A boosting approach for automated trading*

Germán Creamer
Center for Computational Learning Systems
Columbia University
475 Riverside MC 7717
New York, NY 10115
ggc14@columbia.edu

Yoav Freund
Department of Computer Science
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0114
yfreund@cs.ucsd.edu

ABSTRACT

This paper describes an algorithm for short-term technical trading. The algorithm was tested in the context of the Penn-Lehman Automated Trading (PLAT) competition. The algorithm is based on three main ideas. The first idea is to use a combination of technical indicators to predict the daily trend of the stock, the combination is optimized using a boosting algorithm. The second idea is to use the constant rebalanced portfolios within the day in order to take advantage of market volatility without increasing risk. The third idea is to use limit orders rather than market orders in order to minimize transaction costs.

1. INTRODUCTION

The recent development of electronic communication networks (ECNs) or electronic financial markets has allowed a direct communication between investors, avoiding the additional cost of intermediaries such as the specialists of the New York Stock Exchange (NYSE). A very important aspect of the ECNs is the access and publication of the real-time limit order book. For many years such access was not available to most traders. For example, in the NYSE only specialists could observe the entries of the limit order book. Other investors could only see the price and number of shares of the executed orders.

Electronic markets maintain a centralized order book for each traded stock. This book maintains lists of all active limit orders and is used as the basis for matching buyers and sellers. By making the content of this book accessible to traders, electronic markets provide a very detailed view of the state of the market and allow for new and profitable trading strategies. For example, Kakade, Kearns, Mansour, and Ortiz in [14] present a competitive algorithm using volume weighted average prices (VWAP).[1] Kavajecz and Odders-White [17] study how technical analysis indicators can capture changes in the state of the limit order book.

In this paper we present an automated trading algorithm that was tested in the context of the Penn-Lehman Automated Trading (PLAT) competition. The algorithm is based on three main ideas. The first idea is to use a combination of technical indicators to predict the daily trend of the stock. The trading algorithm uses the stock price of the previous ninety days, and the open price of the current trading day to calculate a set of well-known technical analysis indicators. Based on this information, the trader anticipates the direction of the market using a boosting algorithm, and then takes a long or short position if it expects that the market will go up or a down respectively. The second idea is to use constant rebalanced portfolios [1] within the day in order to take advantage of market volatility without increasing risk. This part of the trading algorithm puts limit orders to assure that there is a constant mix between the value of the stocks and of the portfolio. The third idea is to use limit orders rather than market orders in order to minimize transaction costs. The trader accesses the order book to put limit orders out of the bid-ask spread to capture the rebates that ECNs such as ISLAND pay to the trader whose submission was in the order books at the moment of execution. [2]

The rest of the paper is organized as follows: section 2 introduces boosting; section 3 presents the PLAT competition and our trading strategy; section 4 presents the results of the participation of our trading algorithm in the PLAT competition; section 5 introduces improvements to our algorithm

*A preliminary version of this paper was presented at the Data Mining for Business Applications Workshop on International Conference on Knowledge Discovery and Data Mining (KDD), Philadelphia, 2006.

.

[1]VWAP is calculated using the volumes and prices present on the order book.

[2]A market order is an order to buy an asset at the current market price. A buy (sell) limit order is executed only at a price less (greater) or equal than the limit price. The ECNs register the orders in the order book which is continuously updated with new orders or when an order is executed.

The bid-ask spread refers to the difference between the bid price or the highest price that a trader is willing to pay for an asset, and the ask price or the lowest price that a trader is willing to sell an asset.

A long position is the result of buying a security expecting that the value of the underlying asset goes up. A short position is the result of selling a borrowed security expecting that the value of the underlying asset goes down.

Technical analysis is a method to forecast security prices and trends using patterns of prices, volumes, or volatility (see the appendix).

$$
\begin{array}{|l|}
\hline
\text{for } t = 1 \ldots T \\
\quad w_i^t = \frac{1}{1+e^{y_i F_{t-1}(x_i)}} \\
\quad \text{Get } h_t \text{ from } \textit{weak learner} \\
\quad \alpha_t = \frac{1}{2} \ln \left(\frac{\sum_{i:h_t(x_i)=1, y_i=1} w_i^t}{\sum_{i:h_t(x_i)=1, y_i=-1} w_i^t} \right) \\
\quad F_{t+1} = F_t + \alpha_t h_t \\
\hline
\end{array}
$$

Figure 1: The Logitboost algorithm. y_i is the binary label to be predicted, x_i corresponds to the features of an instance i, w_i^t is the weight of instance i at time t, h_t and $F_t(x)$ are the prediction rule and the prediction score at time t respectively

such as the integration of the market maker strategy, and section 6 concludes and discusses futures lines of research.

2. METHODS

2.1 Boosting

Adaboost is a machine learning algorithm invented by Freund and Schapire [12] that classify its outputs applying a simple learning algorithm (weak learner) to several iterations of the training set where the missclasified observations receive more weight.

Friedman et al [13], followed by Collins, Schapire, and Singer [6] suggested a modification of Adaboost, called Logitboost. Logitboost can be interpreted as an algorithm for step-wise logistic regression. This modified version of Adaboost –known as Logitboost– assumes that the labels $y_i's$ were stochastically generated as a function of the $x_i's$. Then it includes $F_{t-1}(x_i)$ in the logistic function to calculate the probability of y_i, and the exponent of the logistic function becomes the weight of the training examples. Figure 1 describes Logitboost.

