SOPS: Stock Prediction using Web Sentiment

Vivek Sehgal and Charles Song
Department of Computer Science
University of Maryland
College Park, Maryland, USA
{viveks, csfalcon}@cs.umd.edu

Abstract

Recently, the web has rapidly emerged as a great source of financial information ranging from news articles to personal opinions. Data mining and analysis of such financial information can aid stock market predictions. Traditional approaches have usually relied on predictions based on past performance of the stocks. In this paper, we introduce a novel way to do stock market prediction based on sentiments of web users. Our method involves scanning for financial message boards and extracting sentiments expressed by individual authors. The system then learns the correlation between the sentiments and the stock values. The learned model can then be used to make future predictions about stock values. In our experiments, we show that our method is able to predict the sentiment with high precision and we also show that the stock performance and its recent web sentiments are also closely correlated.

1. Introduction

As web based technologies continue to be embraced by the financial sector, abundance of financial information is becoming available for the investors. One of the popular forms of financial information is the message board; these websites have emerged as a major source for exchanging ideas and information. Message boards provide a platform to investors from across the spectrum to come together and share their opinions on companies and stocks. However, extracting good information from message boards is still difficult. The problem is the good information is hidden within vast amount of data and it is nearly impossible for an investor to read all these websites and sort out the information. Therefore, providing computer software that can extract information can help investors greatly.

The data contained in the websites are almost always unstructured. Unstructured data makes an interesting yet challenging machine learning problem. On a message board, each data entry relates to a discussion to some stocks. This can be visualized as a temporal data where the frequency of words and topics is changing with time. The opinions on a stock changes both with time and its performance on the stock exchanges. To put it more formally, there is correlation between the sentiment and performance of a stock. In a study done recently [4], email spam and blogs have been found to closely predict stock market behavior. Message board data posses a challenge to data mining; opinions on message boards can be bullish, bearish, spam, rumor, vitriolic or simply unrelated to the stock. We found that the number of useless messages surpasses useful ones. Fortunately for us, the sheer size of total messages allows significant number of posts of relevant opinions to be analyzed, as long as we are able to filter out the noise.

In this paper, we introduce a novel way to do sentiment prediction using features extracted from the messages. As mentioned before, many of the sentiments extracted are irrelevant. To solve this problem, we develop a new measure known as "TrustValue" which assigns trust to each message based on its author. This method rewards those authors who write relevant information and whose sentiments closely follow stock performance. The sentiments along with their "TrustValue" are then used to learn their relation with the stock behavior. In our experiments, we find our hypothesis that stock values and sentiments are correlated is true for many message boards.

2. Related Work

In previous work on stock market predictions, [8] analyzed how daily variations in financial message counts effect volumes in stock market. [5] used computer algorithms to identify news stories that influence markets, and then traded successfully on this information. Both did real market stimulations to test the efficacy of their systems.

Another approach used in [4], tried to extract sentiments for a stock from the messages posted on web. They trained a meta-classifier which can classify messages as "good",

"bad" or "neutral" based on the contents. The system used both the naive bag of word models and a primitive language model. They found that time series and cross-sectional aggregation of message sentiment improved the quality of sentiment index. [4] reaffirmed the findings of [8] by showing that market activity is strongly correlated to small investor sentiments. [4] further showed that overnight message posting volume predicts changes in next day stock trading volume.

There has been a lot of work on sentiment extraction. In [3], a social network is used to study how the sentiment propagates across like minded blogs. The problem required natural language processing of text [6], they concluded that computation can become intractable for a large corpus like Web. Our goal is develop an efficient model to predict sentiment of message boards.

3. Approach

3.1 System Overview

Figure 1 gives an overall outline of our system. The first step involved data collection. In this step we crawled message boards and stored the data in a database. The next step was extraction of information from the unstructured data. We removed HTML tags and extracted useful features such as date, rating, message text etc. The information extracted is then used to build sentiment classifiers. By comparing on the sentiments predicted from the web data and the actual stock value, our system calculated each author's trust value. This trust value is then applied to filter noise, thus improving our classifier's performance. Finally, our system can make predictions on stock behavior using all the features extracted or calculated.

3.2 Data Collection

We collected over 260,000 messages for 52 popular stocks on $http://finance.yahoo.com$. The stocks were chosen to cover a good spectrum from technology to oil sector. The messages covered over 6 month time period; this large amount of data gave us a big time window to analyze the stock behavior. All the data extracted was stored in a repository for later processing.

