Technical Trading Rules as a Prior Knowledge to a Neural Networks Prediction System for the S&P 500 Index

Tim Chenoweth[1,2,3] Zoran Obradović[1] Steven Lee[4]

[1] School of Electrical Engineering and Computer Science
[2] Department of Management and Systems
[3] Department of Economics
Washington State University, Pullman WA 99164-2752

[4] Division of Statistics
University of Idaho, Moscow, ID 83843

***Abstract*— Financial markets data is very noise and non-stationary which makes modeling through machine learning from historical information a challenging problem. Our experience indicates that in markets modeling through neural network learning, significant data preprocessing is needed. We have recently proposed a promising multi-component prediction system for the S&P 500 index which yields a higher return with fewer trades as compared to a neural network predictor alone. The multicomponent system consists of a statistical feature selection, a simple data filtering, two specialized neural networks for extraction of nonlinear relationships from selected data, and a symbolic decision rule base for determining buy/sell recommendations. The objective of this study is to explore if a more sophisticated data filtering process in our multicomponent system leads to further improvements in return or to a reduced number of trades as compared to our current system. The new systems is using some well-known technical trading rules/indicators as a prior symbolic knowledge to develop a directional filter that splits the financial data into up, down, and sideway data sets. We use the directional movement indicators to detect whether the market is trending, and to measure the strength of the trend if it exists. Various experimental results using this system to predict S&P 500 index returns are presented and the result compared to our previously developed multi-component system. The system performance is measured by computing the annual rate of return and the return per trade.**

I. Introduction

In general, most quantitative methods that attempt to predict stock market movements are based on statistical time series models [1]. These paradigms are largely unsuccessful due to the inherent complexity of financial markets in general and the stock market in particular. The efficient market hypotheses says that stock prices adjust to new information very rapidly, usually by the time the information becomes public knowledge, making it impossible for statistical paradigms based on this information to make accurate predictions [5].

While the efficient market hypotheses seems to be correct for static and linear relationships between stock prices and historical information, it is possible that dynamic or nonlinear relationships exist that traditional statistical time series methods are incapable of modeling [5]. If this is true, it may be possible to capture these relationships using a nonparametric machine learning approach of multilayer artificial neural networks (NN). Such NN's are powerful computational systems that can approximate any nonlinear continuous function on a compact domain to any desired degree of accuracy [3]. In addition, a NN can account for fundamental changes in the underlying function through incremental retraining using the back-propagation learning algorithm [7].

This paper focuses on the pattern filter component of our previously proposed system [2]. This component is used to separate the training patterns into three disjoint sets; a training set used by the "up" NN, a training set used by the "down" NN, and a noisy set that is discarded. The original system used a simple filter based on whether the target return for a pattern was greater than or less than zero. The objective of this study is to see if a more sophisticated preprocessing technique of determining the S&P 500 market direction and whether the market is trending or moving sideways improved the overall system performance. For that matter, the directional movement indicator DMI [4, 6] is incorporated into the data filtering component of the system and the results are compared to those achieved using the simple filtering approach. Section II describes the system with details of the two filtering techniques discussed in Section III. Experimental results are presented in Section IV and conclusions in Section V.

II. Multi-Component System Architecture

Recently, promising results are obtained by incorporating two specialized neural networks into a hybrid multi – component nonlinear system for S&P 500 stock market predictions [2]. The system uses a filtering component for identification of the most relevant patterns for two specialized NN's trained to predict stock market returns. A high level decision rule is used for determining buy/sell recommendations as a function of the two predictions obtained from the "up" and "down" NN's.

This work was supported in part by the National Science Foundation under grant IRI-9308523 to Z. Obradović.