We implemented boosting with a decision tree learning algorithm called an *alternating decision tree* (ADT) [11]. In this algorithm, boosting is used to obtain the decision rules and to combine them using a weighted majority vote (See Creamer and Freund [9] for a previous application to several finance problems).

The importance of features used to predict earnings surprises, and cumulative abnormal returns may change significantly in different periods of time. As we do not know in advance what the most important features are and because of its feature selection capability, its error bound proofs [12], its interpretability, and its capacity to combine quantitative, and qualitative variables we decided to use boosting as our learning algorithm.

3. TRADING STRATEGIES AND PLAT COMPETITION

3.1 Automated trading – PLAT

Our trading algorithm is tested in the Penn-Lehman Automated Trading Project[3] (see Kearns and Ortiz [18]). This project, which is a partnership between the University of Pennsylvania and the quantitative trading division of Lehman Brothers, simulates ISLAND, one of the major ECNs, and has had trading competitions since the Fall of 2002.

The simulator that supports PLAT captures price and volume information of ISLAND about every 3 seconds, and provides an architecture where clients can connect and submit limit orders. During the competition of April-May 2004, Microsoft (MSFT) is the only stock that is traded. The simulator creates its own order book receiving the information of ISLAND and mixes it with the orders of its clients.

The simulator generates detailed information about the position of each trader: market and price simulator, outstanding shares, present value, and profit and loss position.

PLAT is different from the well-known trading agent competition (TAC) run at the University of Michigan [24] because of PLAT's strict limitation to the financial market and because only one stock is traded: Microsoft. The classic TAC game is based on the travel industry market, and since 2003, it has also included a supply chain management game. Wellman et al. in [24] reports recent results of TAC. Both competitions, PLAT and TAC, are similar in terms of offering a platform and software for agents to develop their trading strategies.

3.1.1 PLAT Competition

We designed the trading algorithm "CRP_TA" that participated in the PLAT competition run in the period April 26 to May 7, 2004. The rules used during this competition were[4]:

1. The performance of each trader is measured by the Sharpe ratio calculated as the mean return and standard deviation of the 10-day profit and loss positions.

2. Traders do not have a limit in terms of number of shares that they can hold. However, positions must be liquidated at the end of the day. Any long position will completely lost its value, and any short position must pay a penalty of twice its market value.

3. Transaction costs will follow ISLAND's fee/rebate policy: when a trade is executed, the party whose order was in the order books shall receive a rebate of \$ 0.002, and the party that submitted the incoming order shall pay a transaction fee of \$ 0.003

During this competition, participants were split into two groups: red and blue. Our agent was team 1 in the red group.

The competition also included an agent per team that bought and sold large number of shares each day following the volume weighted average price (VWAP).

3.2 Trading algorithm

Our basic approach is to separate our analysis of the market into two time scales. The long time scale is on the order of days or hours, the short time scale is on the order of seconds or minutes. When operating on the long time scale we use a variety of technical indicators (see the appendix) to predict price trends. In other words, we try to predict whether the stock price will go up or down in the next day or next hour. When operating on the short time scale we stick to the prediction given by the long time scale analysis

[3]This description of PLAT refers to Spring 2004 when we participated in the competition.

[4]Further explanation of the PLAT project can be found at <http://www.cis.upenn.edu/~mkearns/projects/plat.html>

and place orders in a way that would take maximal advantage of volatility, and minimize transaction costs.

In more detail, our long time scale analysis is based on an adaptive combination of technical analysis indicators. The combination is optimized using the boosting learning algorithm and past month as training data. The short time-scale trading is based on constant rebalanced portfolios with a time-based profile selected according to the long-term analysis. Finally, the actual market orders are generated in a way designed to take advantage of the transaction cost policy used in ISLAND, one of the major ECNs and the one used as the data source for the PLAT competition.

We call our trading algorithm CRP_TA because it implements a hybrid strategy of a) forecasting the daily stock price with Logitboost using technical indicators (TA), and b) intra-day trading following a constant rebalanced portfolio (CRP) strategy.

3.2.1 *Applying Logitboost to the selection of technical trading rules*

The trading algorithm CRP_TA forecasts the direction of the stock price using ADTs which are implemented with Logitboost. We introduced this algorithm in section 2.1. CRP_TA trains ADTs using the following technical analysis indicators of the previous ninety days and described in the appendix: simple moving average, average directional movement index, directional movement index, Bollinger bands, moving average convergence divergence, relative strength index, stochastic indicators, and money flow index. We calculated these indicators using R and its financial engineering package called Rmetrics. [5]

The instances are labeled using the following rules:

Buy, if $P^c \geq P^o + \tau$

Sell, if $P^c \leq P^o - \tau$

Hold, otherwise

where τ is a constant that at least covers the transaction costs ($0.003), and P^o and P^c are the close and open price respectively.

Logitboost generates a new set of trading rules. Hence, instead of using the rules that each technical analysis indicator suggests, Logitboost defines what are the appropriate rules based on the market conditions and the combination of a list of very well-known technical indicators.