On this website, the messages are organized by stocks symbol; a message board exists for each stock traded on major stock exchange such as NYSE and NASDAQ. Users must sign up for an account before they can post messages and every message posted is associated with the author. This features makes author accountability possible in our data processing step. Along with text messages, the authors can express the sentiments of their posts as "StrongBuy", "Buy", "Hold", "Sell" and "StrongSell". Aslo, other users

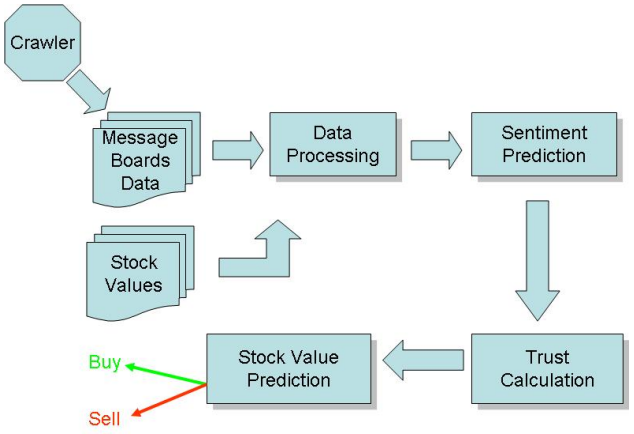

Figure 1. System Overview: The message board data is collected and processed. Then we predict the sentiments and calculate the TrustValues. These new features are then used to predict stock behavior.

can rate the messages they read according to their opinions and views, the rating is scaled out of five stars. And as with any message board, each message has a date, a title and message text.

We used a scraper program to extract message board posts for the chosen set of stock symbols. Figure 2 shows a sample message from Yahoo! Finance along with the relevant information circled. In this sample message, the author expressed extreme optimism about YAHOO! stock and encouraged others to buy the Yahoo! stocks as its current very underpriced with the possibility of a stock split. Our scraper program would extract the subject, text, stock symbol, rating, number of rating, author name, date and author's sentiment.

3.3 Feature Representation

After the relevant information has been extracted. We converted each message to a vector of words and author names. The dates are mapped to an integer values. The value of each entry in the vector is then calculated using TFIDF formula:

$$TFIDF(w) = TF(w) \times IDF(w)$$

$$TF(w) = \frac{n(w)}{\sum_{w'} n(w')}$$

$$IDF(w) = log(\frac{|M|}{\{m : w \in m\}})$$

M is the set of all messages while n(w) is the frequency of the term w in a message. The TFIDF (Term Frequency

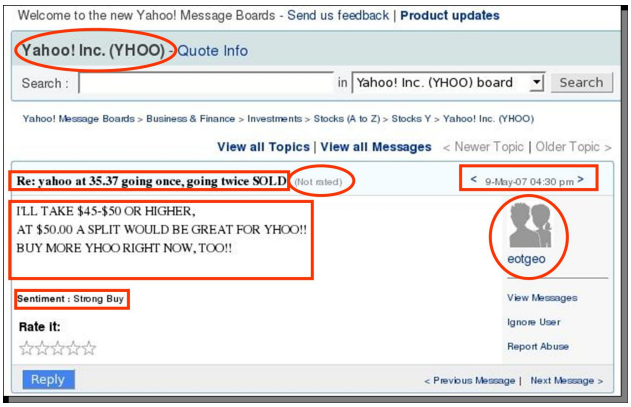

Figure 2. We extracted relevant information from the above message such as subject, text, date, author etc.

Inverse Document Frequency) weight is a statistical measure used to evaluate how important a term (i.e., word, feature, etc) is to a message in a corpus. The importance increases proportionally to the number of times the term appears in the message but it is offset by the frequency of the term in the corpus. Thus if the term is common in all the messages of the corpus then it is not a good indicator in differentiating messages.

3.4 Sentiment Prediction

We assumed that the sentiment of a stock is highly responsive to the performance of the stock and recent news about the company. For example, the news about introduction of iPhone by Apple can fuel considerable interest in the users, it can also affect their sentiments positively. Similarly, the sentiment can change when there is sharp change in stock performance in the near past. Using the above intuition, we modeled the sentiments as conditionally dependent upon the messages and stock value over the past one day (it can be extended to a longer time period in our system). The sentiment for a message m at time instant i is modeled as follows:

$$P(Sentiment|\theta) = P(Sentiment|m, M_i, SV_i)$$

This can also be visualized as a Markov process where the prediction at time instance i depends upon the values at previous time instance. In the above formula, M_{i-1} and SV_{i-1} correspond to the set of messages and stock value at time instant i-1 respectively. The parameters for the above model can now be learned using a suitable learning algorithm. For our experimentation, we used Naive Bayes, Decision trees [3] and Bagging [2] to learn the corresponding classifier. Naive Bayes is a simple model and can easily generalize to large data. Unfortunately, it is not able to model complex relationships. Decision trees on the other hand can encode complex relationships but can often lead to over-fitting. Over-fitting can cause the model to perform well for training data but is unable to show similar performance for actual data.

