FORECASTING THE S&P 500 INDEX: A COMPARISON OF METHODS

Mary Malliaris and A.G. Malliaris
Quinlan School of Business, Loyola University Chicago, 1 E. Pearson, Chicago, IL 60611
mmallia@luc.edu (312-915-7064), tmallia@luc.edu (312-915-6063)

ABSTRACT

This paper develops and compares several methods of forecasting the S&P 500 Index using only data based on the closing value and trained over a six-decade data set. The methodologies include a C5.0 decision tree, a neural network, and a group of forecasts based on training set patterns of directional change from one to seven days in length. Methods are compared by using the number of correct forecast directions, and by calculating the amount of gain/loss. We find that the neural network yielded the most gain, but the six-day string pattern did best predicting that the Index would move Up.

INTRODUCTION

Many different strategies have been used in predicting the S&P 500 stock index and its behavior. Some models use technical indicators and others add fundamental indicators or economic growth indicators. In data mining techniques, such as decision trees and neural network models, the focus is often on a short period of time with a network structure optimized over that period. The aim of this paper is to develop and compare several methods of forecasting the S&P 500 Index using only data based on the closing value of the S&P 500 Index and trained over a very long time period of six decades.

The methods used vary from using simple countable patterns to the complex black-box neural network methodology. To make the comparison of results more understandable, one hundred thousand random simulations predicting the entire validation set were run and tabulated. The results from each predictive methodology are compared to the distribution values generated by the simulations. This allowed us to easily compare the results from the predictive methodologies. All methods were applied to an out-of-sample validation set of one and one-half years.

since the early 1970s, literature on the behavior of stock prices has followed two streams, one supporting market efficiency and the other supporting active portfolio management. Followers of market efficiency reason that information is incorporated quickly into the market and that prices fully reflect this information. Because of this, prices cannot be predicted because they fluctuate based on this continuous arrival of new information. Traders, alternatively, maintain the belief that forecasting is achievable. However, only a small number of them have managed to outperform the market over a long period of time. Thus, while market efficiency remains the prevailing theory, much effort is expended both by money managers and academics in an effort to forecast well over time. One common methodology used in active trading is technical analysis.
The instances of studies over the years that have examined the usefulness of active management strategies have been numerous (Fama and Blume, 1966; Jensen and Benington, 1970; Brown and Jennings, 1989; Brock, Lakonishok and LeBaron, 1992; Blume, Easley and O’Hara, 1994; Gencay 1998; Allen and Karjalainen, 1999; Lo, Mamaysky and Wang, 2000). These studies span the spectrum of findings. Some are critical of simple technical rules and find the random walk does as well; others find that, once transactions costs are included, the predictive advantage of trading rules is moderated; and finally, some find evidence that some technical indicators have significant ability to aid in predictions.

A more recent paper (Schulmeister, 2007) looked at technical trading strategies on the S&P 500 futures and their ability at predicting returns. This paper found that, in the 1960s and 1970s, the use of daily stock data was profitable. But the same indicators from 2000-2006 had lessening results for those strategies.

Other papers have focused more on fundamental aspects and macroeconomic data when developing forecasting models. Doran, Ronn, Goldberg (2005) found that short term expected returns were highly volatile. Avramov and Chordia (2002) used firm specific factors to predict returns. Prominent factors for predicting S&P returns are the Treasury yield and dividend yield. However, this predictability holds best for small-cap stocks, growth stocks, and momentum stocks, and not the broader market. Hajizadeh, et al (2012) used Garch and neural network models to successfully forecast the S&P volatility. Niaki and Hoseinzade (2013) looked at 27 potential financial and economic variables from March 1994 through June 2008 and were successful forecasting using this large set of internal and external variables. Fukushima (2011) followed a number of hybrid models on monthly data and recommended complex hybrid models as the best method for forecasting. Tsiah et al (1998) had earlier developed a hybrid neural network and rule-based system that predicted effectively over a six-year period. This paper develops a number of specialized signals similar to those of technical analysis. Kara et al (2011) used ten technical indicators as inputs in both an artificial neural network model and a support vector machines model and found that the ANN outperformed the SVM.

