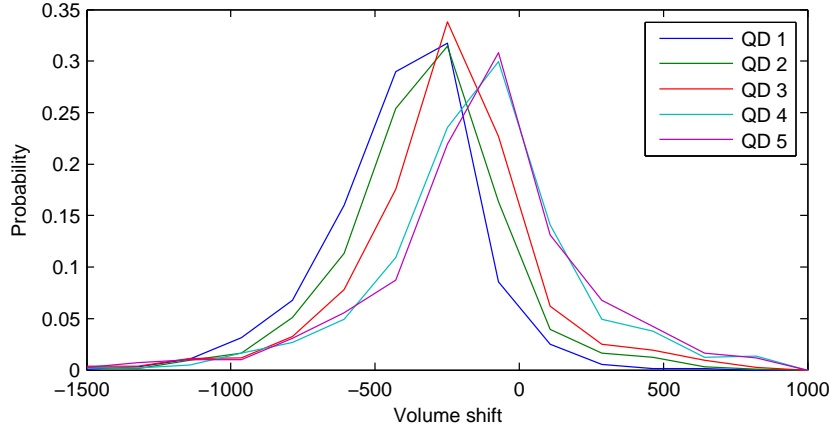


Fig. 2. Probability of how much the volume at a particular queue depth (QD) has changed immediately after an upward market shift (i.e., the best bid increased); note a QD of 1 represents the inside quote. Near the inside quote (QD’s of 1, 2, and 3), there is a pronounced negative skew indicating that new bids have decreased after an upward market shift, whereas bids away from the inside quote (QD’s of 4 and 5) are symmetric around 0 suggesting that they are less effected by market shifts. We also track these distributions for downward market shifts (plot not shown).

The prediction coefficients are then determined from

$$\begin{aligned}
 \mathbf{c}_a(k) &= (\mathbf{Q}^T(k) \mathbf{Q}(k) + \lambda \mathbf{I})^{-1} \mathbf{Q}^T(k) \mathbf{y}(k) \\
 \mathbf{y}(k) &= (A(k); \dots; A(k-L+1)) \\
 \mathbf{Q}^T(k) &= (\mathbf{x}^T(k-M); \dots; \mathbf{x}^T(k-M-L+1))
 \end{aligned} \tag{1}$$

where \mathbf{y} is an $L \times 1$ vector of historical observations of the variable to be estimated, \mathbf{Q}^T is a $P \times L$ matrix of the data to be used in the estimation, and λ is for regularization of a possibly ill-conditioned matrix $\mathbf{Q}^T \mathbf{Q}$. Note that in the calculation of the coefficient vector, estimations at time k is based on the historical book data vector at time $k-M$. Similar equations hold for calculation of the coefficient vector for estimation of the best bid prices. Note that the prediction coefficients are updated at each time step in order to adapt to changes in the data.

Note that this method allows for many types of input vectors to be used in predicting the best bid/ask prices at the next time bin. The input vector \mathbf{x} may consist of the best bid/ask prices, bid/ask volumes, or other functions or statistics computed from the data over a recent historical window. Randomly including various parameters in the input vector will not result in a good predictor; parameters need to be judiciously chosen.

IV. ORDER CLEARING

In order to test the above algorithms for market making, we must have some sort of methodology for simulating a stock exchange order matching algorithm. Here we discuss this methodology. We assume there is a uniform network latency so that limit orders intended for time bin $k+M$ arrive at the exchange at this time bin. If the bid limit orders on the book at time $k+M$ are greater than the best bid order of the actual data during this time bin, then we assume the orders clear with probability 1. If the bid orders on the book equal the actual best bid order, then the orders clear with probability p (hereafter the “edge clearing” probability). Bid orders less than the actual best bid price do not clear and remain on the book until they clear or until they are canceled. Similarly, ask orders on the book at time $k+M$ which are less than the actual ask price during this bin clear with probability 1. If the ask orders equal the actual ask price, then the orders clear with probability p . Ask orders greater than the actual best ask price do not clear and remain on the book until they clear or until they are canceled. All of this assumes that we trade in small enough quantities so as to not perturb what would have actually happened. Higher order simulations requires data for how your algorithm interacts with the exchange.