For classifier training, we used a popular toolkit known as weka [1] which provides all the standard classifiers such as Naive Bayes, Decision Trees, etc. In our data, a small fractions of messages had sentiments already assigned to them; their authors expressed the sentiment explicitly. These messages were used as ground-truth while training the sentiment classifier. We trained a classifier for each stock on a daily bases and each message is classified as "StrongBuy", "Buy", "Hold", "Sell" or "StrongSell". The number of features used by each classifier was in the range of 10,000.

3.5 TrustValue Calculation

We acknowledge that some authors are more knowledgeable than others about the stock market. For example, professional financial analysts should receive more trust, meaning their posts should carry more weight than the posts by the less knowledgeable authors. However, obtaining an author's background is tricky and difficult. Message boards commonly provide the user profile feature where the authors themselves can fill in information about their background. But this feature is often left unused or filled with inaccurate information.

Instead of discovering each author's background, we use an algorithm to calculate an author's TrustValue base on his or her historical performance on the message boards. For each message, we used the author's sentiment from the sentiment prediction step and compare them to the actual stock performance around the time of the post. If the author's post supported the actual performance, then author's trust value is increased. Not only do we care about the direction in which the stock price went, we also care about the magnitude. For example, if the author gave a strong sell sentiment in the post, but in reality the stock price only drop slightly, then the author should earn less TrustValue. We used percentage difference in stock price at closing bell as the normalized measure of stock performance.

Our algorithm also takes into account the fact that a single author cannot be expert on all stocks. It's commonplace for even professional financial analysts to keep track of only a set of stocks. This means an author can only be trusted for the set stocks he or she knows best. Each author can be assigned different trust values for different stocks, this feature enables us to paint a clear picture of each author's abilities with our algorithm. The TrustValue is calculated as follows:

$$TrustValue = \frac{PredictionScore}{NumberOfPreictions} + \frac{ExactPredictions + ClosePredictions}{NumberOfPredictions + ActivityConstant}$$

PredictionScore is equal to author's prediction performance that is how closely does the author's prediction follow the stock market. NumberOfPredictions is equals to the total number of predictions made by the author. ExactPredictions is the number of exact predictions made by the author. ClosePredictions is the number of "good enough" predictions made by the author. ActivityConstant is a constant used to penalize low activity or predictions by the author.

3.6 Stock Prediction

Stock prediction is a difficult task. In our method, we performed stock prediction on the basis of web sentiments about the stock. To formalize, we predicted whether the stock value at time instance i would go up or down on the basis of recent sentiments about the stock:

$$P(\Delta stock - value_i|\Theta) = P(\Delta stock - value_i|sentiment_i, trust_{i-1}, features_{i-1})$$

Figure 3 illustrates our stock prediction model. We learned a classifier which can predict whether the stock price would go up or down using the features extracted or calculated over the past one day. We use all of the features including sentiment and TrustValue to train classifiers such as Decision Tree, Naive Bayes and Bagging. In our experiments, we show strong evidence that stock value and sentiment are indeed correlated and one can predict changes in stock value using sentiments.

4 Experiments and Results

4.1 Evaluation

Sentiment prediction can be evaluated using statistical measures. Accuracy, which is defined as the percent of sentiments correctly predicted, is one method for evaluating approaches. The quality of results is also measured by comparing two standard performance measures, recall and precision. Recall is defined as the proportion of positive sentiments which are correctly identified:

$$recall = \frac{Positive\ instances\ predicted}{Total\ positive\ instances}$$

Precision is defined as ratio between the numbers of correct sentiments predicted to the total number of matches predicted:

