Feature Investigation for Stock market Prediction

Hui Lin
Department of Aeronautics and Astronautics
Stanford University
linhui@stanford.edu

ABSTRACT

In this project, machine learning algorithms were used to forecast the price of the future stock market. Notably, MATLAB's Neural Networks (NNets) and Support Vector Machines (SVM) were used for the classification problem of whether the stock price has increased or decreased compared to the price at the last timestamp. The amount of data shifting was also investigated. It was found that the more recent the data that was used for prediction, the better the prediction accuracy was.

In implementing these machine learning algorithms, 42 features such as the quarterly PE ratio, COGS and Net Income were used. Using NNets, an accuracy of up to 87% was achieved for Microsoft depending on the selected features. With SVM, an accuracy of up to 66% was achieved.

Feature selection techniques such as forward search and backward search were also investigated in order to improve the accuracy of the stock price predictions.

SUMMARY

1. Preliminary investigation

In the beginning of this project, a preliminary investigation was done to determine the effectiveness of the NNets in predicting the future price of Microsoft. The previous-day closing price of Microsoft, previous-day closing price S&P 500, and moving averages were used as the features to predict the next-day closing price of Microsoft. Hidden layers (HL) ranging from 2 to 100 for the NNets were tested.

Regardless of the number of HLs or the selected features, the highest accuracy that was achieved was 55%. Upon further investigation and reading more papers, it was concluded that using only the historical prices as the features would not be enough to generate accurate predictions.

2. Data collection

More feature data were collected upon concluding that the historical prices were not enough to generate a good prediction. Notably, 42 key financial ratios ranging from S&P 500 price, PE ratio, Cost of Good Sold (COGS) to Net Income were obtained. Following that, quarterly data ranging from 1992 to 2013 were extracted since these ratios were only available on a quarterly basis. Another reason for using a quarterly data was that using a slightly longer-term data may have been able to filter out some of the random noises from the daily fluctuations of the stock prices.

3. Data shifting

In predicting the future output values, the shifted historical data was used. For example, by shifting the data set by 1, y_n could be predicted using x_{n-1}. This can be extended to

any amount of data shifting where y_n can be predicted using x_{n-k}. However, after running a few simulations, it was quickly found out that the amount of data shifting should be minimized for a good prediction accuracy. This makes sense intuitively as the future price of the market is closely related to the recent historical data rather than the older historical data.

4. Training and validation

After gathering the data, it was split into 80% for training and 20% for testing. Within the 80% for training, it was furthermore split into 90% for training and 10% for validation. This was required as the criterion for convergence for the training of the neural network was when the validation error would be larger than the training error.

Upon convergence, the accuracy of the training model was determined. Two forms of cross validations were used. The 'static' algorithm used $\{x_1, x_2, \dots x_k\}$ to train the model, and the model was used to predict $\{y_{k+1}, y_{k+2}, \dots y_n\}$. The 'dynamic' algorithm is an online-algorithm that goes as follows. During the first iteration, $\{x_1, x_2, \dots x_k\}$ was used to train the model, and the model was used to predict y_{k+1}. In the next iteration, $\{x_1, x_2, \dots x_{k+1}\}$ was used to training the model, and y_{k+2} was predicted. This was repeated until there were no more points in the testing set to predict. The dynamic training algorithm had higher prediction accuracy since the training data set contained the information of the latest financial ratios.

The metric that was used for determining the accuracy goes as follows. Given a test output set $\{y_1, y_2, \dots y_n\}$, the corresponding labels $\{a_1, a_2, \dots a_{n-1}\}$ were generated. Where

$$a_i = \begin{cases} 1, if\ x_{i+1} > x_i \\ 0, otherwise \end{cases}$$

The accuracy of the trained model is then, given by the total number of correct label realization divided by $n - 1$.

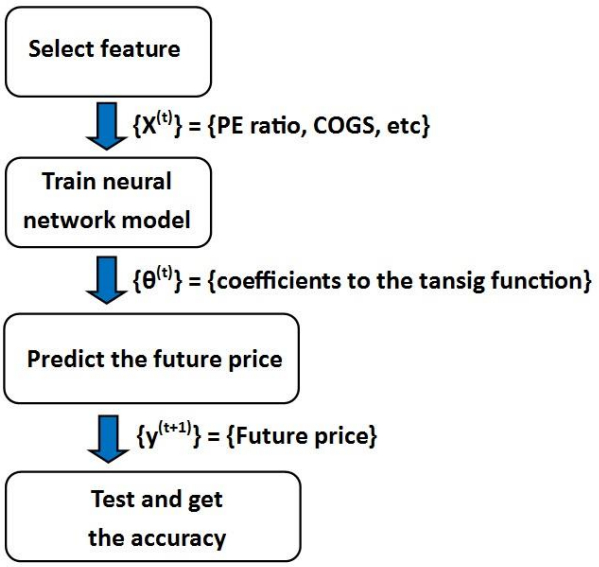

Figure 1: Flow diagram for the neural network training model

5. Summary of the results

The following tables summarize the table from running the NNets. Table 1 summarizes the prediction accuracy determined by averaging all of the accuracies for a single-feature neural network for a certain number of HL. As shown in table 1, using a HL of 5 gave the highest average prediction accuracy.