ISBN# 0-7803-2639-3

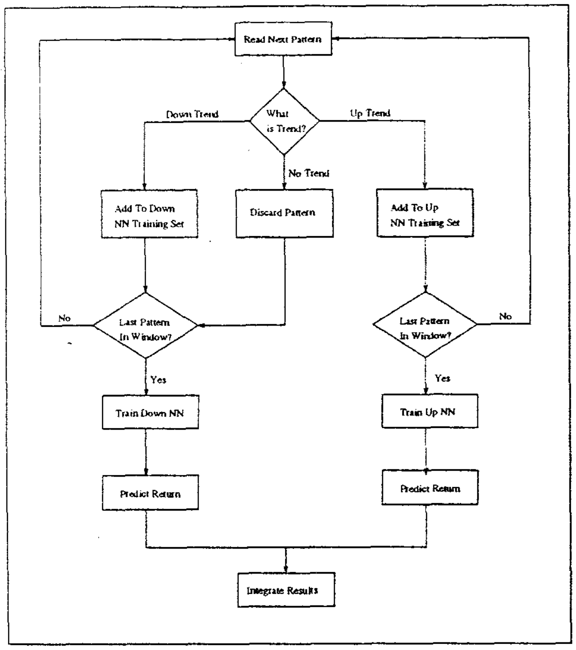

Figure 1: System Architecture.

A. Return Rate Prediction

The return rate prediction component, shown in Fig. 1, consists of two NN's (an "up" network and a "down" network) that are trained using the back-propagation algorithm. Two distinct filtering schemes used to separate the training patterns into "up trend" and "down trend" patterns are tested. The first directional filter is a previously used simple approach that separates the training patterns according to the sign of the target return for a specific pattern. The second directional filter is computing the direction movement indicator (DMI) to determine market direction and how strong the market is trending. Patterns are designated to the "up" and "down" networks based on which direction the market is trending. Section III describes the details for both schemes. Once both NN's are trained, the test pattern is presented to each and the corresponding predictions are collected. A decision rule base is applied to these predictions and a buy/sell recommendation made as explained in Section III.B.

The learning scheme consists of a sequence of training and prediction sessions where the NN's are retrained after each session using more resent information while the older information is discarded. This is achieved by training the NN's using patterns from a fixed size window covering a continuous time segment of historic data. The target return for the time unit immediately following the window is predicted by both NN's and the predictions used by the rule base. Then the training window is shifted forward one time unit (i.e., one trading day), the patterns from the new window used to retrain the NN's, and a prediction made for the next time unit. This process is repeated until the data set is exhausted.

B. Predicted Returns Integration

The predicted returns from both NN components are used as input to the rule based integration component (see Fig. 1). This component analyzes the predicted returns and outputs a buy/sell recommendation that is used to establish either a long or short position in the market. A long position means purchasing an asset for later resale, while a short position means selling a borrowed asset now and purchasing it later.

The rule used in this study is an extension of the "buy and hold" strategy in that if the system does not have a recommendation a long position is established. The decision rule first compares the "up NN" prediction r_u to the "down NN" prediction r_d and recommends a long position in the market if $r_u > 0$ and $r_d \geq 0$, and a short position if $r_u \leq 0$ and $r_d < 0$. Otherwise the decision rule computes the normalized difference *diff* as

$$diff = \frac{\max\{|r_u|, |r_d|\} - \min\{|r_u|, |r_d|\}}{\max\{|r_u|, |r_d|\}},$$

compares this ratio to a predefined decision threshold value y, and determines a buy/sell recommendation as follows. If $r_u > 0$, $r_d < 0$, $diff > y$, and $|r_u| < |r_d|$, the system recommends a short position. Otherwise the recommendation is to take a long position.

III. Pattern Filtering Schemes

A. Previous Directional Filter

The original system used a simple filtering approach where for each training session the target return corresponding to each pattern in the window is compared to a threshold value h_1. If the return is greater than h_1 the corresponding pattern is added to the "up NN" training set, if the return is less than $-h_1$ the pattern is added to the "down NN" training set. Any pattern with a target return between $-h_1$ and h_1 is discarded.

B. Current Directional Filter

The more sophisticated approach uses the directional movement indicator (DMI) to determine market direction and the average direction movement index (ADX), a derivative of the DMI, to determine if the market is trending. As explained in [4, 6], the directional movement calculation (DM) is based on the assumption that, when the trend is up, today's high price should be above yesterday's high. Conversely, when the trend is down, today's low price should be lower than yesterday's low. The difference between today's high and yesterday's high is the "up" direction movement, or DM^+, and the difference between today's low and yesterday's low is the "down" directional movement, or DM^-. The DM^+ and DM^- are each averaged over k_1 days to obtain $DM^+_{k_1}$ and $DM^-_{k_1}$ values. If the DM^+ is greater than the DM^-, the directional movement is up, otherwise it is down. As the two values diverge, the directional movement increases. The greater the difference between DM^+ and DM^-, the more directional or trending is the market. The ADX is a measure of this difference, which will be used to separate the patterns into "up" and "down" training sets. The details on how to compute ADX are explained in the following section.