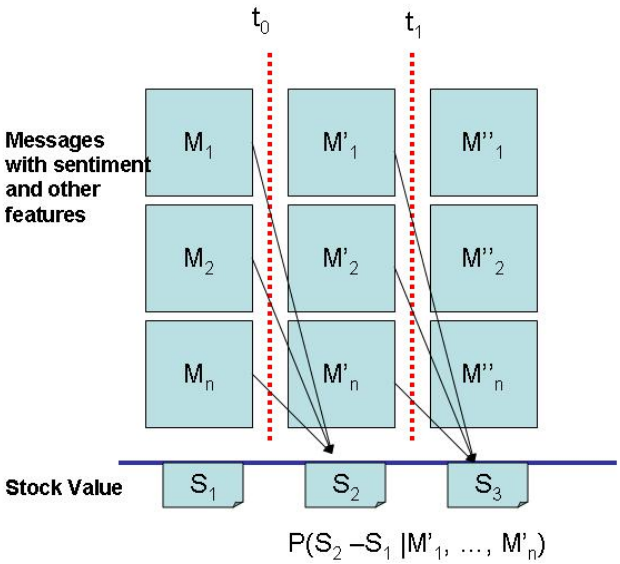

Figure 3. Our hypothesis: change in stock value is effected by sentiments of the past day. Figure illustrates our model as Bayesian Network.

$$precision = \frac{True\ positive\ instances\ predicted}{Total\ instances\ predicted}$$

One can increase recall by increasing the number of sentiments predicted or by relaxing the threshold criteria. But this would often decrease the precision of the result. In general, there is an inverse relationship between recall and precision. An ideal learning model has high recall and high precision. Sometimes recall and precision are combined together into a single number called F1, a harmonic mean of recall and precision:

$$F1 = \frac{2 \times recall \times precision}{recall + precision}$$

4.2 Experiments

4.2.1 Sentiment Prediction

We chose 3 stocks to show our sentiment predictions, namely Apple, ExxonMobile and Starbucks, covering different sectors of the economy. The aim of this experiment is to find out if sentiment prediction is possible using only features present in the message. Table 1 to 3 show our results. The sentiment classes are StrongBuy and StrongSell. We find that sentiment prediction can be done with both high accuracy and recall in our system. This implies that

Table 1. StrongBuy & StrongSell sentiment prediction for Apple.

StrongBuy			
Classifier	Recall	Precision	F1
Naive Bayes	0.24	0.42	0.31
Decision Tree	0.30	0.40	0.35
Bagging	0.56	0.20	0.30

StrongSell			
Classifier	Recall	Precision	F1
Naive Bayes	0.86	0.76	0.82
Decision Tree	0.87	0.79	0.83
Bagging	0.97	0.77	0.86

Table 2. StrongBuy & StrongSell sentiment prediction for ExxonMobile.

StrongBuy			
Classifier	Recall	Precision	F1
Naive Bayes	0.78	0.65	0.70
Decision Tree	0.76	0.67	0.76
Bagging	0.71	0.66	0.68

StrongSell			
Classifier	Recall	Precision	F1
Naive Bayes	0.03	0.17	0.05
Decision Tree	0.24	0.30	0.26
Bagging	0.21	0.35	0.26

Table 3. StrongBuy & StrongSell sentiment prediction for Startbucks.

StrongBuy			
Classifier	Recall	Precision	F1
Naive Bayes	0.84	0.70	0.76
Decision Tree	0.81	0.84	0.82
Bagging	0.82	0.83	0.82

StrongSell			
Classifier	Recall	Precision	F1
Naive Bayes	0.41	0.61	0.49
Decision Tree	0.74	0.61	0.67
Bagging	0.76	0.64	0.68

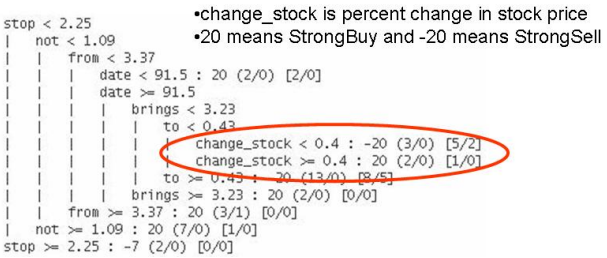

Figure 4. Sentiment correlated with change in stock price.

the classifier is able to predict all the message for a particular sentiment quite accurately though it might produce a few false positives.

In our decision tree model for Apple's sentiment, a feature near the root of the tree was the word "sheep". At first this word seemed to be completely unrelated to the stock market or Apple. We were puzzled why it played such a big roll in classifying negative sentiments. After some research, we learned that in a popular 1984 movie about the stock market — Wall Street, the main character used the word sheep to describe weak fund managers. The entire phrase was "they're sheep, and sheep get slaughtered". We then confirmed the validity of our model when we found the word "slaughtered" a few features away from "sheep". We also observed that when the word "sheep" was present, the classifier predicted a negative sentiment.