The recurrence of patterns in the S&P 500 over time and their possible usefulness in forecasting has been detailed in Malliaris and Malliaris (2013). This current paper uses these recurring patterns with various forecasting methodologies to see which yields the best set of forecasts over a one and one-half year period. Forecasts were evaluated two ways, first by counting the number of correct and incorrect predictions, and second by calculating the amount of gain or loss achieved by following each of the strategies.

**DATA SETS AND CALCULATIONS**

Closing values of the S&P 500 Index were downloaded from Yahoo Finance for January 1950 through July 2011. For each day in this data set, the change in value from one day to the next was calculated, and a column indicating the direction of the change (up or down) was added. In addition, fields with patterns of up and down movements of consecutive days were concatenated from individual day movements.
These columns of concatenated directional movement were formed for two days through seven days. An example of a five-day string of these movements might be UDDUU. A last set of columns was created by counting the number of up movements only within each of these strings of concatenated movement. A column counting up movements was created for each of the two through seven day string columns. Using the five-day string example just mentioned, this column would have a value of 3 for that day.

The data set was then split into disjoint sequential training and validation sets. The training set included all days from 1950 through 2009 (over 15,000 days), and the validation set contained values from 2010 through mid-2011 (387 days).

One of the columns contained in the data set captured the amount of change for each day in comparison to the previous day. That is, today’s close minus yesterday’s close. The absolute value of this change is used as a measurement of the movement for the day. This value was added to a total to represent gain, or subtracted from the total to represent loss. So, for each day in the validation set, we have the amount the Index changed, and the direction the Index actually moved in comparison to the day before.

We made forecasts for each day of the validation set using numerous methodologies. Each prediction for a day was either right or wrong. The predicted direction was determined before the market opened that day and the position was closed at the end of the day. For each day, if the prediction was right, then the absolute value of the change was the amount made. If the prediction was wrong, the absolute value of the change was the amount lost. Examples of the calculated gain or loss are shown in Table 1.

<table>
<thead>
<tr>
<th>Change in Index</th>
<th>Actual Direction</th>
<th>Predicted Direction</th>
<th>Amount of Gain/Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>Down</td>
<td>Down</td>
<td>5</td>
</tr>
<tr>
<td>-5</td>
<td>Down</td>
<td>Up</td>
<td>-5</td>
</tr>
<tr>
<td>5</td>
<td>Up</td>
<td>Down</td>
<td>-5</td>
</tr>
<tr>
<td>5</td>
<td>Up</td>
<td>Up</td>
<td>5</td>
</tr>
</tbody>
</table>

At the end of the validation set (387 days), the daily values gained or lost were totaled. This became the amount made over a year and a half using a particular method of forecasting. The total amount gained (or lost) allowed for comparison among methodologies.

**RANDOM WALK SIMULATIONS AND FORECASTS**

For a single simulation, random numbers were generated from the uniform distribution for each day of the validation set. If a random number was greater than 0.5, then Up was predicted as the direction the S&P 500 Index would move tomorrow, otherwise the prediction was set to Down. Predictions were compared to actual direction, and the daily amount gained or lost was recorded. At the end of each run of the simulation, a total was calculated by summing the daily gains/losses for each of the 387 days.
This was repeated 100,000 times, with each single run yielding a set of random number forecasts for the entire validation set and a total value of daily gains/losses. These 100,000 totals were graphed in a histogram in Excel and ranged from a low of -1028 to a high of 1048. The mean and standard deviation for these totals were -0.0995 and 235.48 respectively. The histogram is shown in the Figure 1.

Figure 1. Total gains/losses for 100,000 simulated random forecasts for 387 days

PATTERN FORECASTS

The training set was used to determine the most dominant patterns with all strings of a given length. These dominant patterns were then used to determine the direction of the forecasts for each day in the validation set.

Using only the training data set, we counted the actual number of times strings of Up and Down patterns occurred for one through seven days. A sample of the patterns and the number of times each sample occurred in the set of training rows from January 1950 through December 2009 is shown in Table 2. We found that it was not possible to construct useful pattern combinations past those of seven days in length because every possible pattern did not occur in the training set. When such a pattern does not occur in the training set, we have no way of forecasting if/when it does happen in the validation set. Thus, we used only string patterns up to seven days in length.

