A strategy for stock trading based on multiple models and trading rules

CS 395T Project Report
Subramanian Ramamoorthy
December 5, 2003

1. Introduction

The stock market represents a very interesting dynamical system that has intrigued analysts from a number of disciplines. A recent trend among the machine learning community is to build automated agent programs that trade in this market in a relatively autonomous fashion. This is part of a much larger program of research into autonomous agent systems [1].

This report is an account of the design of one such agent. This agent is designed to function within the framework of the Penn-Lehman Automated Trading Project (PLAT) [2]. The details of PLAT are not discussed here. For details on the system architecture and design, the reader is referred to the paper [2], and the PLAT project web site [3].

The structure of the game is essentially as follows. The PXS (Penn Exchange Simulator) server provides an electronic exchange for Microsoft shares, ticker symbol MSFT. This electronic exchange is a hybrid of a real world electronic crossing network (ECN) within NASDAQ and an internal market of agents such as mine. The agents are software programs written to communicate in a client-server protocol with the PXS server. Each agent has the option of invoking various actions such as “Buy” and “Sell”, has control over price and volume of trades and has access to various indicators of both the internal (simulator) and the external (Island ECN) markets. The objective of the agent design is to design an automated strategy that can satisfy the following key requirements:

1. Maximize profits. Specifically, maximize expected return in a strict statistical sense, measured by the Sharpe ratio, to be defined below
2. Unwind (i.e., bring to zero) the entire share position at the end of the trading day. Failure to do so results in a penalty. Any excess shares held at the end of the trading day are valued at zero. Shares that are ‘short’, i.e., a negative holding, needs to be bought back by the agent at twice the closing price.
3. Take into account transaction costs, i.e., a fee on every share traded.

The reader may note that there are clearly multiple competing objectives at play here. The need to maximize profits by taking extreme positions is countered by the need to unwind one’s position at the end of the trading day. Moreover, one needs to maximize profits in a “robust” way so as to maximize a statistical measure of performance, the Sharpe ratio, defined as the ratio of the average of the daily returns and the standard deviation of daily returns. Daily returns are defined as:

Return, R = Cash – Unwinding Penalty – Trading Fees + Trading Rebates

The average daily return is,

$$\overline{R} = \frac{1}{N}\sum_{i=1}^{N} R_i$$

The standard deviation of returns is,

$$\sigma = \left(\frac{1}{N-1}\sum_{i=1}^{N}\left(R_i - \overline{R}\right)^2\right)^{\frac{1}{2}}$$

The Sharpe ratio is,

$$SR = \frac{\overline{R}}{\sigma}$$

The key point to note from this formulation is that it is desirable to make consistent small profits, as opposed to adopting strategies that provide very high returns but also result in very large variance, neutralizing the benefits of the large returns. In this sense, the need is for a *controlled* trading strategy.

2. Brief survey of the literature

There exists an immense body of work on the mathematical analysis of the behavior of stock prices, stock markets and successful strategies for trading in these environments. In recent times, a variety of different approaches have been tried for achieving the goals outlined in section 1. Some representative examples from the machine learning perspective would include rule induction and genetic algorithms [4,5], neural networks [6], reinforcement learning [7], etc. While these are powerful techniques, they also require extensive data handling and processing and the use of computational power.

Apart from these sophisticated approaches, a popular approach among practitioners is the use of technical trading rules. These techniques assume that, notwithstanding the efficient market hypothesis, there exist patterns in stock returns and that they can be exploited by analysis of the history of stock prices, returns and other key indicators. The reader is referred to [8] for more details on technical analysis for stock trading. The paper, [9], is a very good description of why it is a suitable tool, notwithstanding some controversy regarding their utility, for the problem at hand.

3. An 'intuitive' approach to stock trading

My research for this project is centered on the notion of using 'intuitive' strategies, i.e., that appeal to "common sense" and embody assumptions similar to those held by human traders. The final working strategy that resulted from my research is a robust strategy designed by composing multiple 'intuitive' strategies into a workable global strategy. The hypothesis is that robustness and relatively complex global behaviors are achievable by synthesizing multiple, intuitively obvious and structurally well defined local behaviors.

A simple intuition behind stock trading is summarized by the dictum "Buy Low, Sell High". Essentially, if one bought a share at a 'low' price p and then sold it at a 'high' price $p+\delta$, one makes a profit of δ. A consistent series of such trades would quickly accumulate wealth for the trader.

Clearly, the problem in real stock markets is that one never really knows the future and the decision of when to buy in anticipation of a future price increase is a difficult one. This decision is further complicated by the fact that there is a strict penalty for not unwinding one's position. So, if one were to take an extreme position in anticipation of a future price increase and such an increase were to never materialize, one takes a big loss for that trading day.

