Can Twitter predict the stock market?

Volodymyr Kuleshov

December 16, 2011

1 Introduction

Last year, in a famous paper, Bollen et al. (2010) made the claim that Twitter mood is correlated with the Dow Jones Industrial Average (DJIA), and that it can be used to forecast the direction of DJIA changes with 87% accuracy. Besides its obvious significance in investing, this surprising result challenges several fundamental notions in the social sciences, such as the efficient market hypothesis[1].

In this project, I verify whether the surprising results of Bollen et al. can be reproduced and whether they can produce a profitable investment strategy. Unfortunately, I find that measuring Twitter mood does not offer an improvement over a learning algorithm that only uses past DJIA values.

2 Background

Bollen et al. (2010) measure Twitter mood according to six dimensions (calm, alert, sure vital, kind, happy) by counting how often people tweet certain words. These words are taken from a popular psychometric test called "Profile of Mood States (Bipolar)" (POMS-bi). They find that the mood dimension "calm" is correlated with the DJIA at $p < 0.05$ and that a Self-Organizing Fuzzy Neural Network (SOFNN) that receives as inputs the DJIA and the calmness scores for the past three days predicts the direction of change of the DJIA on the following day with an 87% accuracy.

3 Methods

I evaluate several approaches to using Twitter mood for predicting the DJIA, including that of Bollen et al. I start with a dataset of about 30GB of tweets from June to December 2010. I use August-October (72 weekdays) for training and November-December (33 weekdays) for testing. The months of June and July are discarded because they contain much fewer tweets that the later months (including several days with almost no tweets). Using June and July skews the normalization of inputs to the learning algorithms and results in significantly worse performance.

I parse the data using a sentiment-aware tokenizer that preserves Twitter symbols (@,#), smileys, and that turns into standard form repeated punctuation marks (e.g. "!!!!").

[1]The efficient market hypothesis states states market prices are nothing more than a rational aggregate of factual information about a good.

3.1 Reproducing the approach of Bollen et al.

Since, Bollen et al. do not clearly describe their methods, I implement a close approximation to their approach. The most important missing information is the the POMS-bi vocabulary (only the regular POMS vocabulary is publicly available), which forces me to define my own word list. I perform a 2-step WordNet propagation starting from synonyms of "calm" and "excited" and from POMS (regular) words related to calmness and excitedness. I then discard all words that do not describe mood to obtain two sets V_c, V_e of adjectives related to calmness and excitedness. The two sets contain about 50 words in total, which is close to the number that Bollen et al. used in each mood dimension.

Using V_c and V_e, I define for every day the following mood features. Given a day i and a vocabulary V, let $p_i(V)$ denote the percentage of tweeted words on day i that are in V. Also, let d_i denote the percentage change in DJIA on day i: $(\text{DJIA}_i - \text{DJIA}_{i-1})/\text{DJIA}_{i-1}$. To every day i in the dataset, I associate the nine features $\bigcup_{j=i-3}^{i-1}\{d_j, p_j(V_c), p_j(V_e)\}$ and a target output of d_i.

In order to predict the DJIA percentage changes d_i, I use a neural net (NN) instead of the SOFNN of Bollen et al. Specifically, I train a perceptron using backpropagation for about 20,000 epochs (roughly, until convergence). I also experimented with multi-layer networks, but they would usually overfit the training set. All inputs to the perceptron are normalized to have mean zero and standard deviation one. Like Bollen et al., I train only on tweets that include phrases like "I feel", "I am", etc. Training on all tweets did not improve performance.

Besides using a different vocabulary and and a different learning algorithm, the above method completely replicates the approach of Bollen et al.

3.2 SVM classification

Since the above method does not come close to achieving the desired 87% accuracy, I propose and evaluate alternative ways of using Twitter mood to predict the DJIA. First, I focus on the simpler problem of classifying the direction of DJIA movements and use an SVM as my classifier. Besides often working well in practice, SVMs admit regularization parameters that can reduce the high variance I observed with neural nets. I use a Gaussian kernel for the SVM and I select all hyperparameters through cross-validation. I normalize all features so that over the training set, they fall precisely in $[0, 1]$. I also measure sentiment over all tweets; focusing only on tweets containing phrases like "I feel" did not produce better results.