V. TRADING ALGORITHM

We test our bid/ask price prediction methods by using a straightforward trading rule which works as follows. As a caveat, this trading rule is by no means optimal and certainly is in need of risk-management to be built in with it. We use it merely as a means of keeping score of how well the price prediction methods are working. At time k , we employ $1/N$ of our capital to send a limit bid order to the exchange at the predicted bid price. The parameter N can be thought of as the number of time steps it initially takes to convert all our cash into stock assuming that all orders are filled; spreading purchases over several time steps gives time diversity to the strategy. All purchases are made in increments of 100 shares with 200 shares being the

minimum number of shares in a bid order. Furthermore at time k , we attempt to sell all shares that we own with a limit ask order at the predicted ask price. Orders remain on the exchange until they are filled or canceled.

We must set aside capital for bid orders until they are filled or canceled. If the best bid/ask prices ever move far enough away from existing orders on the book, the orders are canceled to free up the allocated capital for bid/ask orders at prices more likely to be filled. We assume that transaction fees are 0.35 ¢/share for executed bid/ask orders and also that order cancellation costs 12 ¢ . Admittedly, the fees are steep compared to what funds actually pay.

VI. MARKET MAKING SIMULATIONS

In this section, we run some back-tests applying the methodology as discussed above. We use the ES contract using some trading days in February and march of 2008. Simulations initially assume that we start with 1.5 million. Figure 3 shows that the accumulated wealth for market making is a strong function of the edge clearing probability. The solid lines depict wealth accumulation for an edge clearing probability of 40% whereas the dashed lines assume a probability of 20%; wealth increases over time at the higher probability and decreases over time with the lower probability. The MMSE approach using best bids/asks is compared with two other price prediction strategies: use the last best bid/ask as a prediction for the next time step and use a moving average of best bids/asks. Note that the MMSE approach here does not perform any better than the two other more simplistic strategies. As shown in Figure 4, which uses an edge clearing probability of 30%, using higher order book data increases profitability. Figure 5 shows the lifetime of bid/ask orders, i.e. how many time steps does it take to fill the orders across the various strategies. Also shown is how many times steps orders remain on the book until canceled. Note that the higher order approach has slightly higher bid/ask order executions. Figure 8 shows an example trading day for the ES contract (3-20-08). The strategies have difficult times during times of sudden price movements as would be expected – there is still a need to incorporate risk management.

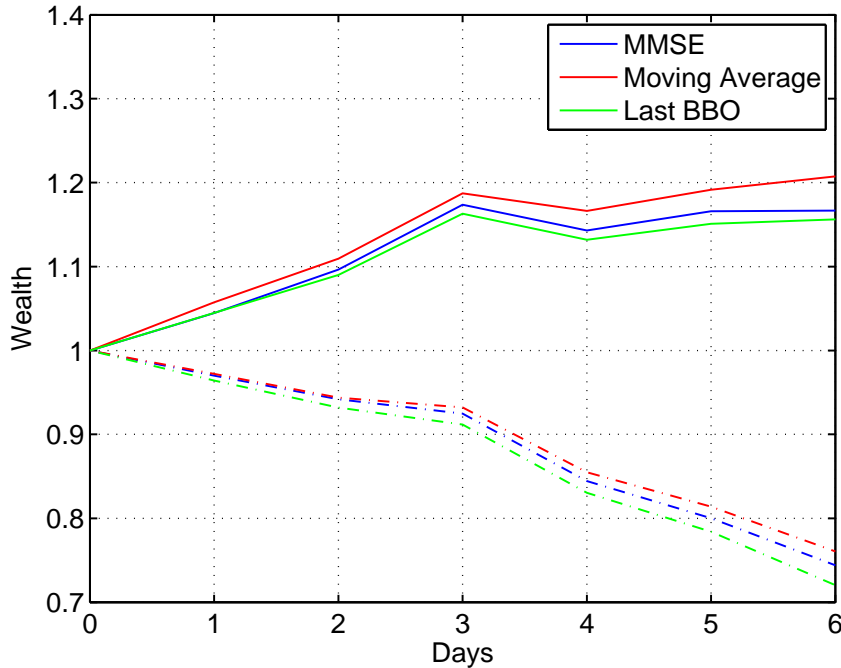


Fig. 3. Wealth versus time depends on the probability of orders being filled at the best bid/ask. The solid curves show that wealth increases for an edge clearing probability of 40% while the dashed curves show that wealth decrease when the probability is reduced to 20%.