A working assumption that helps alleviate this problem is that stock returns are mean reverting. This is an assumption that is made often in the financial theory literature, and is supported by some empirical evidence. From the perspective of the current problem, one may assume that stock prices are mean reverting and so it is possible to trade now in expectation of a future movement in the opposite direction (In later sections, we will examine how to deal with the violation of this assumption). A simple trading strategy based on this assumption is as shown below.

Algorithm 1:
MA_1 = Moving Average of Stock Price over a horizon N_1
MA_2 = Moving Average of Stock Price over a horizon N_2 ($N_2 > N_1$)
Begin Loop:
Calculate MA_1, MA_2 using the Last Price from Island ECN
If($MA_1 > MA_2$ + Threshold)
    Then *Sell* m shares @ SellPrice
Else If ($MA_2 > MA_1$ + Threshold)
    Then *Buy* m shares @ BuyPrice
Else
    Do Nothing
End Loop.

The selection of individual variables in this algorithm will be addressed in a future section.

The need to unwind positions is clearly not included in this algorithm. A preliminary approach to this problem is to decide on a fixed window before closing time and to divest all of one's holdings.

Including this strategy, we get a modified version of the above algorithm as:

Algorithm 1.1
If (time < window)
    Then **Algorithm 1**
Else
    If(Number of Shares > 0)

Then *Sell* m shares
Else
Buy m shares
m: A fraction of the current share holdings

This algorithm would attempt to end with zero holdings and during the active trading period, it can be expected to make gains.

This algorithm is an attempt at a simple trend based strategy. In actual experiments, this strategy was found to be capable of making large gains, but it was equally likely to take large losses. The figures below show typical plots of results obtained with this algorithm, when the return is favorable.

Ram's client

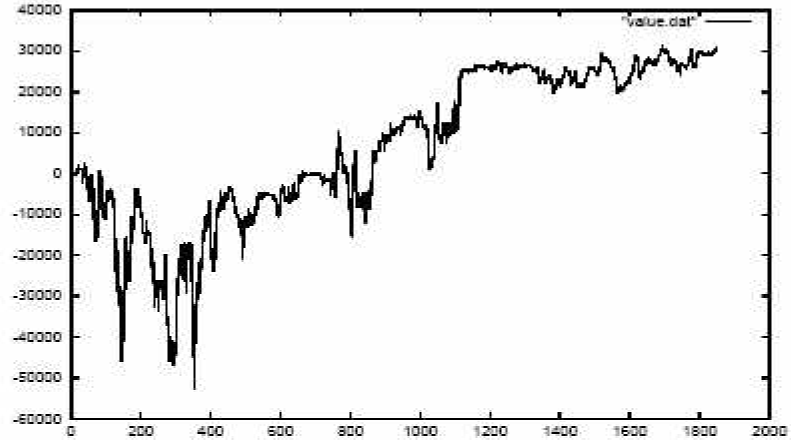


Figure 1: Value of the shares traded for the day

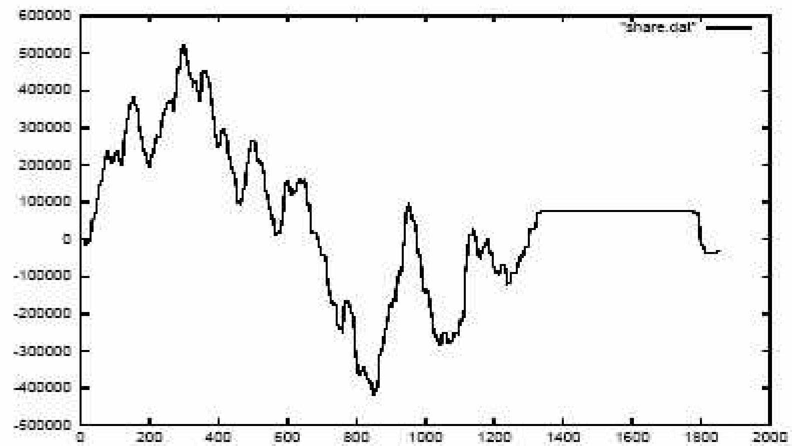


Figure 2: Volume of Shares traded for the day

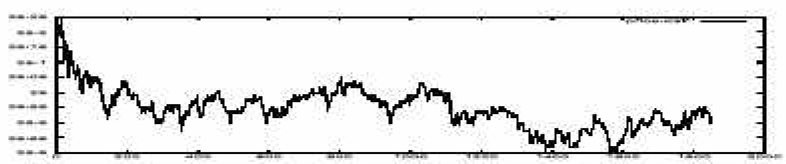


Figure 3: Price of a MSFT share for the day

These plots illustrate the following key points:

- The stock price over the trading day showed both positive and negative fluctuations. In response, the agent took both positive and negative positions and unwound successfully, ending in a profitable position.
- The trend was very favorable in that positive holdings were followed by an increasing trend, resulting in profits.