3.2.2 *Constant rebalanced portfolio*

Constant rebalanced portfolio, known in the financial world as constant mix, is a well-known strategy in the investment community. Kelly [19] showed that individuals that invest the same proportion of their money on a specific asset–the constant rebalanced portfolio–their portfolio value will increase (or decrease) exponentially. Kelly introduced the log-optimal portfolio as the one which achieves the maximum exponential rate of growth. Algoet and Cover [1] showed that if the market is stationary ergodic, the maximum capital growth rate of a log-optimal portfolio is equivalent to the maximum expected log return. Cover [8] and later on many other researchers such as Vovk and Watkins [23], Cover and Ordentlich [7] Blum and Kalai [2], and Kalai and Vempala [15] extended CRP to the concept of universal constant rebalanced portfolio.

CRP simply requires that traders maintain a fixed proportion of stocks to portfolio value. If stock price increases (decreases), the stock to portfolio value ratio increases (decreases), then part of the stocks must be liquidated (bought). This strategy works better when the stock price is unstable, so the trader is able to sell when the price is high, and buy when the price is low.

We tested the trading algorithm CRP_TA in the PLAT competition run between April 26 to May 7, 2004. Every day of the competition CRP_TA trains an ADT with Logitboost using the information of the last ninety days and then using P^o takes a long position (50% of the portfolio invested in MSFT), short position (25% of the portfolio) or do not trade. During the first half hour CRP_TA builds its position, and during the half hour before the market closes, CRP_TA liquidates its position. There is an asymmetry between the long position (50%) and the short position (-25%) because of the higher penalty that a trader with a short position would pay during the competition. The training of ADTs was done using the MLJAVA package.[6]

The trading algorithm CRP_TA trades during the day balancing the portfolio according to a goal mix as Figure 2 explains. CRP_TA intends to increase revenues sending limit orders and expects that these orders arrive before than the counterparty's orders when the orders are executed. In this case, the trader receives rebates, and avoids paying fees.

4. PLAT COMPETITION RESULTS

After ten trading days of participating in the PLAT competition, CRP_TA obtained a return of $27,686 and the Sharpe ratio was 0.83. Its performance was the second best in its group as Figure 3 shows. CRP_TA forecasted correctly a short or long position eight out of the ten days of the PLAT competition. These results were better than the results of a simulation for a sample of 840 days when the predictor was trained with information of the last 90 days. In this last case the test error was 48.8%. These differences could be explained because the optimization of the parameters used to calculate the technical indicators at the beginning of the competition might have not been adequate for other periods. We spent a significant amount of time fine tuning the parameters used for the forecast. Additionally, the trader did not get its position at the open price as the above simulation did it. It reached its position after the first half hour of trading.

To understand the intra-day dynamic, we present the results of a trading day when the market is up and down (Figure 4). May 3rd was a very volatile day and the market was up, while CRP_TA got a short position. The losses of a short position were partially compensated by the benefits of intra-day trading thanks to the CRP strategy. On April 28th the market went down. CRP_TA assumed a short position that led to a profitable position. This last result is evident in the top panel of Figure 4 that shows an important difference between the portfolio value index and the index price or buy and hold (B&H) position.

[5]Information about R and Rmetrics can be found at <http://cran.r-project.org> and at <http://www.rmetrics.org> respectively.

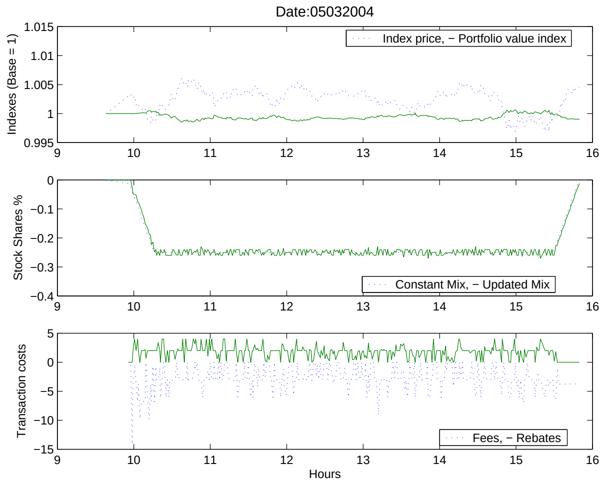
[6]If interested in using MLJAVA, please contact yfreund@cs.ucsd.edu.

Input:
Set of price series (open (P^{o}), close P^{c}, high (P^{h}), low (P^{l})), and volume
τ is a constant that at least covers the transaction costs (\$0.003)
q_g is goal mix of stocks and cash for MSFT
Forecast with machine learning algorithm (Logitboost) and technical indicators (TA):
1. At the beginning of the day, train an ADT with Logitboost using training set with technical analysis indicators, and labels (see the appendix) calculated with price and volume series of the last 90 days.
2. Forecast trend of P^{c} using P^{o} and technical analysis indicators for trading day, and take one of the following positions for single stock (MSFT) in first half hour of trading:
Long ($q_g = 50\%$), if $E(P^{\text{c}}) \geq P^{\text{o}} + \tau$
Short ($q_g = -25\%$), if $E(P^{\text{c}}) \leq P^{\text{o}} - \tau$
Hold, otherwise
Intra-day constant rebalanced portfolio (CRP):
3. Sends simultaneously a buy and sell limit orders for δ according to:
Submit buy limit order for δ, if $q_t < q_g - \delta/W$
Submit sell limit order for δ, if $q_t > q_g - \delta/W$
Hold, otherwise
where W is net value portfolio, q_t is current mix of stocks and cash for MSFT, and δ is amount of dollars to buy or sell in order to reach q_g.
4. If ($q_t != q_g$) after 60 ticks (about one minute), cancel limit orders, submit market orders to obtain q_t, and submit new limit orders.
5. Liquidate position in the last half hour before market closes.
Output:
Profit/loss of algorithm

