Predicting Stock Price Movements Using Supervised Learning Algorithms

Zhao Yang yz931129@gmail.com
University of California, Los Angeles, Department of Computer Science, CA 90025 USA

Abstract

In this project, I aim to apply supervised machine learning algorithms to predict the price movements of Alphabet Inc.’s Class A shares. Using bootstrap aggregation, I achieve a superior >90% accuracy on long-term price movement prediction as compared to the baseline accuracy of ~60% when trivially predicting all examples using the most popular label.

1. Background

Institutional and private investors make investment decisions by analyzing relevant market data and financial data. Most times, the essential part of the decision making process comes down to personal interpretations of the meaning of the data, which is based on past experiences and individual preferences. Moreover, there are two stock value evaluation methodologies, fundamental analysis and technical analysis. Fundamental analysis aims to extract and analyze fundamental data that reflects the health and potential of a business, while technical analysis makes predictions on future stock prices using past prices and other trading variables.

Both methods have their limitations. Data required by fundamental analysis is often incomplete, biased, and difficult to obtain. Technical analysis is heavily influenced by personal preferences in the interpretation of data, with different personal judgement yielding different predictions.

Therefore, in this paper I make the first step to develop an automatic stock price prediction mechanism. In addition, the findings of this project offer insights into the debate regarding the truth and falsity of the efficient market theory, which generally states that past prices do not contain useful information for the prediction of future stock prices.

The core objective of this research project is to classify one company’s future price movements over some time window as “Up” or “Down” given a time-stamped data entry which consists of past and current pricing information. This paper investigates the application of several popular supervised machine learning algorithms to the prediction of price movements of Alphabet Inc.’s Class A shares over varying prediction time windows.

2. Data

All of the financial and stock data of Alphabet Inc. was collected from Bloomberg Terminal at Anderson School of Management at UCLA under the license for academic use.

2.1 Format

All fields in a data entry are of numerical value. Data entries are sorted in reverse chronological order with the most recent appears on the top and the oldest appears at the bottom. Each data entry represents one trading day and may only contain past and current information. Future information is not contained in the data entry.

2.2 Content

There are two types of collected data, technical indicators and financial data. The period is from Nov. 2008 to Nov. 2015. Technical indicators are either past and current stock prices or derivatives from past and current prices. Financial data incorporates fundamental data which measures the health of and risks associated with the company.

Technical Indicators: Commodity Channel Index (CMCI), Bollinger Bands (%B), Directional Movement Index (DMI), General Overview Chart (GOC), Moving Average Oscillator (Osc), Relative Strength Index (RSI), Simple Moving Average on Close for 5, 20, 50, 100 and 200 Days, Stochastics, Trading Envelopes (TE), Williams %R Value, Open, High, Low, Close, Volume, and Simple Moving Average on Volume for 15 Days.

Financial Data: PE Ratio, Enterprise Value, Volatility 10 Day, Overridable Alpha, Alpha for Beta Plus Minus, Overridable Raw Beta, and Basic Earnings per Share.

2.3 Label Generation

Number 1 represents an upward price movement; number -1 represents a downward price movement. 1 and -1 are signs of the difference between a future price and the current price. For example, in Table 1, the label for the data entry from 11/19/2015 is 1 because on 11/20/2015, the price goes up to $777.00 from $759.94 on 11/19/2015. Similarly, the label for the data entry from 11/18/2015 is -1 because on 11/19/2015, the closing price dropped to $759.94 from $760.01 on 11/19/2015. The time window in Table 1 is one day. For longer periods, we use the price a corresponding number of days into the future and the price of the current day. By defining the label this way, it is ensured that the algorithm is predicting a future price using only past and current information.

Date	Closing Price	Label
11/20/2015	777.00	
11/19/2015	759.94	1
11/18/2015	760.01	-1

Table 1. Label Generation

3. Feature Selection

Forward feature selection is used to greedily select a subset of possible features, and 5-fold validation is used to measure the effectiveness of a feature. Rather than use all available features at hands without any basic understanding of their individual importance, I used the *sequentialfs* function from MATLAB to progressively choose a small set of features to ensure that the selected features achieve top-tier accuracies while remain efficient. Fewer features can also help reduce the risk of overfitting. In my data set, since there are over 1700 data points and only 27 feature candidates, the curse of dimensionality is not a problem. Nonetheless, feature selection improves the efficiency of the classification model.