For a specific day, if the ADX indicates that the market is moving sideways (ie. there is no trend) then the pattern corresponding to that day is discarded (ie. it is not used in either the "up" or the "down" training sets). If the ADX indicates that the market is trending, then the $\mathrm{DM}_{k_1}^+$ and $\mathrm{DM}_{k_1}^-$ for that day are compared to determine the market direction. If the $\mathrm{DM}_{k_1}^+$ is greater than the $\mathrm{DM}_{k_1}^-$ the market is trending up and the pattern is added to the "up" training set. If the $\mathrm{DM}_{k_1}^-$ is greater than the $\mathrm{DM}_{k_1}^+$ then the market is trending down and the pattern is added to the "down" training set.

Two rules were tested for filtering the data:

- **Filtering rule** $\mathrm{DMI}_1(k_1, k_2, h_2)$ compares today's $\mathrm{ADX}_1(k_1, k_2, h_2)$ to a threshold value h_2. If today's $\mathrm{ADX}_1(k_1, k_2, h_2)$ is less then h_2 or is less than yesterday's $\mathrm{ADX}_1(k_1, k_2, h_2)$ then the pattern is discarded. If today's $\mathrm{ADX}_1(k_1, k_2, h_2)$ is greater than h_2 *and* is larger than yesterday's $\mathrm{ADX}_1(k_1, k_2, h_2)$ then today's $\mathrm{DM}_{k_1}^+$ is compared to today's $\mathrm{DM}_{k_1}^-$. If $\mathrm{DM}_{k_1}^+ > \mathrm{DM}_{k_1}^-$ then the pattern is added to the "up" NN's training set; else add the pattern to the "down" NN's training set.
- **Filtering rule** $\mathrm{DMI}_2(k_1, k_2, h_2)$ again compares today's $\mathrm{ADX}_2(k_1, k_2, h_2)$ to a threshold h_2. If either today's $\mathrm{ADX}_2(k_1, k_2, h_2)$ or yesterday's $\mathrm{ADX}_2(k_1, k_2, h_2)$ is below h_2 then discard the pattern. If today's $\mathrm{ADX}_2(k_1, k_2, h_2)$ is above h_2 for two consecutive days (ie. today and yesterday) then compare today's $\mathrm{DM}_{k_1}^+$ and $\mathrm{DM}_{k_1}^-$ and assign the pattern to the correct training set as in filtering rule 1.

C. Computing ADX

ADX is computed using the following algorithm.

1. Compute directional movement (DM^+ and DM^-) as

$$DM^+ = \begin{cases} \max\{T_h - Y_h, 0\} & \text{if } T_h - Y_h > Y_l - T_l \\ 0 & \text{otherwise} \end{cases}$$

and

$$DM^- = \begin{cases} \max\{Y_l - T_l, 0\} & \text{if } T_h - Y_h < Y_l - T_l \\ 0 & \text{otherwise} \end{cases}$$

where T_h and T_l are today's market high and low values, and Y_h and Y_l are yesterday's market high and low values. It is important to note that every day has both a DM^+ and a DM^-, and that at most one of these two values is positive one is a positive value while the other is zero. For example, suppose today's high and low values are 150 and 100 respectively and yesterday's high and low values are 140 and 105. Since $|150 - 140| > |100 - 105|$ the DM^+ will equal 10 while the DM^- is zero.
2. Compute $\mathrm{DM}_{k_1}^+$ and $\mathrm{DM}_{k_1}^-$ as the average DM^+ and DM^- for the previous k_1 days.
3. Now to derive the ADX first compute the difference between $\mathrm{DM}_{k_1}^+$ and $\mathrm{DM}_{k_1}^-$ and then the sum

$$DM_{dif} = |DM_{k_1}^+ - DM_{k_1}^-|,$$

$$DM_{sum} = DM_{k_1}^+ + DM_{k_1}^-$$

S&P 500 index return
S&P 500 index return lagged one day
S&P 500 index return lagged two days
U.S Treasure Rate lagged 2 months
U.S Treasure Rate lagged 3 months
30 Year Government Bond Rate

Table 1: Features of Each Pattern.