I separate days into classes in two ways: into up/down classes, and into up/stable/down, where stable is defined as a percentage change of less than 0.2%. There were about 10 stable days in the test set.

Finally, I collect and feed mood data into the SVM using two methods.

3.2.1 Vocabulary-based

The first mimics the algorithm of Bollen et al., except that it replaces the calmness and excitedness vocabularies V_c and V_e by general vocabularies V^+ and V^- of positive and negative words. As before, the SVM receives as inputs percentages $p_i(V^+)$, $p_i(V^-)$ for the past three days. I try two approaches to constructing the vocabularies V^+ and V^-: greedy forward model building and information gain.

In the greedy model-building approach, I start with two larger sets S^+, S^- of about 100 terms each that I build using WordNet propagation. Words in the sets S^+, S^- are respectively positively and negatively associated with either "calmness" or "happiness", the two dimensions that correlate with the DJIA according to Bollen et al. Based on S^+, S^-, I greedily build V^+ and V^- by iteratively adding to its corresponding set the word that produces the largest increase in cross-validation accuracy.

In the information gain approach, I associate to each word $w \in S^+ \cup S^-$ a variable X_w that takes one of 3 values: low, medium or high. The variable X_w takes the value "low" (resp. "med.", "high") when the $[0,1]$-normalized percentage of tweeted words on day i that equal w falls in $[0, 1/3)$ (resp. $[1/3, 2/3)$, $[2/3, 1]$). To find words that correlate with DJIA movements, I compute the information gain of sign(ΔDJIA) and X_w, and define V^+, V^- to be the sets of words in S^+ and S^- that have an information gain greater than some $g > 0$. I experimented with $g = 0.4, 0.25, 0.15, 0.08$.

For up/stable/down classification, I calculate the IG between X_w and a target variable that can take the three possible class labels.

3.2.2 Word-based

The second approach is to directly feed the SVM percentages $p_i(\{w\})$ for all words w in a vocabulary V. To obtain the vocabularies V, I use the the same methods as in the previous section. In greedy model-building, I iteratively add to V the word in $S^+ \cup S^-$ that yields the largest increase in cross-validation accuracy. The information gain approach is identical to the one outlined above.

3.3 SVM Regression

It is usually more important to correctly identify large DJIA movements than smaller ones, since they produce high profits or losses. Therefore it is worth trying to predict the actual value of d_i, rather than only its sign. Although neural nets have been applied to that problem in Section 3.1, I also consider predicting the d_i using SVM regression, since that algorithm allows for regularization and can be easily combined with my model selection techniques for classification. I use the same inputs to the SVM and the same model selection algorithms as in the classification setting. See Section 3.2 for details.

4 Results

4.1 Reproducing the approach of Bollen et al.

The approach described in Section 3.1 yielded a test accuracy of 67% on the direction of DJIA movements. However, this is almost certainly due to overfitting, as I experimented with slight variants (e.g. a slightly different vocabulary) and none had a higher accuracy than 62%. Moreover, training an SVM on exactly the same inputs also resulted in only a 62% accuracy.

4.2 SVM Classification

Overall, SVM classification yielded accuracies of approximately 60%. Although this may seem significant, this accuracy can be achieved simply by predicting "up" constantly. In fact, almost

all SVM classifiers learned to do precisely that, and classifiers that scored higher than 60% simply predicted one or two correct "downs" in addition to classifying everything else as an "up". Most notably, an SVM that received as inputs only the DJIA percentage changes for the past three days would always learn to do precisely that. Therefore, the results for classification cannot be considered significantly better than this baseline approach.

4.2.1 Vocabulary-based approaches

The table below presents the accuracy of algorithms described in Section 3.2.1. Each cell contains two numbers: the first is the cross-validation accuracy, the second is the test set accuracy. Note that greedy model building clearly overfits the training set.