A key point to note is that the return is largely dependent on the nature of the trends. A favorable trend would cause the agent to take a strong position and divest at a profitable rate. On the other hand, a move to the other direction would leave the agent exposed and it would take large losses. As it stands, the result is entirely dependent upon the nature of the trend.

The following figures illustrate this aspect of the algorithm.

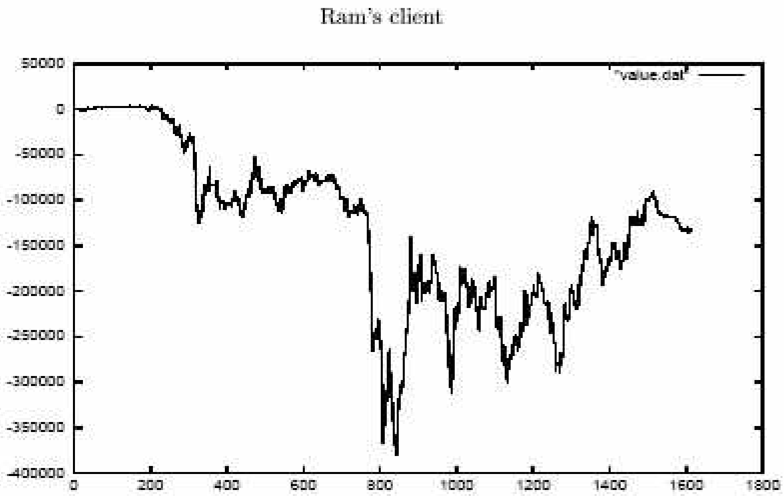


Figure 1: Value of the shares traded for the day

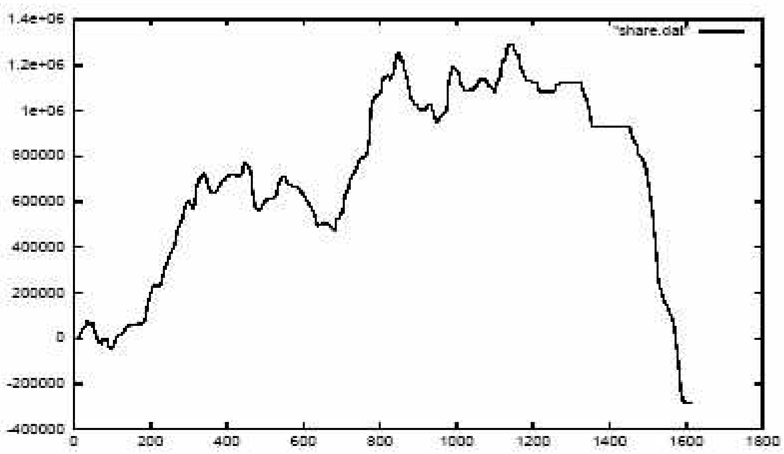


Figure 2: Volume of Shares traded for the day

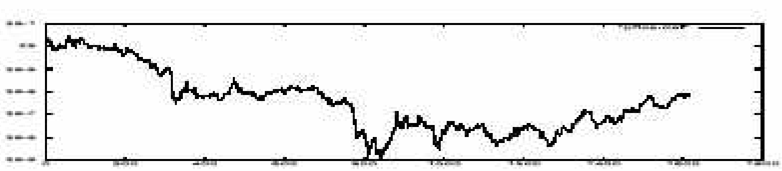


Figure 3: Price of a MSFT share for the day

Note that the agent took a strong positive position in response to a decreasing trend. However, the subsequent trend flattened out and the investment was not favorably recouped.

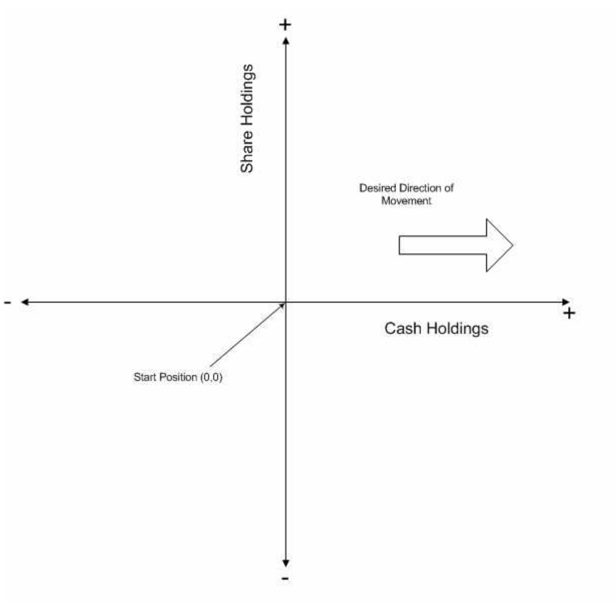
What this suggests is that a naïve dependence on the mean reverting nature of stock prices and the use of drifts and trends is susceptible to a lot of variation.