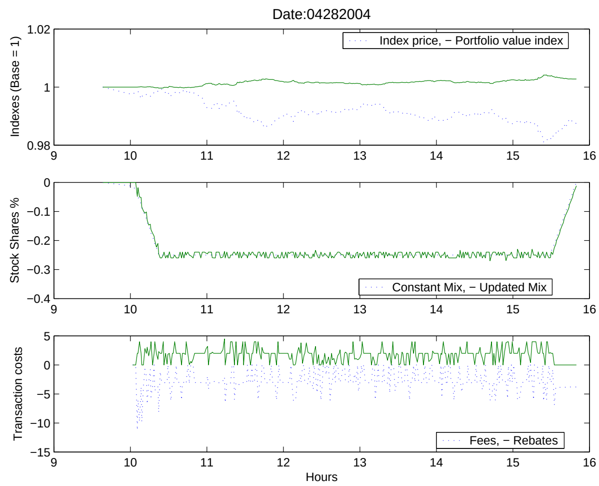
Figure 2: The CRP_TA algorithm.

	Profit and loss											
	Sharpe Ratio	26/4	27/4	28/4	29/4	30/4	3/5	4/5	5/5	6/5	7/5	Total
Team1	**0.8334**	**2249**	**-151**	**7527**	**7198**	**6628**	**-2523**	**1567**	**2238**	**1885**	**1068**	**27687**
Team2	-0.1619	27	-513	-3062	1219	3204	-153	327	15	61	-4601	-3476
Team3	1.1221	3574	7083	-127	-2832	2040	6691	4335	6108	5915	3061	35847
Team4	-0.4232	-44962	3147	-1185	-1832	-988	-88302	946	1129	1907	2316	-127825
Team5	-12.6200	-9.E+06	-8.E+06	-9.E+06	-8.E+06	-9.E+06	-7.E+06	-8.E+06	-8.E+06	-8.E+06	-7.E+06	-8.E+07
Team6	0.7213	1045	4729	243	-6694	12508	11065	-2377	5708	9271	11755	47252
Team7	2.4963	3433	1374	2508	2928	3717	3444	1322	3300	2199	966	25190
Team8	0.7559	271	538	-242	-248	13	636	386	452	461	121	2387
Team9	0.5778	1307	2891	-1563	-1349	-1339	3230	1850	2037	2465	1041	10569
Team10	0.0432	-4655	-1370	2178	2820	2766	2961	2665	-5746	2402	-2545	1475
Team11	-12.5931	-9.E+06	-8.E+06	-7.E+06	-8.E+06	-8.E+06	-7.E+06	-8.E+06	-8.E+06	-8.E+06	-8.E+06	-8.E+07

Figure 3: Profit and loss of PLAT competition for all players. Competition was split in the first five teams (red group) and the next five teams (blue group). First column shows the Sharp ratio for each team during the whole competition. Additional columns have daily profits or losses for each team expressed in US\$. CRP_TA is team 1. Teams 5 and 11 are artificial traders who bought and sold large volume of shares following the VWAP.



(a) Market up



(b) Market down

Figure 4: Representative intraday results of PLAT competition for CRP_TA when market is up (a) and down (b). Top graphs compare portfolio value index with an index price or a simple buy and hold (B&H) position. Middle graphs compare the goal or constant mix of stocks and cash with the updated mix according to the trading algorithm. The steeper curve at the beginning and at the end of the trading day is the period when CRP_TA builds and liquidates its goal position. Bottom graphs include fees (> 0) and rebates (< 0). The differences between rebates and fees are transaction costs.

During each trading day there were a large number of trading operations. However, the process to adjust the portfolio to reach the goal mix affected the results because the trader CRP_TA paid more fees than received rebates as the bottom of Figure 4 shows. The winner on CRP_TA's group during the PLAT competition, team 3, acted as a market maker placing limit orders outside the current spread. Hence, an important amount of CRP_TA's orders were plausibly traded with this team; however this trader did not pay fees, only received rebates because their orders were limit orders that most of the time arrived first than CRP_TA's orders. If CRP_TA could incorporate this market marker strategy, probably its results may improve as we show in the next section.

5. IMPROVED ORDER STRATEGY

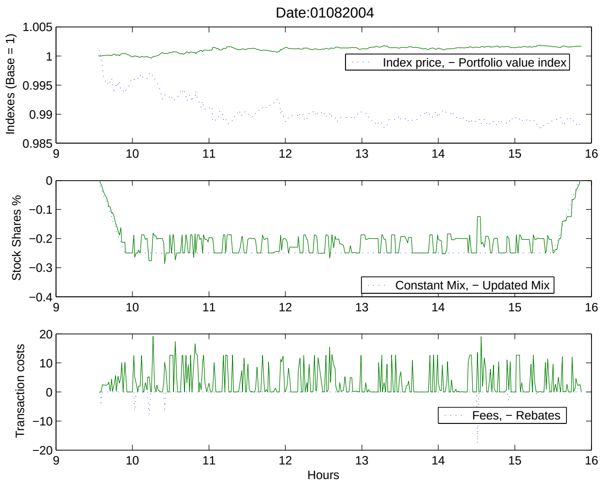
After the PLAT competition, we integrated the market maker strategy into the CRP_TA, and we call the modified version of the algorithm as the "Market maker CRP_TA". The most important aspect of the revised version of the algorithm is that the orders should be executed as limit orders, and not as market orders as follows: Market maker CRP_TA starts with a balanced position according to the proportion of shares over portfolio value established as a goal (q_g). Then it sends simultaneously a buy limit order at a price slightly below ($0.005) than the price at the top of the buy order book (P_{BuyB}), and a sell limit order at a price slightly above ($0.005) than the price at the top of the sell order book (P_{SellB}). If the order is not completely filled within ten minutes of being issued, existent limit orders are canceled, and limit orders are reissued. In all cases, orders are reissued for the amount necessary to reach the goal mix of stocks and cash (see Figure 5).