We compared each pair of patterns that differed in only the last value of Up or Down in the training set. The pattern that occurred more often in the pair was specified as dominant and was used to determine a forecast on the validation set.
For example, if we look at the training patterns using only one day, would always predict Up tomorrow in the validation set because Up (8078 occurrences) dominated Down (7009 occurrences). In two day forecasts, there are two possible strings beginning with U [UD and UU]. Of these two, UD occurred in the training set fewer times than UU. Thus, when we have a movement of U for today in the validation set, we will forecast that tomorrow would also be U. With the days where D occurred first in the string of two days, the training set shows that DD occurred more often than DU. In this case, when D is the first day of a two-day string in the validation set, we will forecast that the second day in the string will also be D. Using the seven day training patterns DDDDUDD and DDDDUDD illustrated above, we see that DDDDUDD is dominant. Thus, when we see DDDDUDD in the validation set, this set of patterns would indicate that we should predict D for the next day. Table 3 illustrates this using these patterns from the training and validation sets respectively.

Thus, for each pair of strings from two to seven characters in length that differ in the last letter of the pattern, the forecast for tomorrow in the validation set was determined by the next movement that occurred most often in the training set. We used this methodology to generate seven forecasts for each row in the validation set.

Then, for each of these forecasts on the validation set, we compared the value predicted with the actual value that occurred on day t+1 and used the information to construct a gain/loss column. If the actual and predicted values matched, the absolute value of the change of the S&P from day t to day t+1 was recorded. If the actual and predicted values did not match, then the negative of the absolute value of change was recorded. At the end of the validation set, all gains and losses
were totaled. This total was calculated for each of the string forecasts, from 1 day to seven days, for seven forecasts overall.

**NEURAL NETWORK FORECASTS**

A neural network was built with IBM’s SPSS Modeler data mining package. Using this software, a network was constructed with two hidden layers of seven and five units respectively. The inputs used for training the network included the closing value and percent change in the closing value of the Index, the up and down pattern strings for one day to seven days, and the number of Up days in the 2 through 5 day strings.

Figure 2. Architecture of the neural network with two hidden layers.

This network used a hyperbolic tangent function in the hidden layers, and a random seed of 229176228. During training, 30% of the data set was held out for testing to prevent over-fitting. The Modeler package also generates a relative importance output for the variable used. This relative importance measure gives the impact of each variable on changes in the target. In this comparative list, the most important variables were the percent change in the close, the patterns in the day 4 through day 7 strings, and the number of ups in the past three days. Figure 2 shows the network architecture.

After the neural network was trained, the validation set was run through and the forecasts were generated for 2010 and 2011. These forecasts were compared to their actual counterpart, and a total of gains and losses were tabulated.
DECISION TREE FORECASTS

The decision tree for this problem was trained using the data up through December of 2009 and a C5.0 model in IBM’s SPSS Modeler program. We gave the decision tree the following inputs to use in building the tree: the up-down patterns from one to seven days, the number of up days in 1 to five days, and the closing value today. The target column was tomorrow’s direction.

The resulting decision tree was wide and deep, and correct about 67% of the time on the training set and 55% of the time overall on the validation set. In addition to supplying a forecast for future values, this Modeler technique also looks at each of the input variables and ranks them in terms of relative importance to the model, with more important variables occurring higher up on the tree.

The four variables ranked highest in importance to the forecast were the direction today, the closing value today, the 7-day pattern [for example UDUUDDU], and the number of Up movements in the last three days. The decision tree used one-day, seven-day and three-day information. That is, this approach took both close and distant information into consideration when constructing the forecast.

As with the random walk, string-pattern, and neural network forecasts, actual and predicted values from the decision tree were compared for the validation set. The gain or loss from each day was recorded, and totaled at the end of the set of 387 days to give a value we can use to compare the models.