4. A Multiple Model Algorithm

As pointed out in the previous section and in [2], it is not sufficient to trade on the basis of simple trends. The trends do not always have nice properties to aid the profitability of the agent.

To overcome this problem, I adopted a multiple model strategy. The intuition behind this strategy is that there are periods of time when the behavior of the stock return is, in fact, mean reverting and Algorithm 1 in section 3 would, in a statistical sense, produce profits. When the markets deviate from this favorable model, the resulting effect would be observed from instantaneous cash and stock holdings. This could be used to trigger a mode switch to a different strategy that does not assume the mean-reverting nature of stock prices.

I propose that the problem of detecting the agent and the market's mode can be solved by thinking in terms of two key variables – cash held by the agent, net shares held by the agent. This representation can be visualized as a two dimensional state space, as shown below.



The idea is to move in the positive direction on the Cash axis while trying to stay close to zero on the Share axis. The effect of the market is to move the agent's current position along the negative Cash axis. The agent, on its part, can issue commands to move along the Share axis. In this sense, the act of trading is similar to a game against nature where game is played iteratively until the end of trading. Risk is represented by the agent's state being far from the Share axis. The correct response action, as viewed in this state space, would bring the agent down towards the Share axis and towards positive Cash holdings, a position of very low risk. So, the idea is to explicitly control risk and rewards by tuning the one variable available to the agent, the share holdings, in response to observed variables, the last price. In my work, I did not explicitly solve a noncoperative game to calculate the appropriate action, mostly because the payoff functions are hard to estimate and it is not clear what the 'utility' of the opponent would be modeled as in this game against nature. However, that is an avenue for possible future investigations and improvements.

With this representation of the problem domain, one can synthesize global behavior based on multiple local behaviors, listed below.

<u>Regular:</u> Perform regular trading, as proposed in Algorithm 1.

<u>Safe:</u> Try to divest holdings when profitable, otherwise do nothing, as shown in algorithm below. This strategy would never increase holdings in the unfavorable direction (by definition). So, it is safe in the sense that if there is a trend that has been modeled badly, then this algorithm attempts to enfore a change of course in the state space trajectory.

Algorithm 2:
MA_1 = Moving Average of Stock Price over a horizon N_1
MA_2 = Moving Average of Stock Price over a horizon N_2 ($N_2 > N_1$)
Begin Loop:
Calculate MA_1, MA_2 using the Last Price from Island ECN
If (# Shares > 0)
    If($MA_1 > MA_2$ + Threshold)
        Then *Sell* m shares @ SellPrice
If (# Shares < 0)
    If ($MA_2 > MA_1$ + Threshold)
        Then *Buy* m shares @ BuyPrice
Else
    Do Nothing
End Loop.

<u>Risk seeking:</u> Trade with lower margins and larger volumes in expectation of higher returns (Same as Algorithm 1 - except for increased volume, m, and a lower threshold).

These behaviors may be composed as shown in the table below:

Cash Holdings

Share Holdings	Very negative	Negative	Zero	Positive	Very Positive
Very Short	Safe	Safe	Safe	Safe	Safe
Short	Safe	Safe	Regular	Regular	Safe
Zero	Safe	Regular	Regular	Regular	Risk seeking
Long	Safe	Regular	Regular	Risk seeking	Risk seeking
Very Long	Safe	Safe	Safe	Safe	Safe

From the state space plot, we know that the goal is to move towards the state (zero shares, very positive cash). Is this possible from this composition?

To answer this question, let us look at the qualitative behaviors of each action.

<u>Regular:</u> This is the basic mode described earlier. As long as the market is well behaved in the sense of mean reverting operation, the qualitative property is to generate cash, and hence move to the right hand side of the state space mentioned earlier. A strong trend in a single direction would force this agent to take a strong short or long position. However, this would trigger a mode switch to the Safe mode (whose qualitative behavior is described below) and this inhibits unbounded losses due to the agent's actions.

Note that, by using this multiple model strategy, I am not claiming to mitigate the adverse effect of an 'act of nature' such as a market crash. The agent is merely attempting to take the best possible action, given an assumed action space (which is typically derived from company guidelines for a human trader).