Input:
Set of price series (open (P^{o}), close P^{c}, high (P^{h}), low (P^{l}), and volume
τ is a constant that at least covers the transaction costs ($0.003)
q_g is goal mix of stocks and cash for MSFT
κ is minimum amount above or below top price of order books ($0.005)
Forecast with machine learning algorithm (Logitboost) and technical indicators (TA):
1. At the beginning of the day, train an ADT with Logitboost using training set with technical analysis indicators, and labels (see the appendix) calculated with price and volume series of the last 90 days.
2. Forecast trend of P^{c} using P^{o} and technical analysis indicators for trading day, and take one of the following positions for single stock (MSFT) in first half hour of trading:
Long ($q_g = 50\%$), if $E(P^{\mathrm{c}}) \geq P^{\mathrm{o}} + \tau$
Short ($q_g = -25\%$), if $E(P^{\mathrm{c}}) \leq P^{\mathrm{o}} - \tau$
Hold, otherwise
Intra-day market maker constant rebalanced portfolio (CRP):
3. Sends simultaneously a buy and sell limit orders for δ according to:
Buy limit order for δ and $P_B = P_{BuyB} - \kappa$, if $q_t < q_g - \delta/W$
Sell limit order for δ and $P_S = P_{SellB} + \kappa$, if $q_t > q_g - \delta/W$
Hold, otherwise
where W is net portfolio value, q_t is current mix of stocks and cash for MSFT, δ is amount of dollars to buy or sell in order to reach q_g, P_B and P_S are prices of long and short limit orders, P_{BuyB} and P_{SellB} are prices at the top of the buy and sell order book respectively
4. If ($q_t! = q_g$) after 600 ticks (about 10 minutes), cancel and resubmit limit orders to obtain q_t.
5. Liquidate position in last half hour before market closes.
Output:
Profit/loss of algorithm

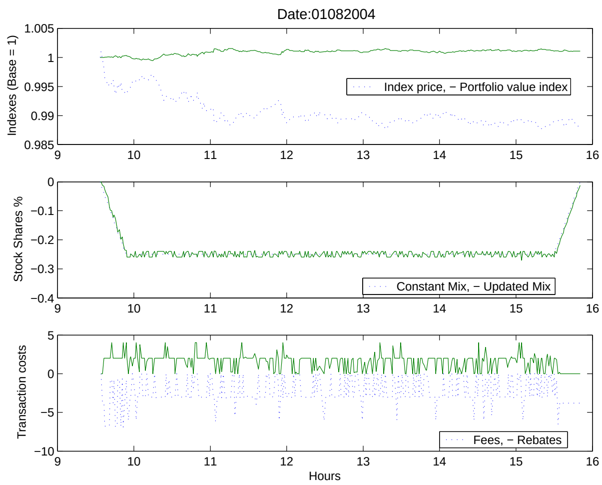
Figure 5: The Money market CRP_TA algorithm.

We run this new trading strategy and the original CRP_TA strategy during the period January 5-9, 2004. We present the results of January 8th for the market maker CRP_TA strategy and for the CRP_TA agent in Figure 6. During the week of January 5-9, the Sharpe ratio is 0.03 and -0.28 for the Market maker CRP_TA strategy and for the CRP_TA strategy respectively. The bottom of Figure 6 shows that Market maker CRP_TA received more in rebates than the amount it had to pay in fees. This difference helped to improve the financial result of the algorithm which is the major shortcoming of the CRP_TA strategy.

Another shortcoming of the CRP_TA strategy is that this strategy takes a high risk when it keeps only a short or long position during the day. A variation of the CRP_TA strategy could be the creation of a portfolio that has a long and short position simultaneously. The scores obtained from Logitboost to forecast the stock price could be used to weight the long and short position. Hence, the position with higher score would have a higher weight. A market neutral portfolio could also be obtained using the same proportion of stocks to portfolio value for the short and long position. We also tried this final alternative for the week of January 5-9, 2004 and the Sharpe ratio deteriorates to -2.06. Obviously, this alternative misses the benefit of market forecasting using ADTs.



(a) Market maker CRP_TA



(b) CRP_TA

Figure 6: Representative intraday results for Market maker CRP_TA (a) and CRP_TA (b) in January 8th, 2004. Top graphs compare portfolio value index with an index price or a simple buy and hold position. Middle graphs compare the goal or constant mix of stocks and cash with the updated mix according to the trading algorithm. The steeper curve at the beginning and at the end of the trading day is the period when trading algorithms build and liquidate their goal position. Bottom graphs present fees (> 0) and rebates (< 0). The differences between fess and rebates are transaction costs.