4. Calculate the DX or directional movement index as

$$DX = \frac{DM_{dif}}{DM_{sum}} * 100$$

The 100 normalizes the DX value so it falls between 0 and 100.
5. Finally, compute a moving average of the DX over k_2 previous days to create $\mathrm{ADX}(k_1, k_2)$.

IV. Results and Analysis

A. Data Description

The system described in Section 2 is used for S&P 500 stock market buy/sell recommendations. The historic data used in this experiment is ordered daily financial time series patterns from the period January 1, 1985 to December 31, 1993. Patterns from January 1, 1985 to December 31, 1988 comprised the initial training window, whereas actual predictions were made for patterns from January 1, 1989 to December 31, 1993. Table 1 shows the six features used in this study as a single patter. A discussion of how these features were derived can be found in [2].

B. Performance Measures

The most important criteria when measuring the performance of a stock market prediction model is whether it will make money and how much. Therefore the model's annual rate of return (ARR) is computed as follows

$$ARR = \frac{k}{n} \sum_{i=1}^{n} r_i,$$

where:

n is the total number of trading time units for the experiment;
k is the number of trading time units per year (i.e., 253 for daily trading);
r_i is the rate of return for time unit i.

The sum, $\sum_{i=1}^{n} r_i$, is computed by either adding or subtracting the actual daily returns for the S&P 500 index. If the system recommends a long position, the actual return is added to the sum; if a short position is recommended, the return is subtracted.

It is also important to minimize transaction costs by controlling excessive trading (i.e., a 10% return with 50 trades is more profitable than a 10% return with 100 trades). Therefore the break even transaction cost ($BETC$), which may be viewed as the return per trade, is computed as follows:

$$BETC = \frac{1}{m} \sum_{i=1}^{n} r_i,$$

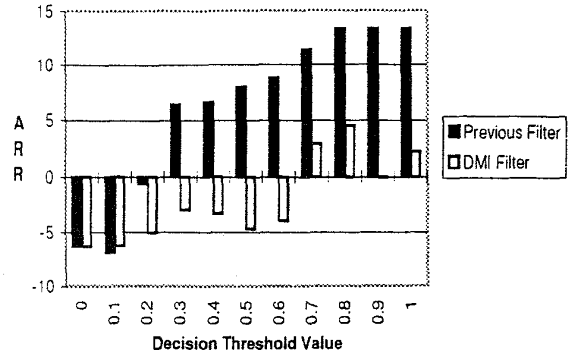

Figure 2: ARR Comparison Between the Previous and $DMI_1(18, 18, 15)$ Directional Filters.

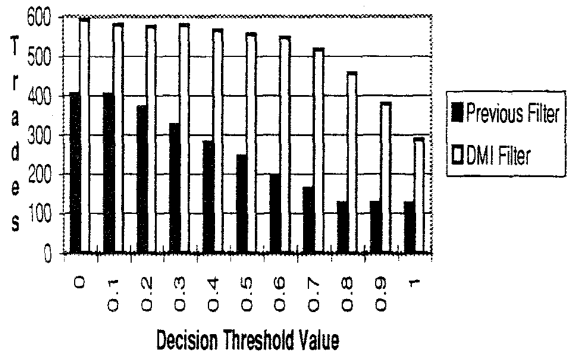

Figure 3: Number of Trades Comparison Between the Previous and $DMI_1(18, 18, 15)$ Directional Filters.

where m is the total number of trading transactions, while r_i and n are defined as previously. A trade is defined as any action that changes a market position. For example, exiting the market constitutes a single trade (i.e., a buy trade to cover a short position or a sell trade to cover a long position), while switching from a short position to a long position constitutes two trades (i.e., one buy trade to cover the short position and another buy to establish the long position).

C. Experiment Description and Results

For the multi-network system, experiments were conducted for various values of the DM smoothing constant k_1, the DX

Parameter	*Value*
Activation Function	Tangent Hyperbolic
Network Topology	6-4-1
Learning Rate	0.03
Tolerance	0.00001
Number of Iterations	5000
Training Window Size Size	1000

Table 2: System Parameter Values.