<u>Safe:</u> This action is defined in such a way that the holdings are always reduced. If the current position were long, shares will be sold, and the movement would be towards the zero shares row. In the process, cash will be generated and the movement will be along the positive cash axis. This is as desired. If the initial position were Short, one should have the proceeds of the sales, so one be in the upper rows and the right action would be to use the cash to buy shares and proceed towards the Zero shares row where a regular action is prescribed. In essence, the qualitative behavior is along the "arrow" shown in state space. Note that an exceptional position would be (Very short and very negative and neighboring areas. If one sold in anticipation of future gains and does not have the proceeds of the sale, the strategy is really misbehaving. It is hard to suggest an appropriate action to recover from this condition. So, I choose to adopt the Safe strategy and at least divest all holdings in such situations.)

<u>Risk Seeking:</u> When in this mode, the qualitative behavior is the same as in the Regular mode, where mean reverting property is assumed. The difference is that larger volumes are traded at lower margins. This implies a certain risk seeking behavior, as discussed earlier. If the result of this risk seeking were that the cash position were jeopardized, the agent would find itself in a Regular or safe mode and the trading parameters would

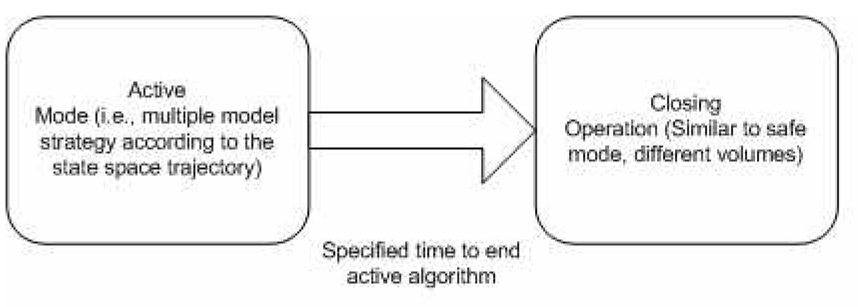
become more conservative in response, preventing further damage. If the risk seeking behavior finds a favorable market, then the influx of revenues will be significantly improved.

In a broad sense, what is being implemented here is a control strategy that bears some resemblance to fuzzy logic and other qualitative multiple model strategies [10]. Clearly, the models are less well understood in this domain, and hence the properties of the various local models are heuristically based rather than rigorously derived. Formalizing these notions is an area for future work and research.

Saying that this is a control strategy based on instantaneous observations implies that each decision is made in a Markovian fashion, with dependence only on current state. This is not really true if there are unexecuted orders that may again be activated when price fluctuates over time. To enforce this Markovian nature, the implementation includes code to explicitly withdraw all orders at the start of every iteration of the trading cycle.

Another interesting aspect of the strategy, related to actual implementation, is the fact that while the above description addresses the question of what an agent should choose to do, it does not address whether that action can actually be achieved. For instance, just because a Sell order is issued does not imply that it will be filled. In this scenario, I make a decision on the appropriate action using the above scheme, which is based on a measurement of 'last price' in the Island ECN. However, in order to execute this action, I look at the order books to find the best existing offer and place orders at a value that is incrementally better than that order (typically, this "undercut" value is 0.00001 – small enough to not make a difference in price but sufficient to enforce a trade by getting ahead of existing orders).

There is also an explicit action that kicks in towards the end of the trading day. This Closing mode is similar to the Safe mode, except that the volume of trades is increased to enforce full divestment. So, the high level mode switch is represented as follows and is triggered by a user specified time. Note that even the Closing mode is essentially safe in the sense that the agent will try to sell at opportune moments in response to trends. However, to enforce divestment, the thresholds are reduced and volumes increased. The Active mode enforces the mode switching logic that was shown in the table above. Once Closing mode is invoked, only one single type of trading behavior is adopted. In this sense, there are two levels of mode switching in my agent.



5. Experimental Results

The following figures are typical results obtained from using this multiple model strategy (corresponding to Dec 1, 2 and 3 respectively). In this experiment, the multiple model strategy was used in an economy against an agent using Static Order Book Imbalance (SOBI) strategy [3] and a version of an agent based on genetic algorithm optimization.

