Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement

Ray Chen, Marius Lazer

Abstract

In this paper, we investigate the relationship between Twitter feed content and stock market movement. Specifically, we wish to see if, and how well, sentiment information extracted from these feeds can be used to predict future shifts in prices. To answer this question, we construct a model, estimate its accuracy, and put it to the test on real market data using a mock portfolio. Our results indicate that the model is successful in generating additional profit.

1 Introduction

Historically, stock market movements have been highly unpredictable. With the advent of technological advances over the past two decades, financial institutions and researchers have developed computerized mathematical models to maximize their returns while minimizing their risk. One recent model by Johan Bollen [1] involves analyzing the public's emotional states, represented by Twitter feeds, in order to predict the market.

The state of the art in sentiment analysis suggests there are 6 important mood states that enable the prediction of mood in the general public. The prediction of mood uses the sentiment word lists obtained in various sources where general state of mood can be found using such word list or emotion tokens. With the number of tweets posted on Twitter, it is believed that the general state of mood can be predicted with certain statistical significance.

According to Bollen's paper, Twitter sentiment is correlated with the market, preceding it by a few days. Specifically, the Google Profile of Mood States' (GPOMS) 'calm' state proved to be a reliable predictor of the market. Due to the proprietary nature of the GPOMS algorithm, we wish to see if a simpler method could provide similar results, while still being able to make accurate enough predictions to be profitable.

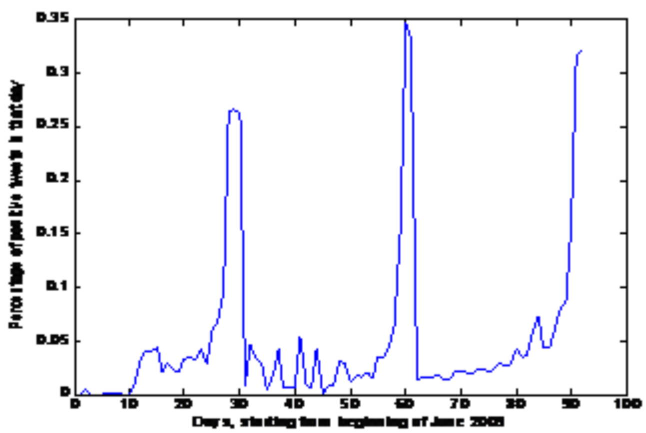
2 Sentiment Analysis

We begin our sentiment analysis by applying Alex Davies' word list [2] in order to see if a simple approach is sufficient enough to correlate to market movement. For this, we use a pre-generated word list of roughly five thousand common words along with log probabilities of 'happy' or 'sad' associated with the respective words.

The process works as follows. First, each tweet is tokenized into a word list. The parsing algorithm separates the tweets using whitespace and punctuation, while accounting for common syntax found in tweets, such as URLs and emoticons. Next, we look up each token's log-probability in the word list; as the word list is not comprehensive, we choose to ignore words that do not appear in the list. The log probabilities of each token was simply added to determine the probability of 'happy' and 'sad' for the entire tweet. These were then averaged per day to obtain a daily sentiment value.

As expected, this method resulted in highly uncorrelated data (with correlation coefficients of almost zero). We tried to improve this by using a more comprehensive and accurate dictionary for positive and negative sentiments. Specifically, we swapped our initial word list with a sentiment score list we generated using SentiWordNet [3], which consisted of over 400 thousand words. Since this list considers relationships between each word and includes multi-word expressions, it provided better results (see next sections).

We also tried representing the daily sentiment value in a different way - instead of averaging the probabilities of each tweet, we counted the frequency of 'happy' tweets (such as using a threshold probability of above 0.5 for happy) and represented this as a percentage of all tweets for that day. While this did not improve the output's correlation with stock market data, it did provide us with more insight into our Twitter data. For example, we see a spike in the percentage 'happy' tweets toward the end of each month (Figure 1). We did not find news events which could have caused these spikes; however, upon investigating the source of Twitter data, we found that it had been pre-filtered for a previous research project [4] (i.e. there may be some bias in what we assumed to be raw Twitter data). Due to a lack of access to better Twitter data, we conclude that using the frequency of happy tweets is not a reliable indicator of sentiment for our application and revert to our averaging method.



