CS 229, Autumn 2011
Modeling the Stock Market Using Twitter Sentiment Analysis

Team members: Daniel Debbini, Philippe Estin, Maxime Goutagny
Supervisor: Mihai Surdeanu (with John Bauer)

1 Introduction

This project is based on Bollen, Mao and Zeng's *Twitter mood predicts the stock market*, which shows that sentiment extracted from Twitter tweets can be used efficiently to enhance forecasts on the direction of stock market moves in the short term [1]. They claim an 87.6% accuracy in predicting stock market moves (up or down). Our objective will thus be to implement a trading mechanism based on Twitter data and past stock price time series, and to compare the performance of Twitter-enhanced strategies with more traditional algorithmic strategies.

2 Data

Courtesy of Jure Lescovic, we have a sample of c. 480 million tweets from June 2009 to December 2009 [2]. Each individual tweet is composed of three items: the time stamp of the tweet, the user who posted the tweet, and the text body of the tweet. Further, we have time series data of the Dow Jones Industrial Average (DJIA), the S&P 500, and individual stock prices during the second half of 2009 (provided by our supervisor).

3 Trading Strategies

Below, we consider three traditional trading strategies.

1. *X* Filter rule. If the price closes at least *x*% up, the strategy consist in opening a long position (buy) and hold until the price moves down at least *x*% from a subsequent high. In this case, you go short (sell) and maintain the short position until the price rises at least *x*% above a subsequent low. Moves of less than *x*% in either direction are ignored.
2. Moving average rule. Let $ma_t = \frac{1}{L}\sum_{i=0}^{L-1} p_{t-i}$, use the strategy to buy if $p_t > ma_t$ and to sell if $p_t < ma_t$. The optimal L needs to be determined.
3. Resistance support rule. Buy if $pt > \max_{1\le i<L} p_{t-i}$ and sell if $pt < \min_{1\le i<L} p_{t-i}$.

Each of these strategies relies on a parameter. We considered a set of parameters, fit each model to a training set (DJIA prices from January 2000 to October 2009) and tested on a test set (DJIA prices from November 2009 to December 2009). The results of the best training fit are in the table on the next page (Table 1). Correlation between the training performance and the test performance is low. This could be due to the non-stationarity of the DJIA process and suggests that we may benefit from a Twitter-enhanced strategy using additional information.

Table 1. Traditional trading strategies: best fit over training set and test performance.

Trading strategy	Parameter	Annual Performance, Training	Annual Performance, Test
X Filter	4.84%	8.37%	0.00%
Moving Average	Lag = 1	-2.78%	-32.61%
Resistance Support	Lag = 1	0.68%	-4.13%

4 Initial Attempt

In Bollen et al., two sentiment analysis tools are applied to tweets [1]. The first is called OpinionFinder which is designed to identify whether sentences are emotionally positive or negative. The second is called GPOMS, which attempts to measure 6 mood dimensions: calm, alert, sure, vital, kind, and happy.

As a first try to capture the sentiment of our tweets, we used Alex Davies' Twitter sentiment analysis word list [3]. For two sentiments, happy and sad, the word list gives the log probability of the word and the sentiment. More formally, for each word, we have an estimate of:

$$\log\big(p(w,s)\big), \qquad w = word, s = sentiment \in \{happy, sad\}$$

Then, Davies proposes a method for estimating the probability that the tweet is happy given the words of the tweet, assuming the prior probabilities of each sentiment are equal [2] (see below), under Naïve Bayes. Assuming a tweet *t* is composed of words *w*:

$$p(happy|t) = \frac{p(happy,t)}{p(t)} = \frac{p(t|happy)p(happy)}{p(t|happy)p(happy) + p(t|sad)p(sad)}$$

$$= \frac{1}{1 + \exp\big(\sum_{w \in t}\big[log\big(p(w,sad)\big) - log\big(p(w,happy)\big)\big]\big)}$$

Thus, using the Twitter sentiment word list, we can compute the probability of each tweet being happy or sad using the above formula. We can then threshold this probability to determine if a tweet is happy, sad, or neutral.

Tweets are split into words, or "tokenized," by separating white space. In this way, we still preserve emoticons such as ":)" and "<3", but tokens like "happy" and "happy!" are different.