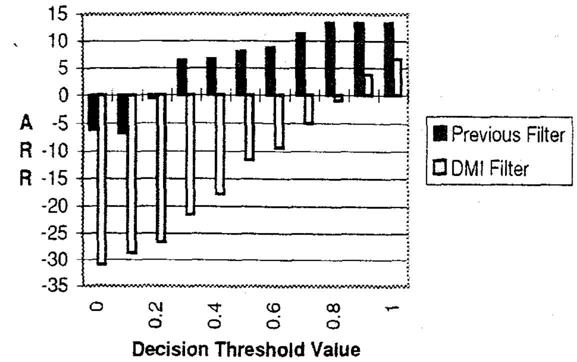

Figure 4: ARR Comparison Between the Previous and the $DMI_1(13, 13, 10)$ Directional Filters.

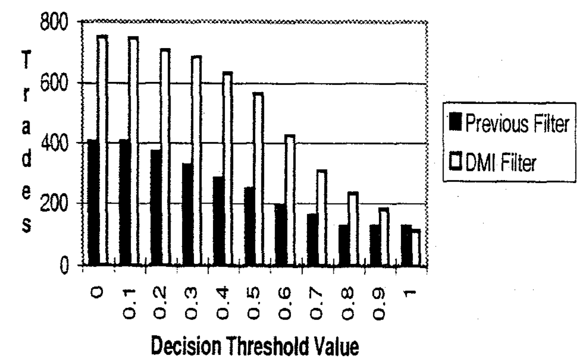

Figure 5: Number of Trades Comparison Between the Previous and the $DMI_1(13, 13, 10)$ Directional Filters.

smoothing constant k_2, and the filter threshold h_2 using the $DMI_1(k_1, k_2, h_2)$ and $DMI_2(k_1, k_2, h_2)$ directional filters. For all experiments, decision threshold y was varied from 0 to 1 in increments of 0.10. The experiments compare the *ARR* and the number of trades using the DMI filters and the best values achieved using the previous directional filter. The best results for the previous system were obtained using filtering threshold h_1 equal to 0.5% with the best *BETC* equal to 0.53%. The return for the buy and hold strategy for the period of this study was 11.56%. Neural network system parameters are displayed in Table 2.

For technical analysis, LeBeau and Lucas recommend using DMI_1(18,18,15) as their results over a variety of data sets indicate that these are the optimal parameter values for manual trading strategies utilizing just the DMI [6]. Therefore, the initial experiments with our trading system focused on using directional filter DMI_1(18,18,15). As can be seen from Figures 2 and 3, the DMI_1(18,18,15) based system achieved a smaller *ARR* using more trades then the system using the previous filtering approach. The best *BETC* achieved by the DMI_1(18,18,15) based system was 0.05% and the best *ARR* was 4.51%.

Other technical analysts actually report using DMI with ADX smoothing parameters k_1 and k_2 in a range of 10 to

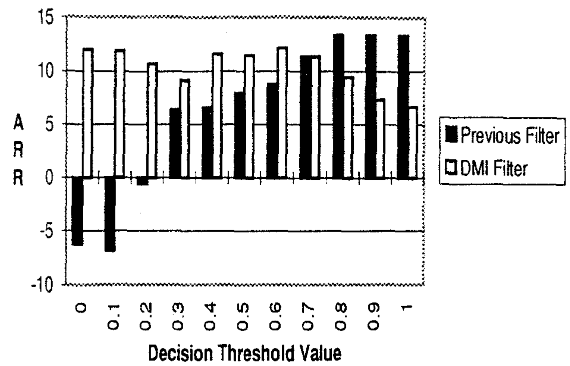

Figure 6: ARR Comparison Between the Previous and the $DMI_2(3,3,65)$ Directional Filters.

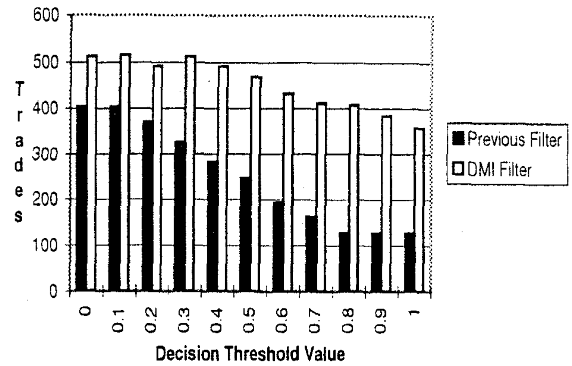

Figure 7: Number of Trades Comparison Between the Previous and the $DMI_2(3,3,65)$ Directional Filters.