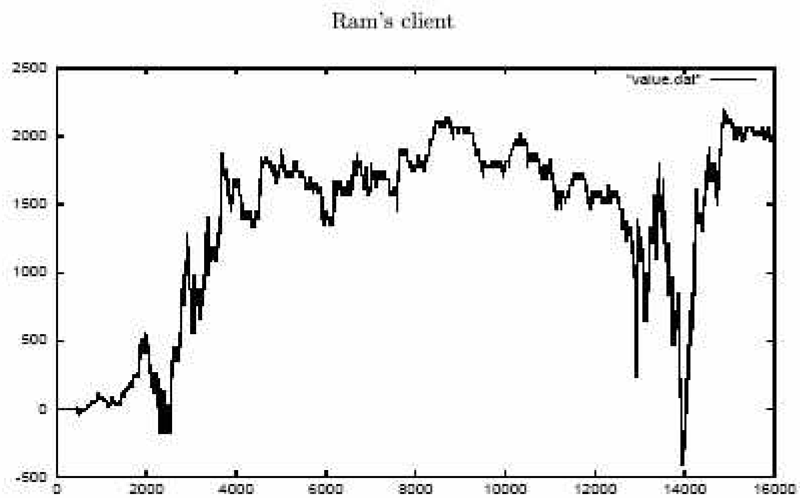


Figure 1: Value of the shares traded for the day

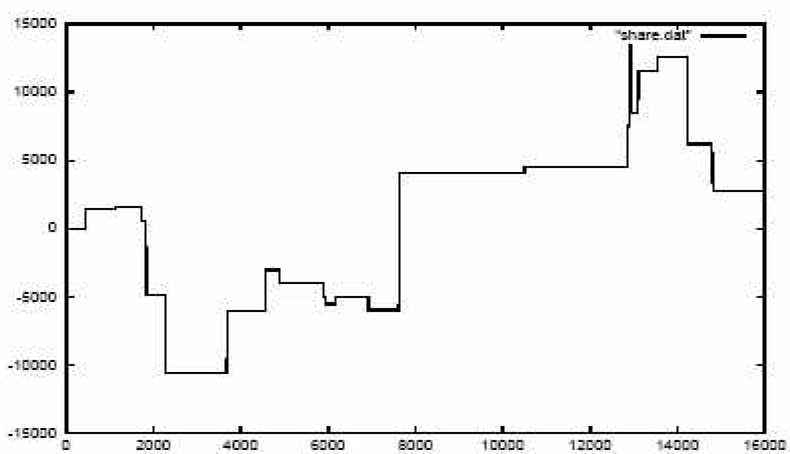


Figure 2: Volume of Shares traded for the day

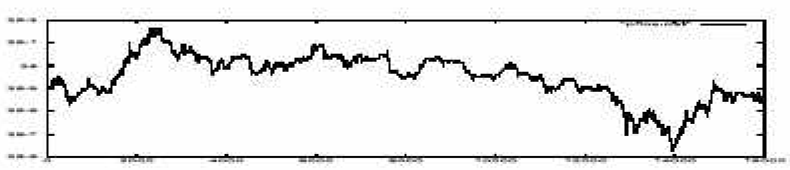


Figure 3: Price of a MSFT share for the day

Ram's client

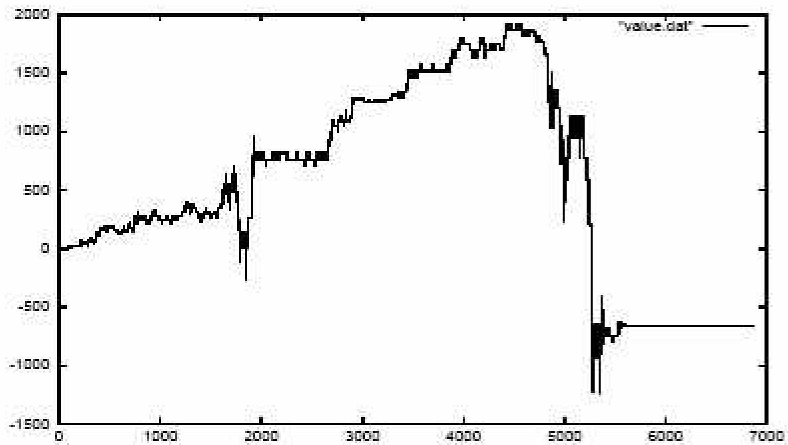


Figure 1: Value of the shares traded for the day

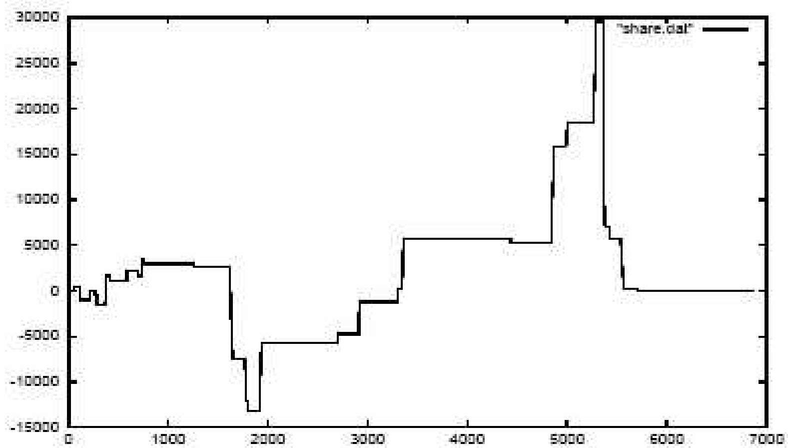


Figure 2: Volume of Shares traded for the day

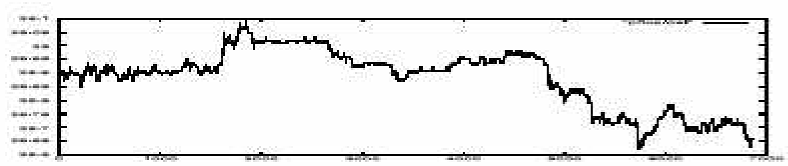


Figure 3: Price of a MSFT share for the day

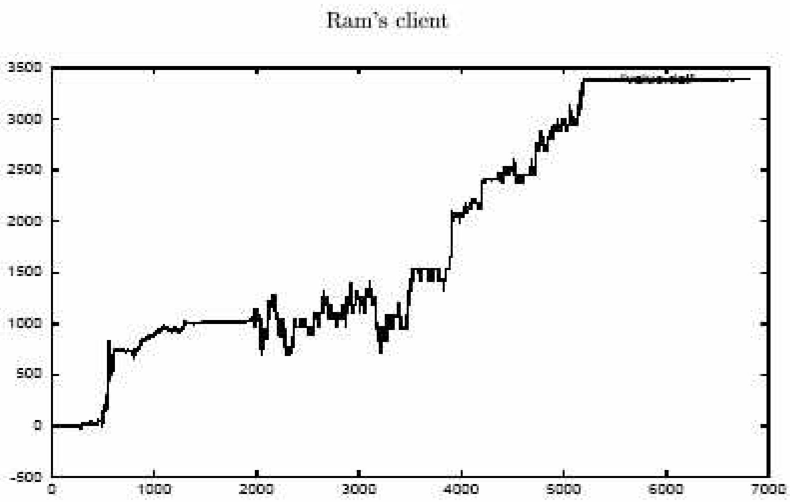


Figure 1: Value of the shares traded for the day

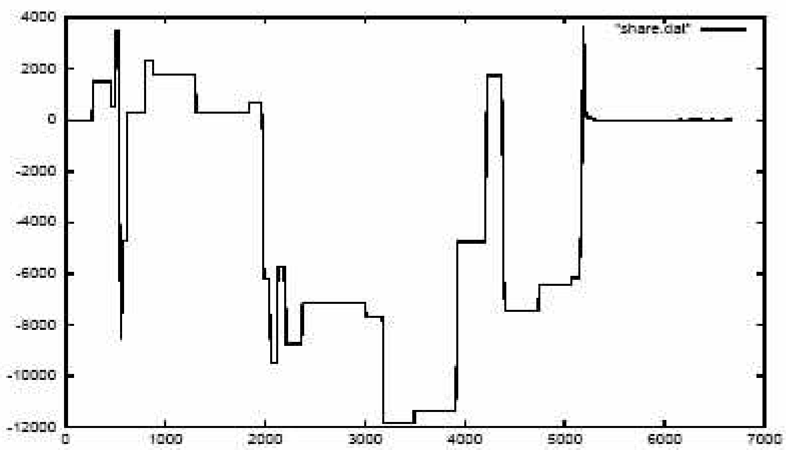


Figure 2: Volume of Shares traded for the day

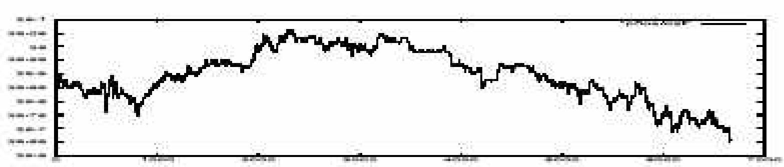


Figure 3: Price of a MSFT share for the day

Some key observations based on these results are:

1. The overall volume of trading is much lower than in earlier trials. This was found necessary due to the strict penalties for not unwinding one's position at the end of the day.
2. Both the holdings and the returns are within controlled (and user specified) bounds. This is an important ingredient of the strategy and while it leads to the

resulting low trade volumes and profits, it aids the achievement of consistency and hence higher Sharpe ratio.

3. A key variable in this algorithm is the selection of the threshold on the difference between two moving averages that triggers a trade. This threshold has a significant impact on profitability. Setting it too low (in my experiments, ~ 0.005), while sufficient to offset transaction costs, is not sufficient to stay profitable. My hypothesis here is that this value refers to a time scale in the tick by tick data at which the moving averages are essentially responding to noise in the signals. I found that a much higher value (~0.01) nicely captures the true trend that we wish the agent to follow.

An observation that came out of the experiments is that there is a difference between the live and historical modes in terms of the speed of execution. If a strategy were tuned in live mode using a certain window size then in a faster historical run this strategy would perceive the gradient of price differently and the thresholds would be incorrectly set. Principled ways of handling these differences would be a useful direction of investigation for the future.