VII. SUPPORT VECTOR MACHINES (SVM)

We also used Support Vector Machines. SVM solve the problem of binary classification (for more information please see chapter 7 of [4]). Say we are given empirical data:

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in \mathbb{R}^n \times \{-1, 1\}$$

where the *patterns* $\mathbf{x}_i \in \mathbb{R}^n$ describe each point in the empirical data and the *labels* $y_i \in \{-1, 1\}$ tell us in which class each point of the empirical data belongs to. SVM trains on this data so that when it receives a new unlabeled data point $\mathbf{x} \in \mathbb{R}^n$, it tries as best as it can to classify it by providing us with a predicted $y_n \in \{-1, 1\}$. The way it does this is by using *optimal soft*

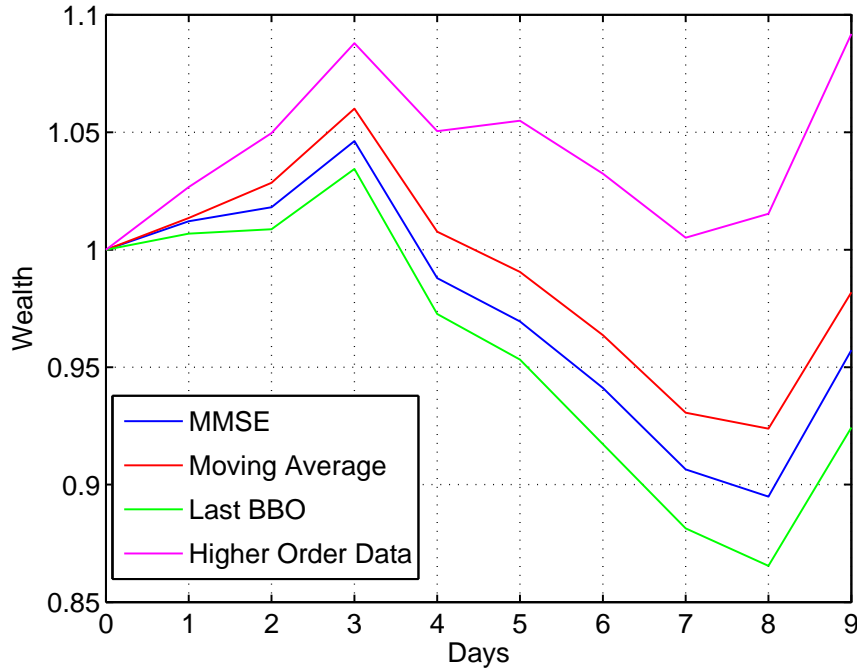


Fig. 4. Higher order use of data increases the returns for an edge clearing probability of 30%.