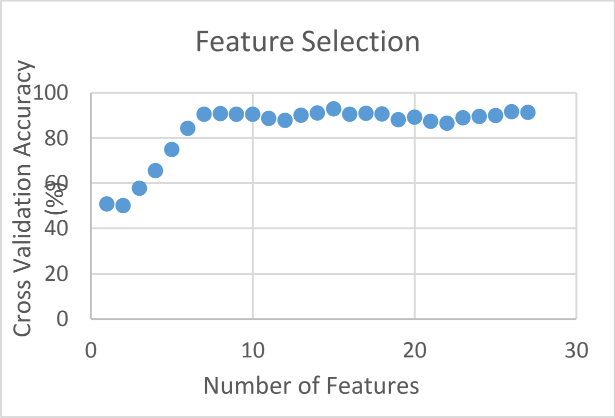

Figure 1. *The relationship between cross validation accuracy and the number of features*

Figure 1 shows the cross validation accuracy of the bootstrap aggregation algorithm over a 30-day time window with respect to the number of features. After the first 7 features, the validation accuracy reaches the peak and does not increase substantially afterwards, which indicates that the first 7 features account for most of the variance in the data and are sufficiently good indicators for the labels. The following table shows the selected 7 features.

%B	SMAVG(5)	SMAVG(50)	PTPS	RSI	%DS	Close

Table 2. Features selected by forward selection

They are all pricing information indicators. Some are stock prices such as Close and SMAVG, and others are derivatives based on prices such as RSI, which calculates the velocity of price movement and %B, which measures price volatility using standard deviations and moving averages.

4. Algorithms

In this project, I applied 4 algorithms, which are bootstrap aggregation, support vector machines, Adaboost, and quadratic discriminant analysis. Quadratic discriminant analysis achieves very poor prediction accuracy, therefore its details are omitted. For each algorithm, I applied 5-fold cross validation and used classification accuracy as performance measurement. For bootstrap aggregation and Adaboost, I analyzed their performance with respect to the number of weak learners to ensure they have converged to local optima.

4.1 Bootstrap Aggregation (Bagging)

I used the *ensemble* function in MATLAB to train a decision-tree based bootstrap aggregation classifier.

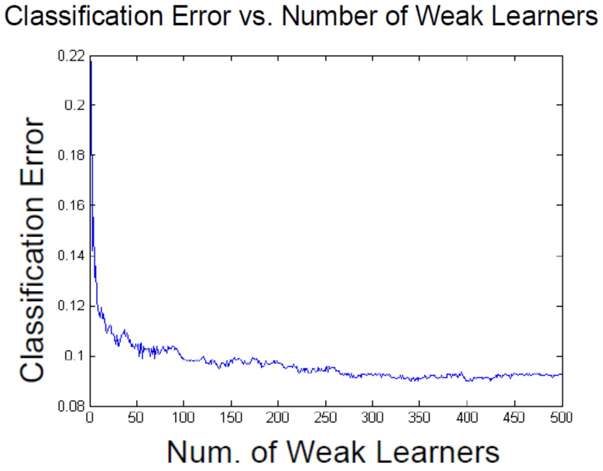

Figure 2. Error analysis for bootstrap aggregation

As shown in Figure 2, the 5-fold cross validation error decreases quickly as the number of trees grows. The error is less than 10% after the number of weak learners reaches 100, and continues to decrease until it remains about 5% for 300 or more decision trees.

4.2 Adaboost

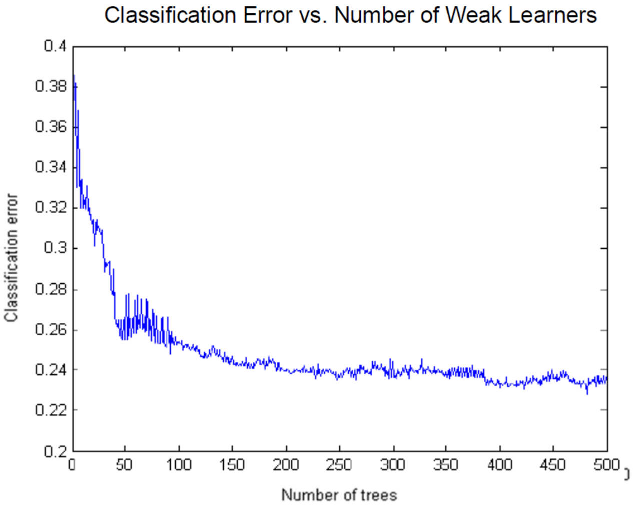

Figure 3. Error analysis for Adaboost

I used the *ensemble* function in MATLAB to train a decision-tree based Adaboost classifier. As shown in Figure 3, the 5-fold cross validation error decreases quickly as the number of trees grows. The smallest error converges to about 23% as the number of weak learners increases to 500.