Using the strategy above, we parsed the tweets and then performed sentiment analysis using a threshold of 0.5 (e.g. if P(t, happy) > 0.5, classify t as happy). An excerpt of our results is shown in Table 2 (on the next page). It seems using the current word list, we are classifying almost the same fraction of tweets per day as happy/sad (the numbers don't add to 100% because some tweets are composed of words not found in our dictionary, and were designated "neutral"). We also tried using different thresholds (e.g. 0.7), but the results were extremely similar.

Table 2. Sentiment analysis performed on a sample of tweets using Alex Davies' sentiment word list, showing the DJIA return and a flag showing whether the market went up. Threshold probability: 0.5.

Date	Percentage of happy tweets	Percentage of sad tweets	Number of Tweets	DJIA return
6/15/2009	66.09%	1.67%	1,055,053	-2.13%
6/16/2009	66.12%	1.65%	982,576	-1.25%

6/17/2009	66.79%	1.63%	864,568	-0.09%
6/18/2009	66.76%	1.67%	819,349	0.69%
6/19/2009	66.38%	1.56%	768,976	-0.19%

5 Tokenization

The first step in extracting better information from Twitter was to improve the parsing of tweets, or "tokenization." Using advice from Christopher Potts' sentiment tutorial [4], we improved the tokenizer by recognizing the following special character strings:

- Phone numbers
- Emoticons
- HTML tags, entities
- Twitter usernames and hashtags
- Websites
- Words with hyphens, dashes, apostrophes, or underscores
- Ellipses
- Decimals and fractions
- Words in all capital letters

Some of the above special cases are worth discussing below. "Hashtags" in Twitter are words preceded by '#' which mark keywords or topics in a tweet. Words in all capital letters usually convey heightened emotion, and thus we preserve words in all capital letters.

One final preprocessing step we used per Christopher Potts was to recognize lengthening by character repetition [4]. From Potts, lengthening "is a reliable indicator of heightened emotion… sequences of three or more identical letters in a row are basically unattested in the standard lexicon, so such sequences are very likely to be lengthening." Thus, we mapped sequences of length 3 or greater to sequences of length 3. For example:

`yaaaaaaaaaaay` becomes `yaaay`
`hahahahahaha` becomes `hahaha`
`lolololololololol` becomes `lololol`

We did not shorten punctuation that was lengthened (e.g. five multiple exclamation marks in a row were treated as five distinct tokens).

Using a tokenizer with the above competency, we parsed all tweets between July 2009 and December 2009. The next step is to extract features from these tokens so that we may fit a model to predict stock market moves.

6 Feature Selection and Mutual Information

Rather than relying on pre-trained lexicons, our method aimed at selecting tokens that have impact on the stock market in a "natural" and automatic fashion. For each day, we compute the occurrences of every token over the two previous days. A token is said to have been "frequent" over the past two days if it is in the top 1,000 occurrences in the past two days. Furthermore, for each day we compute the daily return of the DJIA. A day is said to have "large variation" if the absolute DJIA return on that day is greater than or equal to 1%. Define the indicators:

$$y^{(i)} = \mathbb{1}_{\{\text{day } i \text{ has large variation}\}}$$
$$x_j^{(i)} = \mathbb{1}_{\{\text{token } j \text{ is frequent over days } i-1, i-2\}}$$

We can then compute the mutual information (MI) between x_j and y [5]:

$$\text{MI}(x_j, y) = \sum_{x_j \in \{0,1\}} \sum_{y \in \{0,1\}} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j) p(y)}$$

The probabilities in the above formula can be estimated according to their empirical distributions on the training set.

When fitting models to predict the DJIA move (up or down) on day *i*, the feature associated with a token is the frequency of that token over the past two days (number of occurrences divided by total number of tokens over the past two days). However, since we have such a large number of tokens, we will only consider the tokens with the top *k* mutual information scores (see below for the selection of *k*).

