Stock Price Prediction

Runyu Qian RQIAN@G.UCLA.EDU
Department of Mathematics
University of California, Los Angeles
CA 90095-1555, U.S.

Hao Wang HAOWANGLUD@G.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
CA 90095, U.S.

Xuening Wang SHERRY9788@G.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
CA 90095, U.S.

Shuoyi Wei SHUOYI@G.UCLA.EDU
Department of Mathematics and Department of Economics
University of California, Los Angeles
CA 90095-1555, U.S.

Pei Zhou ZPCS03@G.UCLA.EDU
Department of Mathematics
University of California, Los Angeles
CA 90095, U.S.

Editor: Justin Wood

Abstract

This paper investigate the ability of Twitter posts (tweets) in predicting stock price. With crawled tweets on a specific industry/company as the data set, we employ word2vec to convert words in tweets to vectors. We use manually labels tweets to train sentiment classifiers, including Logistic Regression, Random Forest Algorithm and CNN, to categorize tweets with labels: Positive, Negative and Neutral. Using the results of classification we train a correlation analyzer to develop the relationship between tweets and stock market movements.

Keywords: Data Mining, Natural Language Processing, Sentiment Analysis, Stock Price Prediction

1. Introduction

According to Efficient Market Hypothesis (EMH) developed by Professor Eugene Fama, it is impossible to beat the market as the stock price follows a random walk pattern and has at most 50% accuracy in predictions (Fama, 1965). For years researchers have been looking for efficient stock price predictors that are theoretically supported by behavioral economics, and have already concluded that public sentiment is a valuable indicator of stock market movements.

Twitter is a social media platform where millions of users share their real-time stories/thoughts/opinions with tweets that are limited to 140 characters. Some early works used tweets as indicators of public sentiment and have found strong correlation between the proportion of emotional tweets significantly correlated with the overall stock market movements (Dow Jones, NASDAQ and S&P 500, etc.) (Zhang et al., 2011). Bollen et al. (2011) classified sentiments to 6 different levels to deeply investigate the mechanism of Behavioral Economics. In this paper, we will filter data from Twitter by key words and categorize them into 3 levels: Positive, Negative and Neutral. We will also apply additional information beside tweets, such as number of retweets, to adjust the weights accordingly. Section 4 describes the details in data preparation and processing. Logistic Regression, Random Forest Algorithm and CNN are analyzers we plan to use in categorization of tweets. Evaluations on different methods can be found in Section 5 and will be validated and discussed further after our experimental results out.

Our goal is to construct a significant correlation between public sentiment of a specific industry/company reflected in tweets and the real-time fluctuation of this industry/companys stock price.

2. Problem definition and formalization

The problem we are going to solve is to predict the movement of one or some related stocks by the mood of public. For now, we decide to focus on the stock movement of the big brands tech companies. More specifically, we only care about the trend of stock price, but care about the exactly numerical price of these companies.

For the mood extraction part, the texts, needed to be extracted from the publics twitter posting, are those appeared three days previous to the day we want to forecast. After crawling the text from twitter, we will use Word2vec representation to transform the words

to 300 dimensional vector. The next step comes the data mining part. We are also going to manual label the sentiment of tweets to create the training and testing dataset. And, using this dataset to train the model with the help of random forest algorithm. Upon selecting the suitable model, the final step is to generate all the sentiment labels for each twitter.

Moreover, for the predicting stock price part, we will put the sentiment labels from the models to some statistic tools to find the correlation of the mood and the trend of stock price. Later, compare the trend we got and the true historical movement. Calculating the precision of our predication and trying to use it on the real world problem.

3. Data Preparation and Processing

3.1 Tweets

We stream a collection of public tweets related to target company/industry. Tweets data are crawled through the Twitter REST API, accessed through Python package tweepy. We only take into account tweets wrote in English in order to perform sentiment analysis. The metadata of each tweet, most notably includes the following features, are collected:

- Text content of the Tweet (at most 140 characters)
- Date-time of the submission (GMT+0)
- Location
- Number of favorites
- Number of followers

Text are parsed into words in order to be used as training set. Date-time are rounded into dates to match daily quotes of stock price.

3.2 Stock Price

Historical quotes for each target stock or target sector, including Open, High, Low, Close and Volume are downloaded from Yahoo Finance. All historical data are daily.

3.3 Targeted tweeter account

Targeted twitter accounts are the accounts of individuals whose tweets may affect the market in a profound way. These individuals include executives of the targeted company as well as policy maker of the related industry. Their timelines are collected and analyzed separately from the previous Twitter stream.

4. Methods Description

In this section, we describe major steps and methods used to predict stock prices from Twitter data. After processing the tweets, we label each of them manually with 1 being positive, 0 being neutral and -1 being negative. These labels serve as ground truth classficiations when we classfify future tweets to predict the stock price later.

4.1 Neural Language Model

To convert input of text data with labels to numeric values for later analysis, we used the open-source tool word2vec (Mikolov et al., 2013) to learn word embeddings in specified dimension space. The training objective of word2vec is to learn word vector representations that are good at predicting the nearby words. In the model, $v(w) \in R^d$ is the vector representation of the word $w \in W$, where W is the vocabulary and d is the embedding dimensionality. Given a pair of words (w^t, c), the probability that the word c is observed in the context of word w^t is given by,

$$P(D = 1|v(w_t), v(c)) = \frac{1}{1 + e^{-v(w^t)^T v(c)}} \tag{1}$$

The probability of not observing word c in the context of w^t is given by,

$$P(D = 0|v(w_t), v(c)) = 1 - P(D = 1|v(w_t), v(c)) \tag{2}$$

Given a training set containing the sequence of word types $w_1, w_2, ..., w_T$, the word embeddings are learned by maximizing the log likelihood.