4.3 Support Vector Machines

Support vector machines aim to find the hyperplane that best separates the data points in high-dimensional spaces. It achieves so by maximizing the functional margin (distance from the closest data point to the decision boundary) of the decision boundary while tolerating misclassified data points within this margin. It also employs kernel functions to map features into higher dimensional spaces to search for non-linear decision boundaries.

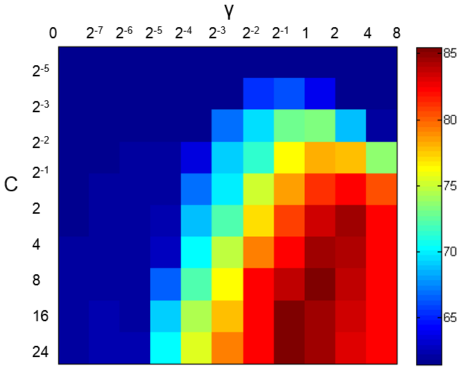

Figure 4. SVM optimization

The parameter which controls the toleration of misclassified data points is C, and the parameter that controls the complexity of the kernel function is γ for radial basis function (RBF). Figure 4 shows the cross validation accuracy at each configuration of these two parameters. From left to right, γ increases, and from top to the bottom, C increases. The value of accuracy increases from blue to red, with dark blue being the lowest and dark red being the highest. We can observe an interesting pattern in the color map, which is the diagonal distribution of accuracies. A certain range of C and γ tend to achieve a stable accuracy value. In addition, the accuracy increases as both parameters increase.

The diagonal distribution of accuracies can be explained as the following. γ represents inverse-width parameter of the RBF kernel. A larger value means smaller influence area of one data point. When γ is large, the decision boundary tends to wrap around individual points and the curvature increases. At this point, a large C will “restore” the curvature by penalizing margin error of the support vectors. Therefore, regionally, C and γ can have counter effects, hence the diagonal distribution of accuracy values. In practice, to save memory, we usually want to limit the value of C and adjust the value of γ.

Both the increase of C and γ will result in the increase of variance of the prediction model. From the blue area to the red area, SVM’s ability to generalize increases, and the area to the left and top of the dark red area are underfit area because both C and γ’s values are smaller in that region.

5. Results

The cross validation accuracies are computed over time period of 1, 2, 3, 4, 5, 10, 15, 20… 90 trade days. The first 5 time windows are defined as short periods, and the rest of the time windows are define as long periods. Short-term predictions and long-term predictions have different results, which is analyzed in section 5.1 and 5.2.

5.1 Short-Term Predictions

Short-term predictions are predictions made over period equal or shorter than 5 trade days.

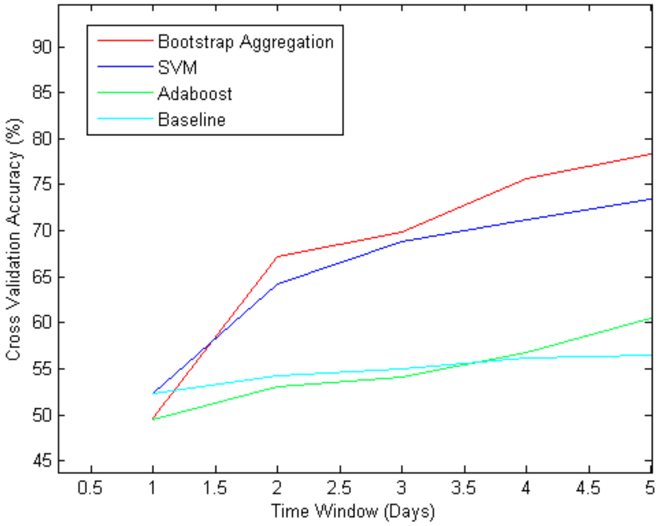

Figure 5. Prediction accuracies over prediction periods for over 5 trade days or fewer

Figure 5 shows that boostrap aggregation achieves the best results over short-term predictions. At time_window = 5, it achieves nearly 80% accuracy as compared to 55% baseline accuracy. SVM is the second best with significant accuracy increase as compared with the baseline accuracy, which achieves 73% accuracy.