The following is the result of a set of experiments on a week of historical data (Dec 1-5) to determine the Sharpe ratio achievable by the proposed algorithm in comparison to other competitive trading agents. These results are taken from historical mode simulations. The other agents use a variety of strategies ranging from mining message board postings to reinforcement learning and genetic algorithm based optimization. For comparison purposes, I report my scores and scores from one other agent (the winner in the class contest) that uses genetic algorithm optimization. The two Sharpe ratios are also shown below.

Date	Value – Multiple Model Strategy	Value – GA Based Strategy
Dec 1, 2003	-5922.9	132.792
Dec 2, 2003	2909.4	-1175.024
Dec 3, 2003	3801.2	711.03
Dec 4, 2003	849.9535	3188.856
Dec 5, 2003	-399.236	1566.149

Sharpe Ratio (MM) = 0.0647
Sharpe Ratio (GA) = 0.5432

It is clear from these results that while the MM strategy is capable of generating revenue in a statistically significant sense it does have a fair amount of variation compared to a more sophisticated learning algorithm. While the proposed strategy has demonstrated some desirable qualities, it suffers from many weaknesses. Firstly, the parameters such as time scale for setting the threshold, the time to start closing mode and the volume to trade at closing mode, are empirically tuned after numerous experiments. In an ideal implementation, these parameters should constitute an adaptation algorithm for the agent to be truly autonomous over larger time frames. Secondly, the moving average based trading rule was a simple rule selected for its intuitive appeal and to explore the question of how much complexity is really required of the local behaviors. Based on the results of the preliminary experiments, it seems like a more powerful prediction mechanism should replace the moving window strategy to eliminate this simplistic predictor. The benefit from a more powerful prediction method (e.g.., nonlinear time series forecasting models) is that the horizons over which one can trust the prediction is larger, there is more differentiation between fine structure of the time variation of price and hence one achieves a more accurate decision of when to invoke an action (Buy/Sell) and with what parameters. It would also be helpful to include more variables for prediction, e.g., volume, fundamental indicators, etc.

6. Conclusions

This project was an attempt to explore a new direction in the design of trading strategies – the use of multiple 'intuitive' trading rules composed to achieve a more sophisticated global behavior. As argued in the previous section, there is evidence that this has been achieved to an extent. The agent had demonstrated the ability to provide profitable operation in a statistically significant sense. It has done so while actively satisfying constraints on risk and cash outlays.

This simple experiment can be greatly extended in future research. While the notion of multiple models and the control strategy based on phase space decomposition has behaved as expected, the trading rules themselves can be vastly improved. One approach might be to replace the moving windows by more powerful predictors such as Kalman filters and nonlinear time series models. This will also enable precise determinations of profitable settings for thresholds, e.g., based on the covariance values in Kalman filters. An interesting empirical finding in these experiments concerns the time scales at which trends need to be followed. It would be interesting to understand this in a formal sense. Lastly, the boundaries between various local regions were empirically determined and tuned. It would help to analyze the process formally and identify rules for this composition.

As a closing remark, this effort has demonstrated an effective method for problem decomposition. Within this space, there is plenty of room for the use of machine learning strategies. Each parameter mentioned in the algorithms needs to be constantly retuned in a nonstationary environment. In practice, intricate details such as the appropriate thresholds for triggers and timescales are best inferred autonomously by analyzing the online data. This is an important area for future work.

7. References

1. A. Greenwald, N.R. Jennings, P. Stone, Agents and Markets, Guest Editor's Introduction, 18(6), pp 12 – 14, Nov/Dec 2003.
2. M. Kearns, L. Ortiz, The Penn-Lehman Automated Trading project, IEEE Intelligent Systems, 18(6), pp 22 – 31, Nov/Dec 2003.
3. http://www.cis.upenn.edu/~mkearns/projects/pat.html

4. G.H. John, P. Miller, R. Kerber, Stock selection using rule induction, IEEE Expert, pp 52 – 58, Oct 1996.
5. F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, J. Financial Economics, Vol 51, pp 245 – 271, 1999.
6. A-P Refenes (ed.), Neural Networks in Capital Markets. Wiley 1995.
7. J. Moody, M. Saffell, Learning to trade via direct reinforcement, IEEE Trans. Neural Networks, 12(4), July 2001.
8. J. Schwager, Technical Analysis. Wiley 1996.
9. W. Brock, J. Lakonishok, B. Le Baron, Simple technical trading rules and the stochastic properties of stock returns, Journal of Finance, 47(5), pp 1731 – 1764, Dec 1992.
10. B. Kuipers and S. Ramamoorthy. Qualitative modeling and heterogeneous control of global system behavior. In C. J. Tomlin and M. R. Greenstreet (Eds.), Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, Springer Verlag, 2002.