COMPARISON OF METHODOLOGIES AND SUMMARY

These various methods of forecasting can be compared in several ways. First, we look simply at the number of times each method correctly forecasted the daily direction out of 387 days. Then, we drill down to inspect whether these correct forecasts were in the Up or Down direction. In the validation data set, of the 387 days, 168 were actual Down movements and 219 were Up. The method having the largest number of correct forecasts was the neural network with 222 correct days. However, looking at the Down direction, we see that the 2-day strings did the best job there, with 72 out of 168 correct matches. Focusing only on Up forecasts, the best method was the 6-day string patterns, with 178 out of 219 correct days. Table 4 shows these counts for each methodology. The last column has the total number of correct directional movements, while columns two and three divide this total into correct forecasts for each of the directions individually.

We can also compare these methodologies using the total amount gained or lost over the 387 day period by following the strategy. As previously explained, if the predicted direction matched the actual direction, then the absolute value of the day’s change was added to a total. If the predicted and actual directions were different, then the absolute value of the day’s change was subtracted from the total.
Table 4. Number of correct directional forecasts for each methodology.

<table>
<thead>
<tr>
<th>Method</th>
<th>Down</th>
<th>Up</th>
<th>Number Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Day String</td>
<td>0</td>
<td>219</td>
<td>219</td>
</tr>
<tr>
<td>2 Day String</td>
<td>72</td>
<td>123</td>
<td>195</td>
</tr>
<tr>
<td>3 Day String</td>
<td>41</td>
<td>164</td>
<td>205</td>
</tr>
<tr>
<td>4 Day String</td>
<td>41</td>
<td>164</td>
<td>205</td>
</tr>
<tr>
<td>5 Day String</td>
<td>45</td>
<td>167</td>
<td>212</td>
</tr>
<tr>
<td>6 Day String</td>
<td>40</td>
<td>178</td>
<td>218</td>
</tr>
<tr>
<td>7 Day String</td>
<td>48</td>
<td>164</td>
<td>212</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>43</td>
<td>168</td>
<td>211</td>
</tr>
<tr>
<td>Neural Net</td>
<td>61</td>
<td>161</td>
<td>222</td>
</tr>
</tbody>
</table>

Table 5. Total amount gained or lost [in S&P points] by following the strategy, and comparison to the random simulations

<table>
<thead>
<tr>
<th>Type of Forecast</th>
<th>Total Points Gained/Lost</th>
<th>Z-Score Compared to Random Histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1DayString</td>
<td>193.96</td>
<td>0.825</td>
</tr>
<tr>
<td>2DayString</td>
<td>-21.08</td>
<td>-0.087</td>
</tr>
<tr>
<td>3DayString</td>
<td>174.34</td>
<td>0.742</td>
</tr>
<tr>
<td>4DayString</td>
<td>174.34</td>
<td>0.742</td>
</tr>
<tr>
<td>5DayString</td>
<td>307.22</td>
<td>1.305</td>
</tr>
<tr>
<td>6DayString</td>
<td>335.84</td>
<td>1.426</td>
</tr>
<tr>
<td>7DayString</td>
<td>306.92</td>
<td>1.303</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>168.16</td>
<td>0.715</td>
</tr>
<tr>
<td>Neural Net</td>
<td>393.40</td>
<td>1.670</td>
</tr>
</tbody>
</table>

Total amounts gained or lost over the 387 day period by following a single strategy are shown in Table 5. Here we see that the only loss was from following the 2-day string pattern strategy. This yielded a loss of $21.08. The highest gain came from the neural network which ended up with a gain of $393.40. The second highest gain can from the 6-day string strategy which had a total of $335.84. Recall that the 100,000 randomly generated simulations had a mean of -0.0995, that is, a loss of about 10 cents with a standard deviation of $235.48. These are approximately normally distributed with the lowest value indicating a loss of $1,028 dollars, and the highest a gain of $1,048 dollars.

The last column of table 5 displays each of the different methodology totals as a z-score using the mean and standard deviation of this distribution. This shows that the neural network gain has a z-score of 1.67, higher than 95% of the randomly generated simulations, while the 6-day string strategy has a z-score of 1.43, higher than 92% of the random simulations. While the neural network did a better job overall, the 6-day string methodology did predict more Up days.
correctly and is a more straight-forward method that is easily understood. Neural networks are considered a “black-box” methodology and every time they are re-trained, the results may differ.

REFERENCES