6. CONCLUSIONS

In this paper we show that the constant rebalanced portfolio or constant mix strategy can improve if a classifier may anticipate the direction of the market: up, down or no change. Additionally, transaction costs play a central role to improve performance. Instead of an automatic rebalance of the portfolio, the results of the PLAT competition indicate that if the CRP strategy is implemented only with limit orders, its results improve because of the rebates.

We used very well known technical indicators such as moving averages or Bollinger bands. Therefore, the capacity to anticipate unexpected market movements is reduced because many other traders might be trying to profit from the same indicators. In our case, this effect is reduced because we tried to discover new trading rules using Logitboost instead of following the trading rules suggested by each indicator. However, we are aware that our predictor may improve if we transform the technical indicators into more accurate ratios or select more informative indicators such as the effect of current news into stock prices.

Our experience in adapting boosting to a trading algorithm is that a simple and straightforward application of boosting to financial time series does not bring a significant improvement in forecasting. There are other well-known methods used for finance problems, such as logistic regression, that have a similar performance to boosting [9]. However, boosting can work with a mixture of quantitative and qualitative indicators, and also with non-linear time series. Furthermore, boosting can be used to understand the nonlinear relationship between the variables, and can automatically select the best features. Our experiments showed that the boosting approach is able to improve the predictive capacity when indicators are combined and aggregated as a single predictor.

Additionally, we recognize that boosting or another learning algorithms used to forecast time series may have a predictive ability for only a certain period of time. However, the randomness and continuous change of the financial market may lead to make ineffective a trading strategy based on boosting or another predictor. Hence, our algorithm can be enriched by the introduction of risk management mechanisms in order to change strategy or liquidate its position if market behaves in unexpected ways.

Acknowledgements

The authors thank Michael Kearns for inviting us to participate into the PLAT competition and for his invaluable comments about our current work. GC also thanks Sal Stolfo for comments about the trading strategy, and to Luis Ortiz and Berk Kapicioglu for their help and suggestions about running the experiments in PLAT.

7. REFERENCES

[1] P. H. Algoet and T. M. Cover. Asymptotic optimality and asymptotic equipartition properties of log-optimum investment. *Annals of Probability*, 16:876–898, 1988.

[2] A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35(3):193–205, 1999. Special Issue for COLT '97.

[3] J. A. Bollinger. *Bollinger on Bollinger bands.* McGraw-Hill, New York, 2001.

[4] T. S. Chande and S. Kroll. *The new technical trader: boost your profit by plugging into the latest indicators.* John Wiley & Sons, Inc., New York, 1994.

[5] J. F. Clayburg. *Four steps to trading success: using everyday indicators to achieve extraordinary profits.* John Wiley & Sons, Inc., New York, 2001.

[6] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and Bregman distances. *Machine Learning*, 48(1-3):253–285, 2004.

[7] T. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2), March 1996.

[8] T. M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

[9] G. Creamer and Y. Freund. Predicting performance and quantifying corporate governance risk for latin american adrs and banks. In *I Proceedings of the Financial Engineering and Applications conference*, MIT-Cambridge, 2004.

[10] J. F. Ehlers. *Rocket science for traders: digital signal processing applications.* John Wiley & Sons, Inc., New York, 2001.

[11] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Machine Learning: Proceedings of the Sixteenth International Conference*, pages 124–133, 1999.

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[13] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, apr 2000.

[14] S. M. Kakade, M. Kearns, Y. Mansour, and L. E. Ortiz. Competitive algorithms for VWAP and limit order trading. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 189–198. ACM Press, 2004.

[15] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3:423–440, 2003.

[16] J. Katz and D. McCormick. *The Encyclopedia of Trading Strategies.* McGraw-Hill, New York, 2000.

[17] K. A. Kavajecz and E. R. Odders-White. Technical analysis and liquidity provision. *Review of Financial Studies*, 2004.

[18] M. Kearns and L. Ortiz. The Penn-Lehman Automated Trading Project. *IEEE Intelligent Systems*, 2003.

[19] J. L. Kelly. A new interpretation of information rate. *Bell System Technical Journal*, 35:917–926, 1956.

[20] M. Pring. *Technical analysis explained.* McGraw-Hill, New York, 4 edition, 2002.

[21] C. J. Sherry. *The new science of technical analysis.* Probus publishing, Chicago, 1994.

[22] T. Stridsman. *Trading systems and money management.* McGraw-Hill, New York, 2003.

[23] V. Vovk and C. Watkins. Universal portfolio selection. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*, pages 12–23, New York, jul 1998. ACM Press.

[24] M. P. Wellman, A. Greenwald, P. Stone, and P. R. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13(1), 2002.

[25] W. Wilder. *New concepts in technical trading systems.* Trend Research, 1978.

[26] E. Zivot and J. Wang. *Modeling financial time series with S-Plus.* Springer, New York, 2003.

Appendix. Technical analysis indicators used during PLAT competition

Technical indicators are statistics of the market that quantify market trends. Most technical indicators have been developed by professional traders using trial and error. It is common practice to use rules based on technical indicators to choose the timing of buy and sell orders. These rules are called buy and sell "signals". In this work we use a combination of market indicators and trading signals. We define these indicators in this appendix and provide the basic intuition that motivates them. Throughout this section we assume a single fixed stock.