Table 1: Average prediction accuracy with respect to the number of hidden layers for Microsoft

Neural Network with single feature (Microsoft data)	
Number of Hidden Layers	Average prediction accuracy of all the features
2	67.86%
5	68.58%
14	63.11%
20	61.21%
30	58.86%

Table 2 shows the prediction accuracy of the 'static' algorithm for the top 5 features using HL of 5.

Table 2: Top 5 features for Microsoft for the 'static' algorithm using neural network

Feature	Accuracy
Shareholders' Equity	83.50%
Common Stock Net	81.06%
Total Current Assets	80.68%
Net income	80.43%
Selling, General & Administrative Expense	78.54%

Table 3 shows the prediction accuracy for the 'dynamic' online algorithm for the top 3 features using HL of 5. As expected, the dynamic algorithm gave a higher prediction accuracy than the static algorithm.

Table 3: Top 5 features for Microsoft for the 'dynamic' algorithm using neural network

Feature	Accuracy
Shareholders' Equity	87.49%
Common Stock Net	86.48%
Total Current Assets	83.87%
Accounts Payable	83.18%
Gross Profit	82.31%

Table 4 shows the prediction accuracy for these features when used in linear regression. As shown in table 4, the prediction accuracy for linear regression averaged at about 50%.

Table 4: Prediction accuracy using linear regression for the top features from neural network

Feature	Accuracy
Shareholders' Equity	55.45%
Common Stock Net	45.98%
Total Current Assets	52.21%
Accounts Payable	50.00%
Gross Profit	53.32%

The same strategy was also done with IBM. For IBM, it was better to include all 42 features rather than just a single feature. With a single feature, the accuracy ranged from 45% - 55%, which was not useful. By including all 42 features and tweaking the number of HL, an accuracy of up to 75% was achieved.

Finally, a preliminary investigation using MATLAB's SVM package with the Gaussian kernel was also done. By using all 42 features, the SVM was able to achieve an accuracy of 66% for Microsoft, which was significantly lower than the accuracy achieved by NNets.

6. Feature selection

Since there were 42 features, feature selection techniques were implemented in order to improve the accuracy.

The first method that was tried was to combine the features with the highest accuracies from the single feature prediction. As shown in table 5, the more the feature was combined, the lower the accuracy was. This was rather unexpected as an increase in the prediction accuracy was anticipated from combining the top features.

Table 5: Prediction accuracy obtained from combining the top features

Feature	Accuracy
Top 1 feature	87.49%
Top 2 features	79.92%
Top 3 features	77.86%
Top 4 features	75.93%
Top 5 features	73.61%

Upon seeing that simply combining the top features didn't increase the accuracy, forward search and backward search feature selections were implemented. The forward search goes as follows. Starting from the first feature, one feature was added at a time. Then, the prediction accuracy was determined for the

combination of those features. If the accuracy increased by adding the new feature, the added feature was kept. If the accuracy dropped, that feature was not added to the feature list. Most of the time, the forward search didn't work well since it ended up with just 1 feature.

On the contrary, backward search started with all 42 features. Then, features were removed or kept one at a time. Figure 2 shows the evolution of the prediction accuracy as new features were kept or removed from the backward search feature selection. As shown in the figure, the backward feature selection did not improve the prediction accuracy.

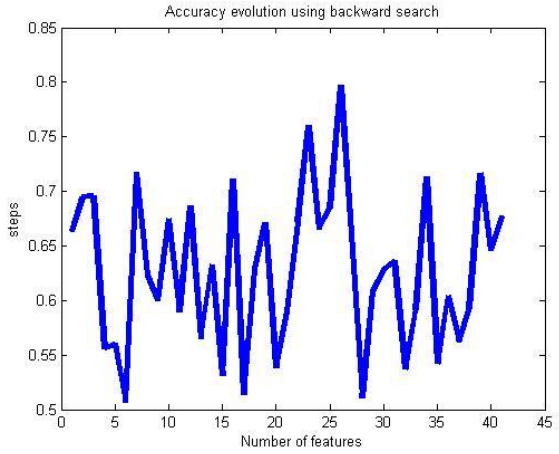

Figure 2: Prediction accuracy evolution from backward search feature selection

CONCLUSION

With neural networks, a prediction accuracy that was noticeably higher than 50% was achieved. It was also noted that the feature selection played an important role in terms of the prediction accuracy. Interestingly, for Microsoft, the highest prediction accuracy was achieved when using only 1 feature as an input. Such was not the case for some of the other companies such as IBM where prediction accuracy using 1 feature averaged at around 50%. With IBM, a much greater prediction accuracy of 75% was achieved when all 42 features were used.

FUTURE WORKS

Some of the future works include a most robust method of coming up with selecting the features that would give the highest prediction accuracy. Neither forward search nor the backward search feature selection worked for this specific stock price prediction problem.

Another point to investigate is to have a moving training data set. Currently, the old training data are always kept even when new data are being added after every iteration. However, in stock market prediction, the data from 20 years ago may not be helpful in predicting the future price of the stock. Thus, the removal of the old data from the training data set when adding in new data should be tested in order to improve the prediction accuracy.

ACKNOWLEDGEMENT

I would like to acknowledge Jordan Klovstad for helping me with acquiring all the important financial data that was used for this study.