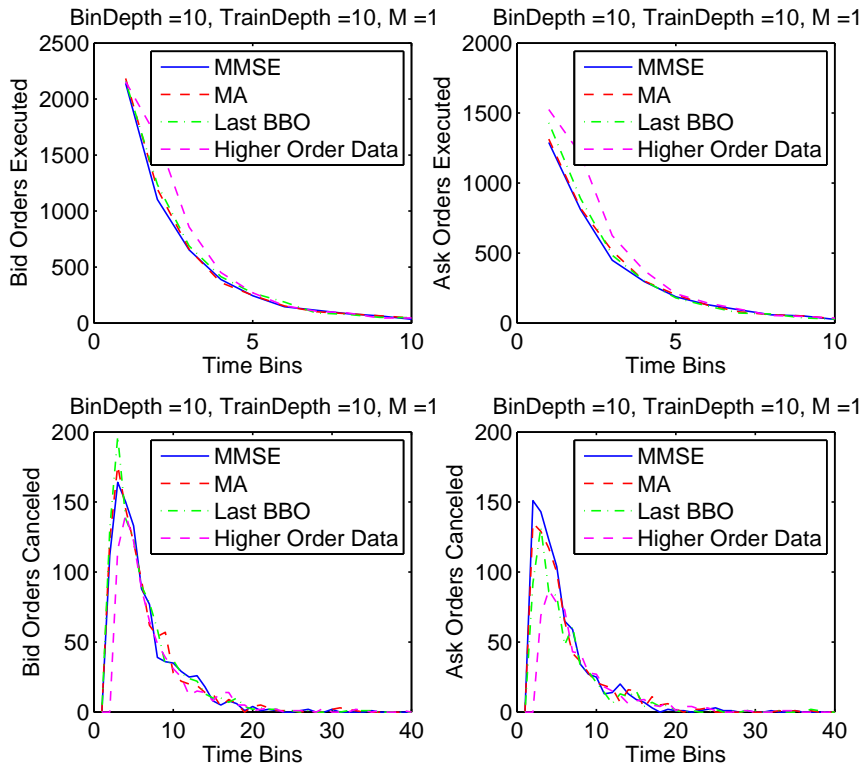


Fig. 5. Number of time steps required to fill bid/ask orders for 1 day of trading. Also shown is the number of time steps orders remain on the book until canceled. Each time bin is a 2 second interval for the ES contract on 3-20-08.

margin hyperplanes. That is, given the empirical data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^n \times \{-1, 1\}$, SVM tries to find a hyperplane such that all the patterns $\mathbf{x}_i \in \mathbb{R}^n$ with label $y_i = -1$ will be on one side of this hyperplane and all patterns with labels $y_i = 1$ will be on the other side. This hyperplane is an *optimal margin* hyperplane in the sense that it will have the greatest possible

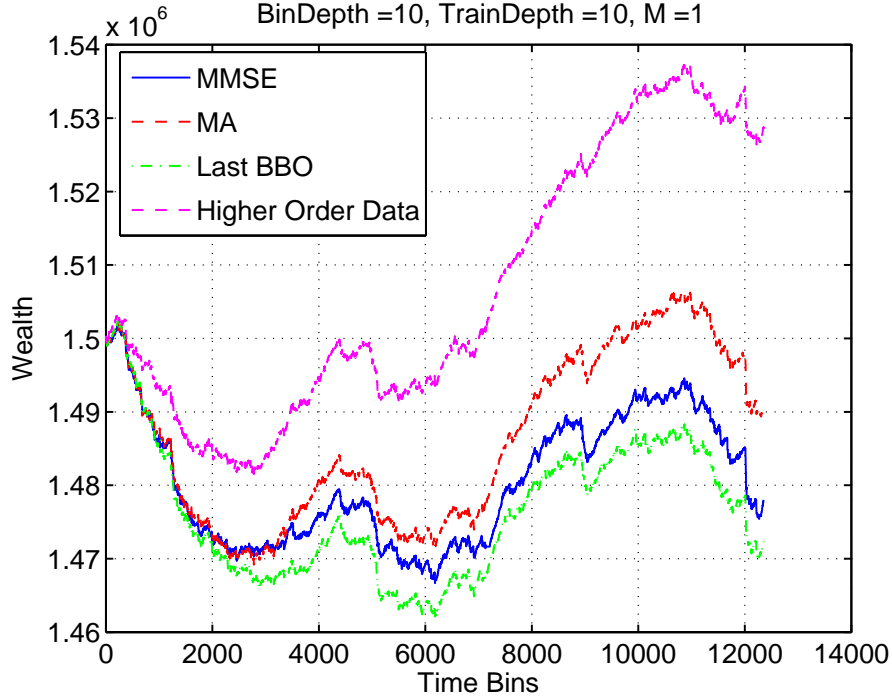


Fig. 6. An example trading day (ES contract on 3-20-08). Each time bin is a 2 second interval and the edge clearing probability is 30%.

distance between the set of $y_i = -1$ patterns and the set of $y_i = 1$ patterns. This turns out to be important because it improves the generalization ability of the classifier (through the use of *Structural Risk Minimization*), i.e. it improves how well it will classify new untested patterns. The last term, *soft* hyperplane, means that if the data is not separable, then slack variables will be used to soften the hyperplane and allow outliers. Finding the optimal soft margin hyperplane can done with the following quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{i=1}^m \xi_i \\ \text{subject to} \quad & y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \forall i \in \{1, \dots, m\} \end{aligned}$$