In our Starbucks sentiment model, phrases "china" and "interested" were important features to indicate negative sentiment. With growing Chinese economy and Starbucks' interest in strong presence in China, it was suprising to see these phrases negatively associated with the stock. But a quick search in recent news reveal the Chinese bloggers have expressed extreme negative opinions in Starbucks' interest in setting up store locations inside the Imperial Forbidden City. This proves our models were able to discover recent news related to the stock. We also found positive adjectives associates with positive sentiment. For example, when building a decision tree for Apple stock, we found that if the authors used words such as "keep", "good", "relax", "announced", "going over" then the sentiment was also positive. Similarly, if the word "dividends" was used while discussing ExxonMobile then the sentiment was again positive.

Our results showed that web sentiment prediction can be done with relatively high accuracy by building a model from the message boards. It also gives us interesting insights into the various things people are discussing about a stock and how it is effecting their opinion. Message boards can be a useful corpus for companies to assess their perception among the public.

Table 4. Accuracy results for stock prediction with and without TrustValue.

COMPANY	CLASSIFIER	SENTIMENT	W/TRUST
APPLE	NAIVEBAYES	71	79
APPLE	DECISIONTREE	72	81
STARBUCKS	NAIVEBAYES	70	69
STARBUCKS	DECISIONTREE	70	71
EXXONMOBILE	NAIVEBAYES	61	62
EXXONMOBILE	DECISIONTREE	61	63

4.2.2 Stock Prediction

The aim of this set of experiments is to find out whether there is a correlation between the sentiment, the TrustValue and the corresponding stock value. For these experiments, we did not use any other feature to prevent biasing the results. Table 4 shows our result on the 3 stocks. Our model was able to make accurate predictions, especially for Apple where it was able to achieve an accuracy of 81% using both Web sentiment and the TrustValue. The model also performed well for Starbucks and Exxon Mobile.

We also found that TrustValue is a helpful feature in stock prediction. As shown in the table, the performance of the model increased when the TrustValue is used. For Apple, the accuracy value increased by 9%. This shows TrustValue can help in removing many irrelevant or noisy sentiments.

To conclude, we shown that there is a strong correlation between stock prices and web sentiment, and one can use sentiment to make predictions about stock prices over a short term period (a day). Figure 4 shows a snapshot of the decision tree validating correlation between change in stock price to sentiment.

5 Future Work

This is an exciting new area which has a lot of potential for research. There are many things that can be done to extend our current approach. In our current approach, we have not exploited information within a sector. For example, a change in stock price of a technology stock (say Yahoo) might be a good indicator of similar trends in other stocks (e.g., Google) in the same sector. It would be interesting to train a stock prediction classifier using information within a sector, in other word, a more specialized classifier. One can go even further and use information across related sectors [7]. For example, automobile industry is strongly linked with performance of oil companies.

Another easy extension to our project would be to extend the model to make long term predictions. It is difficult to say how well this approach would perform but it would definitely give insight into how sentiments and stock values change over time.

6 Conclusion

In this paper, we introduced a novel method to predict sentiment about stock using financial message boards. In our experiments, we found that sentiment can be predicted with high precision and recall. We also found interesting relationships between message text used in the message board to the the corresponding sentimemts.

Web financial information is not always reliable. To take this into consideration, we introduced a new measurement known as TrustValue which takes into account the trustworthiness of an author. We showed that TrustValue improves prediction accuracy by filtering irrelevant or noisy sentiments.

We used the sentiment and TrustValue to make our model for stock prediction. We used the intiution that sentiments effect stock performance over short time period and we captured this with Markov model. Our stock prediction results showed that sentiment and stock value are closely related and web sentiment can be used to predict stock behavior with seasonable accuracy. To conclude, our results showed promising prospects for automatic stock market predictions using web sentiments.

References

[1] Weka: Data mining software in java.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24:123—140, 1996.

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees.

[4] S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *8th Asia Pacific Finance Association Annual Conference*, 2001.

[5] V. Lavrenko, M. Scmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allen. Mining of concurrent text and time series. *Proceedings of the Internation Conference on KDD Text Mining workshop*, 2001.

[6] J. Nasukawa, T. Bunescu, R. Niblack, and W. Yi. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. *Proceedings of International Conference of Data Mining*, 2003.

[7] S. Sarawagi, S. Chakrabarti, and S. Godbole. Cross-training: Learning probabilistic mapping between topics. *Special Interest Group KDD*, 2003.

[8] P. Wysocki. Cheap talk on the web: The determinant of postings on stock message boards. *University of Michigan Business School*, (98025), 1998.