We start with some basic mathematical notation. We index the trading days by $t = 1, 2, \ldots$. We denote by P^{o}_t, P^{c}_t, P^{uc}_t, P^{h}_t, and P^{l}_t, the open, adjusted close, unadjusted close[7], high, and low price of the tth trading day. We eliminate the lower index when we wish to refer to the whole sequence, i.e. P^{c} refers to the whole sequence $P^{\mathrm{c}}_1, P^{\mathrm{c}}_2, \ldots$. Using this notation we define the median price $P^{\mathrm{med}} = (P^{\mathrm{h}} + P^{\mathrm{l}})/2$, the typical or average price $P^{\mathrm{typ}} = (P^{\mathrm{h}} + P^{\mathrm{l}} + P^{\mathrm{uc}})/3$, and the weighted close price $P^{\mathrm{wc}} = (P^{\mathrm{h}} + P^{\mathrm{l}} + 2P^{\mathrm{uc}})/4$.

Many of the technical indicators incorporate time averages of prices or of other indicators. We use two types of time averages, the simple moving average and the exponentially weighted moving average.[8] Let $\mathbf{X}$ denote a time sequence $X_1, X_2, \ldots$. The **simple moving average** is defined as

$$\mathbf{SMA}_t(\mathbf{X}, n) = \frac{1}{n}\sum_{s=0}^{n-1} X_{t-s} \ ,$$

and the **exponentially weighted moving average** is defined as

$$\mathbf{EMA}_t(\mathbf{X}, n) = \lambda \sum_{s=0}^{\infty} (1-\lambda)^s X_{t-s}; \quad \lambda = \frac{2}{n+1} \ .$$

A useful property of $\mathbf{EMA}_t(\mathbf{X}, n)$ is that it can be calculated using a simple update rule:

$$\mathbf{EMA}_t(\mathbf{X}, n) = \lambda X_t + (1-\lambda)\mathbf{EMA}_{t-1}(\mathbf{X}, n) \ .$$

In the following table we describe the technical indicators. The parameters of each indicator are in parentheses. Most of the parameters used refer to the length of the period (n) selected to calculate the indicator. In case of exponential moving average, the parameter used is λ which also depends of n. We have assigned parameters which are typically used in the industry for each indicator.

---

[7] Unadjusted close prices are the actual published prices at the end of the trading day. The adjusted stock price removes the effect of stock splits and dividend payments. Our goal is to predict P^{c}_t, the *adjusted* close price.

[8] We follow Zivot and Wang [26] in describing the technical analysis indicators. Additional useful references about technical analysis and trading are [16, 20, 4, 21, 22, 10, 5].