3 Constructing the Model

In this section, we discuss the construction of our model, from choosing an appropriate algorithm to finding a suitable set of features, and provide justification for these decisions.

3.1 The Algorithm

We chose to model the data using a linear regression. This decision was motivated by several factors:

- Speed - A fast, efficient algorithm was one of our original specifications. This is a must when working with massive amounts of data in real time, as is the case in the stock market.
- Regression - We sought to be able to make investment decisions not only on direction of market movement, but also to quantify this movement. A simple classifier was insufficient for this; we required a regressor.
- Accurate - Naturally, we needed an algorithm that would model the data as accurately as possible. Since our data is, by its nature, very noisy, we chose a simple model to avoid high variance.

3.2 Features

The backbone of our algorithm was, of course, Twitter sentiment data. As such, we designed several features that correspond to these sentiment values at various time-delays to the present. Training in one-dimensional feature space using only this data, we found that the best results were obtained when the Twitter data predated the market by 3 days. Using k-fold cross-validation to quantify our accuracy, we observed that this model was able to make predictions with approximately 60% accuracy, a modest improvement over no information (50% accuracy), but we wanted to see if we could do better.

We designed 2 more classes of features to try: one modeling the change in price of the market each day at various time-delays, the other modeling the total change in price of the market over the past n days. To help us choose a good set of features, we applied a feature selection algorithm using forward search to the problem. From this, we learned that the 'change in price 3 days ago' feature improved our previous model to one with approximately 64% accuracy.

Further tests indicated that several of the other features are also relevant, however, due to relatively small amount of training data (72 days or fewer), training in higher-dimensional feature spaces yielded worse results in practice. Nonetheless, with the availability of more training data, a more complex and diverse set of features could further improve accuracy. We were able to achieve, using nearly all of our available data to train (infeasible for portfolio simulation, see next section), classification accuracy as high as 70%.

4 Testing the Model

We have built a model for predicting changes in the stock market price from day to day. We have identified the accuracy-maximizing set of features and trained our model on these features. Now we must put it to the test using real-world data to determine if it is profitable. In this section, we develop 2 different investment strategies based on predictions from our model, apply them over some time period, report on the results, and compare them to 2 benchmark investment strategies.

4.1 Our Investment Strategies

Classification - The simpler of our 2 strategies considers only the predicted direction of market movement. That is, we look only at the sign of the anticipated change in price. If it is positive, we buy as many shares as possible with our current funds. Otherwise, we buy no shares, and simply sit on the money until the following day when we reevaluate the situation.

Regression - With this more complicated strategy, we seek to exploit knowledge of how much the market will change, rather than simply the direction it will shift. This allows us to base how much we invest on how certain we are of our prediction. There are countless such strategies that one could propose, we chose the following based on observations of good performance:

$$invest = \begin{cases} 100\% & \text{if } .05\% \quad (\textit{predicted \% change}) \\ 25\% & \text{if } -.1\% \quad (\textit{predicted \% change}) \quad .05\% \\ 0\% & \text{if } (\textit{predicted \% change}) \quad -.1\% \end{cases}$$

Here, *invest* is the percent of our funds we use to buy stock and (*predicted % change*) is computed by dividing the predicted change in the market tomorrow by the price today.

4.2 The Benchmark Investment Strategies

Default - This strategy uses no information about the market, and will simply buy as many shares as possible each day.