Furthermore, to be able to find non-linear separation boundaries, SVM uses kernels instead of the dot product. That is, instead of computing $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ for two patterns, it computes $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$, where $k(\cdot, \cdot)$ is a positive definite kernel (i.e. for any $m \in \mathbb{N}$, and for any patterns $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$, the matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite). This enables SVM to find a linear separating hyperplane in a higher dimensional space and then from it, create non-linear separation boundaries in the original data space (see Figure 7).

VIII. INDEPENDENT COMPONENT ANALYSIS (ICA)

When trying to predict market shocks (see below), SVM was not quite reliable enough. Thus we used the Independent Component Analysis (ICA) dimensionality reduction algorithm on the data before sending it to the SVM. The way it works is the following. Say we have a random sample $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ distributed from an inter-dependent random vector $\mathbf{x} \in \mathbb{R}^n$. The goal of ICA is to linearly transform this data into components $\mathbf{s}_1, \dots, \mathbf{s}_m \in \mathbb{R}^p$ which are as independent as possible and such that the dimension p of the \mathbf{s}_i is smaller than the dimension n of the original \mathbf{x}_i (for more information, see chapters 7 and 8 of [5]). This transformation is achieved in two steps. First, we whiten the data, that is, we center it and normalize it so that we get $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m \in \mathbb{R}^n$ sampled from $\tilde{\mathbf{x}} \in \mathbb{R}^n$ with $\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^m \tilde{\mathbf{x}}_i = \mathbf{0}$ and such that the matrix $\Sigma_{ij} = \frac{1}{m} \sum_{r=1}^m \tilde{\mathbf{x}}_{ri} \tilde{\mathbf{x}}_{rj} - \boldsymbol{\mu}_i \boldsymbol{\mu}_j = \frac{1}{m} \sum_{r=1}^m \tilde{\mathbf{x}}_{ri} \tilde{\mathbf{x}}_{rj}$ is the identity matrix. Second, to find independent components, the following heuristic is used. Let \mathbf{B} be the matrix we want to use to transform our data, then

$$\mathbf{s} = \mathbf{B}\tilde{\mathbf{x}}.$$

By the central limit theorem, the sum of any two independent and identically distributed random variables is always more gaussian than the original variables. Hence to get \mathbf{s} which is as independent as possible, ICA finds rows of \mathbf{B} such that the resulting \mathbf{s}_i are as non-gaussian as possible. This is generally done by testing for kurtosis or other features that gaussians are known not to have. Moreover, by limiting the number of rows of \mathbf{B} , we can reduce the dimensionality of the data.



Fig. 7. An illustration of how Support Vector Machines divides data into two separate regions. This example is one for which the data is perfectly separable.

IX. APPLICATION OF ICA + SVM FOR SHOCK PREDICTION

Market making strategies are especially effective during calm trading days. This is because they have been specifically created to trade the spread as many times as possible, which works very well in low volatility days. However, when market shocks occur, the algorithms get into the spiral of either buying as the contract is moving down or selling as the contract is moving up. Indeed, when the market jumps, these strategies still hope that it makes sense to buy at the bottom of the spread in order to be able to shortly sell later on at a higher price. Unfortunately, when a shock occurs, the market does not come back and these orders simply lose money.

To solve this problem, we used ICA+SVM to forecast the direction of the market 1 minute in advance in order to warn and stop the market making strategies operating every 2 seconds before shocks occurred. ICA was applied to very long windows of data (ranging from hours to days) so that it could effectively find the most interesting feature of the data in all of the contract's history. Then the results were fed into a SVM which operated on smaller time windows (ranging from minutes to hours). The trading methodology was the same used as in section VI with data ranging from March 17 2008 through to March 26 2008. The type of data used to make the predictions was: the last best bid, the last best ask, the corrected bid rate and the corrected ask rate (as explained in section II). Finally the parameters of the SVM and ICA were optimized on training data from February 20 2008. These were: 6 independent components for ICA with a time window as long as data permitted, for SVM we used $C = 4$, $\gamma = 4$ and a training window of 25 minutes.