20 days. In [4], Elder suggests using ADX(13,13) and so, our trading system experiments shown in Figures 4 and 5 utilized direction filter $DMI_1(13,13,10)$. Experiments using other values for h_2 were conducted, however h_2 equal to 10 achieved the best overall results. The trading system using directional filter $DMI_1(13,13,10)$ performed better than the system using $DMI_1(18,18,15)$. In addition, the system with direction filter $DMI_1(13,13,10)$ achieved it's best ARR using fewer trades than the best previous system. However, both the best ARR (6.56%) and $BETC$ (0.28%) were significantly smaller than those achieved by the best previous system.

A major drawback with using the DMI for NN preprocessing is that the smoothing parameters introduce a lagging problem. This means that there is a delay between the actual beginning of a trend and the moment the DMI identifies it. This results in important patterns being excluded from the NN training sets. To deal with this problem, in the final set of experiments we used ADX(3,3) which uses smaller smoothing parameters and as such reduces the lagging problem. Technical analysts do not seem to be using smoothing parameters this small, as they do not remove enough of the minor market fluctuations. However, in our trading system DMI is used for preprocessing, with prediction made by the NN's, which may be able to distinguish between major trends and minor fluctuations in the market. Figures 6 and 7 show the results obtained using directional filter $DMI_2(3,3,65)$. The best ARR achieved using $DMI_2(3,3,65)$ was 12.14% which is significantly better than the best ARR achieved using $DMI_1(18,18,15)$ or $DMI_1(13,13,15)$, somewhat better than the ARR of the buy and hold strategy, and close to the best ARR achieved by the previous simple system. However, this ARR was achieved with significantly more trades then the previous simple system (432 versus 126 trades). Due to the large number of trades, the $BETC$ of the best $DMI_2(3,3,65)$ based system was only 0.14%, which is smaller than the $BETC$ of the best $DMI_1(13,13,15)$ based system. Additional experiments using directional filters $DMI_2(3,3,h_2)$ with various values for h_2, and $DMI_1(3,3,15)$ resulted in a smaller ARR as compared to directional filter $DMI_2(3,3,45)$ and as such are not reported.

V. Conclusions and Future Research

This study compares a NN based trading system using a simple directional filter preprocessing technique to a more sophisticated preprocessing technique utilizing the DMI to identify trends in the S&P 500 index. The results indicate that the DMI based directional filter used for preprocessing works better with smaller ADX smoothing parameter values. However, the simple preprocessing technique still outperforms the DMI based technique. We believe this is due to the ADX's inability to adjust quickly to sudden changes in the market's direction taking the form of a spike, even when used with smaller values for the smoothing parameters. This problem is particularly evident in "down" trends for S&P 500 index. We are currently analyzing the obtained experimental results by studying the predictive ability of each individual NN in both the previous trading system and the trading system utilizing the DMI directional filter. The objective of this analysis is to determine if the trading system can be improved by an appropriate integration of the simple and the DMI directional filters.

References

[1] Black, F. and Scholes, M., (May-June 1973) "The pricing of Options and Corporate Liabilities," *Journal of Political Economy, Vol. 81.*

[2] Chenoweth, T. and Obradovic, Z., (1995) "A Multicomponent Approach to Stock Market Predictions," *Proc. the Third Int. Conf. on Artificial Intelligence Applications on Wall Street,* New York, NY, pp. 74-79.

[3] Cybenko, G., (1989) "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems,* Vol. 2, pp. 303-314.

[4] Elder, A., (1993) *Trading For a Living,* Traders Press, Inc.

[5] Hutchinson, J., (1993) *A Radial Basis Function Approach to Financial Time Series Analysis,* PhD Thesis, Massachusetts Institute of Technology.

[6] LeBeau, C. and Lucas, D., (1992) *Computer Analysis of the Futures Market,* Traders Press, Inc.

[7] Rumelhart, etc. (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vols. 1 and 2,* MIT Press, Cambridge, MA.