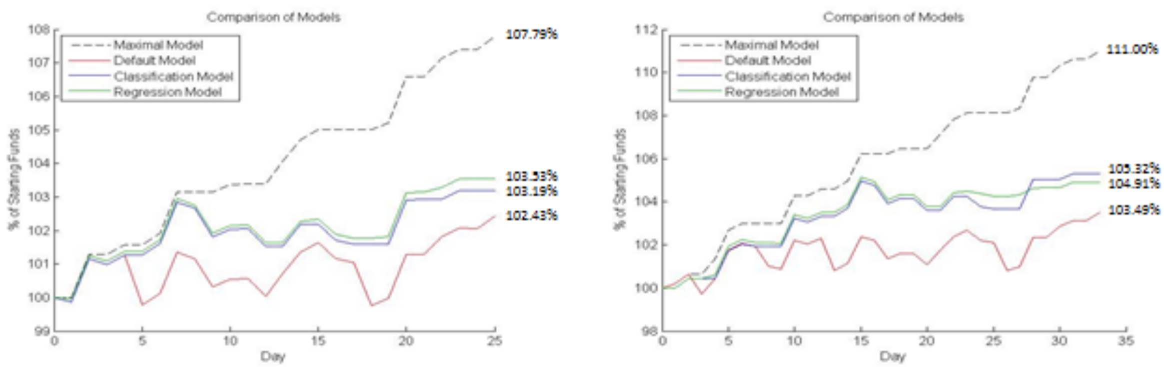
Maximal - This strategy assumes perfect knowledge about future stock prices. We will invest all available funds when we know the market will go up the following day, and invest no funds when we know the market will go down. This strategy is, of course, impossible to execute in reality, and is only being used to quantify the profits from an ideal strategy.

4.3 Simulation

We start with exactly enough money to buy 50 shares of stock on the first day. Note that since we output results as percentages of starting money, they do not depend on this value, and as such it is chosen arbitrarily. At the start of each day, we make a prediction and invest according to some strategy. At the end of the day, we sell all shares at closing price and put the money in this bank. This is done so that any gains or losses can affect future gains or losses by virtue of being able to purchase more or less stock at every time step.

4.4 Results

We ran the simulation for each investment strategy, as described above, on 2 different time intervals. The results are shown below:



In the figure on the left, we trained on about $\frac{3}{4}$ of the data (72 days) and simulated on about $\frac{1}{4}$ of the data (25 days). In the figure on the right, we trained on about $\frac{2}{3}$ of the data (64 days) and simulated on about $\frac{1}{3}$ of the data (33 days). We immediately see that both of our strategies fare better than the default strategy in both simulations.

Note, however, that the regression strategy is more profitable in the first simulation while the classification strategy is more profitable in the second simulation. We observe that on the simulation in which the model was given less training data (figure on the right), on day 27, our regression strategy opted to invest only 25% of funds that day because it perceived gains as being uncertain. This did not happen on the corresponding day in the first simulation (with more training data). Indeed, with less data to train on, imprecision in our model resulted in a poor investment decision when using the more complex regression strategy. In general, the classification strategy tends to be more consistent, while the regression strategy, though theoretically more profitable, is also more sensitive to noise in the model.

5 Conclusion

We have corroborated the results of the original paper and shown that, even with much simpler sentiment analysis methods, a correlation between Twitter sentiment data and stock market movement can be seen. We have discovered that the best results arise when Twitter data predates the market data by about 3 days and created a model capable of making predictions based on this data. Furthermore, despite the fact that the correlation is often loose, it is possible to use this model to derive profitable investment strategies, 2 of which are also presented here. We hope this lays the foundation for further research in algorithm trading methods that incorporate sentiment data to increase accuracy and, ultimately, profits.

6 Acknowledgements

We would like to thank Mihai Surdeanu and John Bauer for the project idea, general guidance, and providing us with the Twitter data.

7 References

[1] J. Bollen, H. Mao, X. Zeng. Twitter Mood Predicts the Stock Marker.
arXiv: http://arxiv.org/abs/1010.3003

[2] Alex Davies. A Word List for Sentiment Analysis of Twitter.
http://alexdavies.net/2011/10/word-lists-for-sentiment-analysis-of-twitter/

[3] SentiWordNet. An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.
http://sentiwordnet.isti.cnr.it/

[4] Jure Lescovic, Twitter Data. http://snap.stanford.edu/data/twitter7.html