Technical indicators used in PLAT competition

Variable	**Description**	**Calculation detail [Source]**
Price indicators:		
$SMA^c_t(n)$	Simple moving average of the last n observations of a time series P^{c}.	$\mathbf{SMA}_t(P^{\mathrm{c}}, n)$ $where\ n = 3,\ and\ 6$
Bollinger bands:	Using the moving average or the median band ($Boll^m_t(n)$) as the reference point, the upper and lower Bollinger [3] bands ($Boll^u_t(n)$ and $Boll^d_t(n)$ respectively) are calculated in function of s standard deviations. When price crosses above (below) the upper (lower) Bollinger band, it is a sign that the market is overbought (oversold). Technical analysts typically calculate Bollinger bands using 20 days for the moving average and 2 standard deviations.	$Boll^m_t(n) = SMA^c_t(n)$ where n=6
$Boll^u_t(n)$	Upper Bollinger band	$Boll^m_t(n) + s \cdot \sigma^2_t(n)$ where s=2.6 [Katz [16]]
$Boll^d_t(n)$	Lower Bollinger band	$Boll^m_t(n) - s \cdot \sigma^2_t(n)$ where s=2.6 [Katz [16]]
$ADX_t(n)$	Average directional movement index: indicates if there is a trend and the overall strength of the market [25]. Range of values from 0 to 100. A high number is a strong trend, and a low number is a weak trend. The directional movement index (DX_t) is the percentage of the true range ($TRange_n$) that is up ($+DI_t(n)$) or down ($-DI_t(n)$). The true range determines the trading range of an asset.	$ADX_{t-1}(n) \cdot (n-1) + DX_t)/n$ where: $DX_t \doteq \frac{(+DI_t(n))-(-DI_t(n))}{(+DI_t(n))+(-DI_t(n))}$ $TRange_n = max(\mathbf{P^h_n}) - min(\mathbf{P^l_n})$ $n = 5$ $\mathbf{P^h_n} = (P^{\mathrm{h}}_{t-n},\ P^{\mathrm{h}}_{t-n+1},\ P^{\mathrm{h}}_{t-n+2},\ \ldots,\ P^{\mathrm{h}}_t)$ $\mathbf{P^l_n} = (P^{\mathrm{l}}_{t-n},\ P^{\mathrm{l}}_{t-n+1},\ P^{\mathrm{l}}_{t-n+2},\ \ldots,\ P^{\mathrm{l}}_t)$
Momentum and oscillation indicators:		
$MACD_t(s, f)$	Moving average convergence divergence: difference between two moving averages of different periods (s, f) where s stands for a slow period and f for a fast period. $MACD_t(s, f)$ is regularly calculated using 26 (s) and 12 (f) periods.	$\mathbf{EMA}_t(P^{\mathrm{c}}, s) - \mathbf{EMA}_t(P^{\mathrm{c}}, f)$ where s=26, and f=12.
$MACDS_t(s, f, n)$	MACD signal line: moving average of $MACD_t(s, f)$ of past n periods. A buy (sell) signal is generated when the $MACD_t(s, f)$ crosses above (below) the signal line or a threshold.	$\mathbf{EMA}_t(MACD_t(s, f), n)$ where f=12, n=9, and s=26.
$MACDH_t(n, l)$	MACD histogram: difference between the fast MACD line and the MACD signal line.	$\mathbf{EMA}_t(c, l, \lambda) - MACDS_t(n)$ $where\ f = 26$
$RSI_t(n)$	Relative strength index: compares the days that stock prices finish up against those periods that stock prices finish down. Technical analysts calculate this indicator using 9, 14 or 25 periods. A buy signal is when $RSI_t(n)$ crosses below a lower band of 30 (oversold) and a sell signal when $RSI_t(n)$ crosses above an upper band of 70 (overbought).	$100 - \dfrac{100}{1 + \dfrac{\mathbf{SMA}_t(\mathbf{P^{up}_n}, n)}{\mathbf{SMA}_t(\mathbf{P^{dn}_n}, n)}}$ where n = 5, and n is the length of the time series $P^{\mathrm{up}}_t = \begin{cases} P^{\mathrm{c}}_t & \text{if } P^{\mathrm{c}}_t > P^{\mathrm{c}}_{t-1} \\ \text{empty} & Otherwise \end{cases}$ $P^{\mathrm{dn}}_t = \begin{cases} P^{\mathrm{c}}_t & \text{if } P^{\mathrm{c}}_t < P^{\mathrm{c}}_{t-1} \\ \text{empty} & Otherwise \end{cases}$ $\mathbf{P^{up}_n} = (P^{\mathrm{up}}_{t-n},\ P^{\mathrm{up}}_{t-n+1},\ P^{\mathrm{up}}_{t-n+2},\ \ldots,\ P^{\mathrm{up}}_t)$ $\mathbf{P^{dn}_n} = (P^{\mathrm{dn}}_{t-n},\ P^{\mathrm{dn}}_{t-n+1},\ P^{\mathrm{dn}}_{t-n+2},\ \ldots,\ P^{\mathrm{dn}}_t)$
Stochastic oscillator:	Compares close price to a price range in a given period to establish if market is moving to higher or lower levels or is just in the middle. The oscillator indicators are:	
$FAST\%K_t(n)$	Percent measure of the last close price in relation to the highest high and lowest low of the last n periods (true range). Typically a period (n) of 5 is used for $FAST\%K_t(n)$ and 3 for the rest of stochastic indicators. We follow this convention.	$\frac{P^{\mathrm{uc}}_t - min(\mathbf{P^l_n})}{max(\mathbf{P^h_n}) - min(\mathbf{P^l_n})}$

	Vector with low prices of last n periods	$\mathbf{P^l_n} = (P^{\mathrm{l}}_{t-n},\ P^{\mathrm{l}}_{t-n+1},\ P^{\mathrm{l}}_{t-n+2},\ \ldots,\ P^{\mathrm{l}}_{t})$
	Vector with high prices of last n periods	$\mathbf{P^h_n} = (P^{\mathrm{h}}_{t-n},\ P^{\mathrm{h}}_{t-n+1},\ P^{\mathrm{h}}_{t-n+2},\ \ldots,\ P^{\mathrm{h}}_{t})$
$FAST\%D_t(n)$	Moving average of $FAST\%K_t(n)$.	$\mathbf{SMA}_t(FAST\%K_t(n), 3)$
$SLOW\%K_t(n)$	Identically calculated to $FAST\%D_t(n)$ using a 3-period moving average of $FAST\%K_t(n)$.	$\mathbf{SMA}_t(FAST\%K_t(n), 3)$
$SLOW\%D_t(n)$	Moving average of $SLOW\%K_t(n)$. Typically a period of 3 is used. A buy (sell) signal is generated when any oscillator (either $\%K$ or $\%D$) crosses below (above) a threshold and then crosses above (below) the same threshold. Typically a threshold of 80 is used for the above threshold, and 20 for the below threshold. Buy and sell signal are also generated when $FAST\%K_t(n)$ or $SLOW\%K_t(n)$ crosses above or below $FAST\%D_t(n)$ or $SLOW\%D_t(n)$ respectively.	$\mathbf{SMA}_t(SLOW\%K_t(n), 3)$
$MFI_t(n)$	Money flow index: measures the strength of money flow (MF_t) in and out of a stock. At difference of the $RSI_t(n)$ which is calculated using stock prices, $MFI_t(n)$ is calculated using volume. When $MFI_t(n)$ crosses above (below) 70 (30), this is a sign that the market is overbought (oversold).	$100 - \dfrac{100}{1 + \dfrac{PMF_t(n)}{NMF_t(n)}}$ $where\ n = 15$ $MF_t = P_t^{\mathrm{typ}} \cdot VOL_t$ $PMF_t(n) = \mathbf{SMA}_t(MF_t, n)$when $MF_t > 0$ $NMF_t(n) = \mathbf{SMA}_t(MF_t, n)$ when $MF_t < 0$ VOL_t is volume of day t $PMF_t(n)$ is positive money flow $NMF_t(n)$ is negative money flow