5.2 Long-Term Predictions

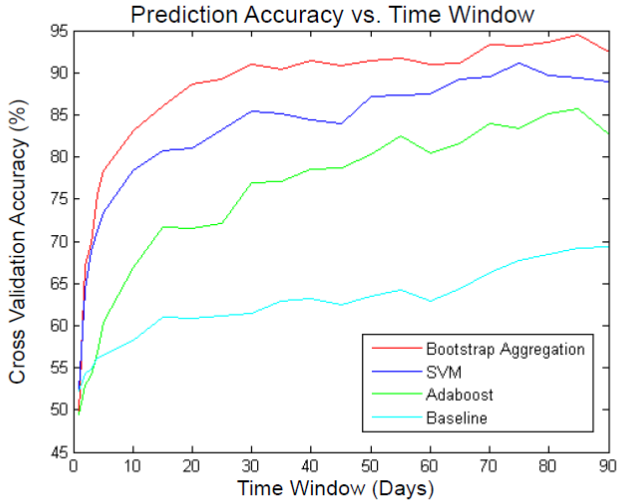

Figure 6. 1-day to 90-day prediction accuracies

As shown in Figure 6, bootstrap aggregation also achieves the best prediction accuracy in long-term predictions. Its highest accuracy is nearly 95% with a baseline accuracy of 68% over 85-day period. The second best is SVM with its top accuracy being 91% over 67% baseline accuracy. Adaboost is the third over almost all time windows. In the next section, the strength of the model is evaluated using relative gain in accuracy compared to the baseline, rather than absolute values.

These results strongly support that there is relevant information in the past prices of a stock and that machine learning algorithms can effectively capture this information to predict price movements.

5.3 Comparison and Evaluation

In this section, we conclude that all three models achieve the best classification accuracy over the 30-day time window and demonstrate the performance measure based on the ratio gain in accuracy.

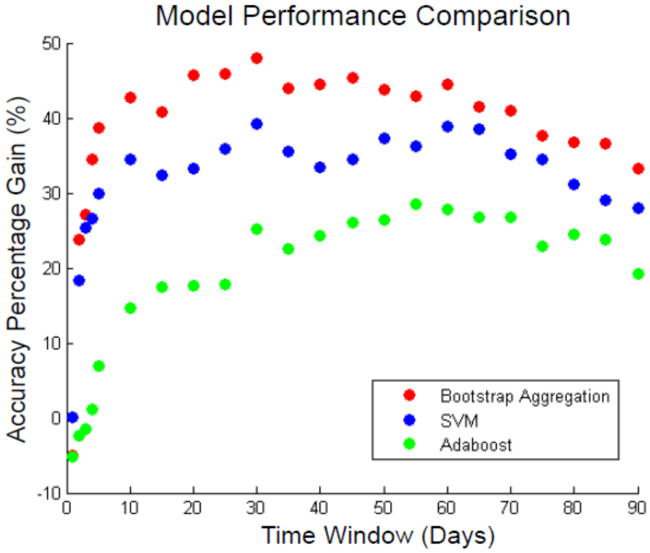

Figure 7. Percentage gain in accuracy for each algorithm from 1-day to 90-day periods

$$Gain = \frac{|Prediction - Baseline|}{Baseline} \times 100\%$$

Gain (percentage gain) measures how much better the prediction model is doing than the trivial classifier. It shows in Figure 7 that bootstrap aggregation achieves 48% increase in classification accuracy, with SVM ranking second with 40% and Adaboost third with 25%.

5.4 Conclusion

The results have shown that past prices can be used to predict future stock price movements with extremely high accuracy in the long term, and reasonable accuracy in the short term for time windows greater than 1 day. The next-day predictions are equivalent to random guesses.

These results pose a challenge to the efficient market theory which states that all available market information is contained in the current stock price, and no investors can beat the market. Although this project did not predict the actual value of stock prices, it demonstrates that current and past prices can be used to predict future price movements. However, the validity of this finding may be undermined if a significant portion of "Up" or "Down" predictions do not translate into significant value changes. For example, a change from $700.01 to $700.02 does not generate any real profit, so in this case a correct "Up" prediction loses significance.

Nonetheless, the project successfully extracted features from past prices and tuned multiple supervised learning models to predict future stock price movements with >90% accuracy on Alphabet Inc.'s Class A shares.

6. Future Direction

From the low next-day prediction accuracy, it can be inferred that past prices cannot capture the daily dynamics of stock prices. Through further research, I have identified the limit order book as the next exciting tool to study intra-day stock price movement. The information hidden within an LOB is dynamic and intrinsically revealing, even though dark pools and frequent order cancellations complicate the prediction process.

References

Asa Ben-Hur and Jason Weston. A User's Guide to Support Vector Machines. *Methods in Molecular Biology*. **2010**, 609, 223-39. DOI: 10.1007/978-1-60327-241-4_13.

Sumit Das, Aritra Dey, Akash Pal and Nabamita Roy. Applications of Artificial Intelligence in Machine Learning: Review and Prospect. *International Journal of Computer Applications* **2015**, 115, 31-41.