	Greedy search	IG > 0.08	IG > 0.15	IG > 0.25	$IG > 0.4$
SVM (u/d)	65%, 45%	64%, 64%	59%, 58%	61%, 61%	56%, 61%
SVM (u/s/d)	71%, 27%	51%, 52%	50%, 52%	51%, 52%	51%, 52%

4.2.2 Word-based approaches

The table below presents the accuracy of algorithms described in Section 3.2.2. The first number in a cell is the cross-validation accuracy, the second is the test set accuracy.

	Greedy search	IG > 0.08	IG > 0.15	IG > 0.25	$IG > 0.4$
SVM (u/d)	75%, 61%	56% 61%	56% 61%	56% 61%	56% 61%
SVM (u/s/d)	71%, 47%	55%, 53%	55%, 53%	55%, 53%	51%, 52%

4.3 SVM Regression

Overall, the regression problem proved to be at least as difficult as classification. Since the SVM does not explicitly focus on predicting directions of change, the accuracy is somewhat worse. Also, I did not observe better accuracy on days with large DJIA changes, and so the regression approach does not appear to be more promising than classification for practical purposes.

The table below contains test-set accuracies for regression-SVMs that are based on the vocabulary and individual-word approaches mentioned in Section 4.3.

	Greedy search	IG > 0.08	IG > 0.15	IG > 0.25	$IG > 0.4$
Voc.-based	34%	53%	55%	55%	53%
Word-based	37%	53%	51%	51%	53%

5 Discussion

Techniques very similar to those of Bollen et al., as well as several alternative methods failed to even come close to the 87% accuracy on the direction of DJIA movements described in the Bollen et al. paper. This raises doubts about their methods and the correctness of their claim.

A first hint at methodology problems comes from the 73% accuracy the authors obtain using only the three previous DJIA values. It seems that for a problem this complex, such a good accuracy is surprising given the simplicity of the inputs. Perhaps the obscure learning algorithm they use is overfitting the test set. Since the authors do not explain how they chose

the 4 algorithm hyperparameters and never mention using a validation set, my first suspicion is that these parameters were not chosen independently of the test set.

Another issue with the paper's methods is that they only report test set accuracies for the eight models they consider. The correct approach would have been compute eight validation set accuracies and report the test set accuracy of the model that performed best in validation. Otherwise, the 87% number may be due to luck: given enough models we will eventually find one with a low test error. Bollen et al. don't seem to realize this when they argue that an unbiased coin is unlikely to fall on heads 87% of the time. In their case, they are throwing eight such coins. One can check that if the coins are independent, the chance of that event happening is about 33%.

Moreover, their baseline algorithm already has a 73% accuracy, and since their test set has only 15 data points, a 14% improvement corresponds to only about 2 additional correct predictions. It does not seem unlikely that out of seven algorithms one would make 2 additional correct predictions purely by chance (especially since the accuracies of the other models seem to be scattered randomly around 73%). In fact, I *was* able to make two additional correct predictions over my baseline by counting sentiment words with an IG of 0.08 or more!

One more subtle mistake Bollen et al. make is to normalize the training and test set simultaneously. Since they perform a regression and not a classification, scaling the test set outputs together with the training set outputs introduces additional information into training. However, since they predict percentage changes, this may not be a big problem.

Finally, in the Granger causality analysis section, the authors again make the mistake of not correcting for multiple hypothesis testing. Although the probability of a given dimension being correlated with the Dow Jones is small, the probability that one out of six is correlated will be higher.

6 Conclusion

Given my results, the answer to the question of whether Twitter can predict the stock market is currently "no". Moreover, my algorithms achieve about a 60% accuracy by always predicting that the DJIA will go up, and therefore obviously cannot be used for constructing a portfolio that would outperform a financial instrument that follows the DJIA.

The methodology problems of Bollen et al., and the fact that several groups were unable to replicate their accuracy raise serious concerns about the validity of these authors' results. Given the boldness of their claims, I believe they ought to either publish their methods and their code, or withdraw these claims.