7 Models and Results

Using the features described in the previous section, we use both logistic regression and a support vector machine (SVM) to predict the DJIA move (up or down) [6]. The training set was the first 80 trading days between June and December 2009, the test set was the last 57 trading days. For the SVM, we used the Gaussian kernel shown below:

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right) = \exp(-\gamma \|x - z\|^2)$$

The only parameter for logistic regression is *k*, the number of features used (see previous section). We chose *k* based on 5-fold cross validation ($k = 5 \ldots 300$), yielding $k = 55$. For the SVM, the parameters to fit are γ, *C*, and *k*. Gamma (γ) parameterizes the kernel and *C* is the cost parameter associated with the regularization of the SVM. The results are shown in the below figures and table.

Figure 1. Sample tokens used in final SVM model (separated by commas). :-), NEW, CAN'T, win, amazing, LMAO, $, :(, +, MONEY, GOD, facebook, bad, friends, #IRANELECTION, summer, hahaha, news, home, BUSINESS, love, JOB, TIMES, game, OFF, NICE, stop.

We see on the graph at the right that on the test set (Nov, Dec 2009), the logistic regression strategy is superimposed with the Dow Jones, while the SVM strategy performs better.

Figure 2. Performance of SVM and logistic regression models

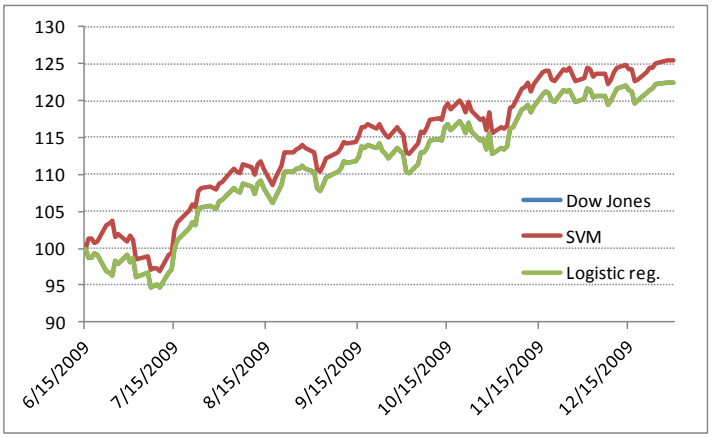

Table 3. Performance

	Train Error	Test Error	Test Set Annualized Return
SVM	0.39	0.37	39%
L.R.	0.43	0.39	34%

8 Conclusions and Future Work

Twitter is a treasure trove of information. With millions of users posting tweets every second, there are opportunities to capture public sentiment/confidence/anxiety. While the SVM performance (63% accuracy on the test set) is not as impressive as in Bollen et al. [1], the annualized performance of the SVM is strong (39% annualized return). The number of tokens used in the model (55) is reasonable.

It is yet to be determined if our accuracy can be sustained over a longer period. Our test set is larger than Bollen et al.'s one (two months against 19 days) but the significance of the results would benefit from being extended to at least a year. Moreover, the Dow Jones performed well in the second half of 2009: the strategy should be tested in stress times as well. We hope our framework can be extended to make more complex predictions about the DJIA instead of a binary up/down decision, using softmax regression and/or a multi-class SVM. Furthermore, we wish to integrate the Twitter data with other more traditional statistical features into one model (e.g. contrarian strategies with Twitter information). Additionally, work needs to be done to analyze the effect of transaction costs and mistiming of trade execution (e.g. we may not be able to buy the DJIA during the open at yesterday's closing price).

9 References

[1] Bollen, Mao, and Zeng. *Twitter mood predicts the stock market*. Journal of Computational Science, 2 (1), March 2011.

[2] J. Leskovec, J. Yang. *Temporal Variation in Online Media*. ACM International Conference on Web Search and Data Mining (WSDM '11), 2011.

[3] Alex Davies. "A word list for sentiment analysis of Twitter." Web. 18 November 2011 <http://alexdavies.net/2011/10/word-lists-for-sentiment-analysis-of-twitter/>.

[4] Christopher Potts. "Sentiment Symposium Tutorial." Web. 18 November 2011. <http://sentiment.christopherpotts.net>.

[5] Andrew Ng. "CS229 Lecure Notes. Regularization and model selection." Web. 15 December 2011. <http://cs229.stanford.edu/notes/cs229-notes5.pdf>.

[6] Chang, Chih-Chung and Lin, Chih-Jen. "LIBSVM – A library for Support Vector Machines." Web. 15 December 2011. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.