The choice of first applying ICA onto the data makes sense for two reasons. First, it reduces the data and keeps only the most relevant information, which stops SVM from stumbling on irrelevant data which would not help it predict anything. The second reason for doing this is based on the fact that a known heuristic of classifiers is that they perform better on independent data rather than dependent data. Using different lengths of time windows for ICA and SVM made sense because giving more data to ICA improves its capacity to create more interesting independent components. Indeed, ICA's goal is to find components which are as different as possible in the data. Thus more data means more possible different components. This implies two things for the SVM which uses a short time window from the end of the time window of the independent components: first since ICA had more data, SVM gets the benefit of getting better independent components, second by optimizing its time window, we are still giving SVM the possibility to chose how far relevant data can be found.

X. CONCLUSIONS AND FUTURE WORK

In this report, we discussed adaptive strategies for high frequency trading. We showed that use of the order book can increase profitability of trading strategies over first-order approaches. Moreover, we proposed a method to help reduce the impact of market shocks which uses Support Vector Machines and Independent Component Analysis. Although further back-testing is still warranted, these methods show promise. Further research directions include optimizing the trading rule to be used in conjunction with the price predictor and to incorporate additional risk-management above and beyond a predictor of market shocks. Moreover, the Support Vector Machine method could be optimized (tweaking the input data, varying the training window, etc.).

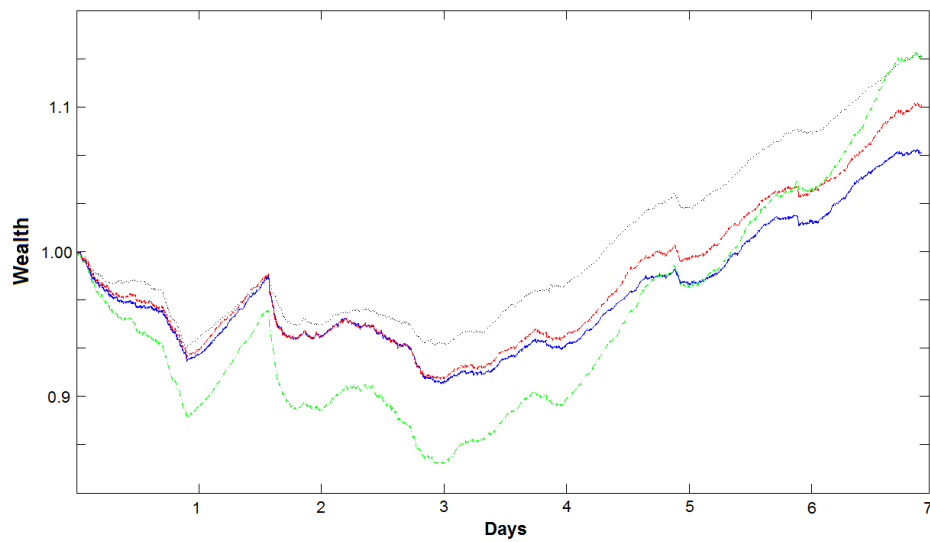


Fig. 8. Example trading days (ES contract on 3-17-08 through 3-26-08) for the SVM (thin black dotted line) compared with other first-order predictors.

REFERENCES

- [1] [Http://code.google.com/p/jbooktrader/](http://code.google.com/p/jbooktrader/).
- [2] V. Plerou, P. Gopikrishnan, X. Gabaix, and H. E. Stanley, "Quantifying stock-price response to demand fluctuations," *Phys. Rev. E*, vol. 66, no. 2, p. 027104, Aug 2002.
- [3] J. D. Farmer, P. Patelli, and I. I. Zovko, "The predictive power of zero intelligence in financial markets." *Proceedings of the National Academy of Science*, vol. 102, no. 6, pp. 2254–2259, February 2005.
- [4] B. Scholkopf and A. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*. Boston, Massachusetts: MIT Press, 2002.
- [5] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, New York: John Wiley and Sons, 2001.