Once we trained this neural language model on our dataset, we can obtain a unique vector representation of each word that appears in the tweets collected. And after we combine all representations of words in a tweet, we obtain an embeddding for the tweet.

4.2 Sentiment Classifier

After we feed the tweets to the neural language model specified above and get vector representations of tweets, we use these embeddings to train several classifier models for sentiment detection task since we have already labeled the tweets. The following describes some classifiers.

4.2.1 Logistic Regression

Logistic regression is a model for classifying categorical variables, in our case, labels of sentiments. It utilizes Maximum Likelihood Estimation and logistic function to classify each data point into a class based on probabilities. More specifically, We used Newton-Raphson update to train our data.

4.2.2 Random Forest Algorithm

Random Forest Algorithm perates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. We chose RF over decision tree because it corrects for decision trees' habit of overfitting to their training set.

4.2.3 Convolutionary Neural Network

We plan to use CNN (Krizhevsky et al., 2012), a Neural Network model that utilizes layers with convolving filters that are applied to local features (LeCun et al., 1998). CNN has shown great performance on task such as Computer Vision and Natural Language Processing, and we use it in the paper specifically for sentence classification task (Kim, 2014).

4.3 Stock Price Prediction

This part specifies the method used after finishing the classification task on sentiments from tweets. We plan to simply set a threshold of the proportion of positive or negative tweets for a day and predict rise or fall accordingly.

5. Experimental Results

In this section, we will study the effectiveness of vector represenation produced by the word2vec algorithm and the performance of the three classifier algorithms, namely Logistic Regression, Random Forest, and CNN. For evalutation word2vec, We will be following the evaluation method of Tomas Mikolov (Mikolov et al., 2013) with semantic and syntactic questions designed specifically for this model. For the classifer algorithms, we will be using real data sets of Twitter from XXX TO XXX ABOUT XXX where there is a significant change in the stock price of XXX to test on the precision and recall of each classifier.

5.1 Word2vec Evaluation

To evaluate the effectiveness of word2vec model, word2vec be using a comprehensive test set that contains five types of semantic questions and nine types of syntactic questions. An example of each question type is provided in the table below.

Type of Relation	Word Pair 1		Word Pair 2	
Common Capital	Beijing	China	Tokyo	Japan
All Capital Cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	RMB	China	Yen	Japan
City-in-state	Chicago	Illinois	Sacramento	California
Man-woman	brother	sister	queen	king
Adjective to Adverb	happy	happily	sad	sadly
Opposite	happy	unhappy	breakable	unbreakable
Comparative	good	better	bad	worse
Superlative	good	best	bad	worst
Present Participle	walk	walking	sleep	sleeping
Nationality Adjective	Chinese	China	Japanese	Japan
Past Tense	eat	ate	sleep	slept
Plural Form	cat	cats	index	indexes
Plural verbs	eat	eats	die	dies

First, a list of word pairs of the above types are created manually, and then a larger list of questions with only the first part of the word pair will be used to test the model. Only when the model answer's the second part of the example, we say the model makes a correct prediction. In other words, only when the closest word of the given question in the latent space produced by word2vec is exactly the same as the second part of the word pair, we consider this as a correct prediction. No sysnonums are thus considered to be correct. One thing to note is that we only consider single token words (thus words like the United States, Los Angeles are not allowed).

The accuary of word2vec is evaluated as the overall accuracy of the whole model as well as the accuracy of each type of questions.

5.2 Classifer Evaluation

To evaluate the three classifier methods, we will be using both Receiver Operating Characteristics curves and Precision-Recall method.

For the ROC curves, we will plot the truth positive rate and false positive rate and compare the area below the curve for each of the three classifiers. Truth positive rate and false positive rate's formula is given as following:

$$TruthPositiveRate = TruthPositive/Positive \tag{3}$$

$$FalsePositiveRate = FalsePositive/Negative \tag{4}$$

For the Precision-Recall method, we will be calculating the precision recall value for each of the three classifiers, where precision and recall is calculated as following:

$$Precision = TruthPositive/(TruthPositive + FalsePositive) \tag{5}$$

$$Recall = TruthPositive/(TruthPositive + FalseNegative) \tag{6}$$

6. Schedule

1. Finish data crawling for the word2vec training, labeling tweets before week 7.
2. Finish three classifier and their evaluation before week 8.
3. Finish the mapping from classified result to stock prediction before week 9.
4. Finish model adjusting before week 10.
5. Finish report.

7. Progress Discussion

1. Most codes are established and tested, however further analysis and adjustments are still needed to reach meaningful conclusion.
2. Current models show promising explanatory power, however the accuracy of prediction is yet to be tested.

References

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1): 34–105, 1965.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Xue Zhang, Hauke Fuehres, and Peter A Gloor. Predicting stock market indicators through twitter i hope it is not as bad as i fear. *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.