Machine Learning Final Project Report

Empirical Study on Stock Market Prediction

Liang Shuailong 1000829

ABSTRACT

Stock price prediction based on history news is an interesting topic and has been researched a lot. However, previous study of stock market prediction relies on shallow features such as Bag of Words, named entity and noun phrases. However, the limitations of these features are that the complex hidden relations can not be well represented. Now with the help of OpenIE technology, it is convenient to extract event information automatically without manual effort. The paper Using Structured Events to Predict Price Movement investigation the possibility to use event feature to predict the price movement compared with the basic Bag of Words features. In addition, SVM as a linear model and Neural Network as a non linear model are also used to compare their respective performance. This project mainly implements the combination of two kinds of feature and two kinds of models.

Keywords

Stock Price, Neural Network, OpenIE

1. INTRODUCTION

As is mentioned before, this project tries to use different features types (BoW, Event) and different models (SVM, NN) to predict price movement. According to the paper, Event + NN is better than Event + SVM, which is better than BoW + NN, which is better than BoW + SVM. So in this project we decided to replicate the experiment check the result.

Considering the workload, only a subset of the experiments is conducted. Specifically, the timespan is fixed at 1 day, i.e. we just want to predict the movement of the stock price based on the news on the previous day. The news sources are Bloomberg and Reuters. In this experiment we just use the result from Bloomberg. What's more, the news titles and news contents can both be used to extract features. The paper shown that using news titles can achieve a better result than using news content or using both, so we just use news titles for this project.

2. DATA PREPROCESSING

2.1 Data Description

The news dataset consists of news from Bloomberg with a time span of 2006-10-20 to 2013-11-26. There are 448395 news articles in 1041 days. We also have the stock prices information of the companies of S&P 500 Index crawled from Yahoo Finance.

2.2 Extract News According to Company

To predict stock price movement for a specific company, we need to first extract the news which is related with the company. For each news article, we just check if the full name of the company appears in the title or in the content. It is a naïve approach, since some news articles may not contain the full name of the company, but the popular name. For instance, Apple Inc. is the full name, but some news report may just use Apple to refer to the company, which we will miss in this case.

2.3 Align Stock Prices with News Articles

After extracting the news, we need to construct the labels for the news, which is +1 if on the next day the stock price rises, and -1 if on the next day the stock price falls. We just compare the open and close price of the stock on that specific day to determine its rise and fall. After that, we construct News-Label pairs as our raw dataset.

2.4 Dataset Division

For learning purpose we divide our dataset into three parts: train, dev and test, according to dates. The dates and the number of positive and negative samples are shown in Table 1.

Table 1 Dataset splitting

	train	dev	test
Time interval	2006/10/20-2012/06/18	2012/06/19-2013/02/21	2013/02/22-2013/11/21
Positive data points	590	90	109
Negative data points	451	78	74

3. FEATURE EXTRACTION

3.1 Bag of Word Feature

For Bag of Word features we use the classic "TF-IDF" score, which is defined below.

$$TFIDF = \frac{1}{|d|} freq(t_l) \times \log\left(\frac{N}{|\{d: freq(t_l) > 0\}|}\right.$$

Where $freq(t_l)$ demotes the number of occurrences of the lth word in the vocabulary in document d and N is the number of documents in the training set.

For simplicity, we use the **TfidfVectorizer** class from sklearn package to extract the TFIDF feature. The stop words are ignored during feature extraction.

3.2 Event Feature

For Event feature, we resort to OpenIE techniques. We use the tuple (O_1, P, O_2, T) to represent a event, where O_1 is the subject, *P* is the action, O_2 is the object and *T* is the timestamp. The event will be very sparse of we just use the tuple to represent it. So here backoff is used to mitigate the sparseness of the event features. The event (O_1, P, O_2, T) is represented by the combination of the elements $(O_1, P, O_2, O_1 + P, P + O_2, O_1 + P + O_2)$. To further reduce the sparseness, we can also use some popular ontologies such as WordNet and VerbNet to generalize the event. In this project we have not adopted this method yet.

After extracting the event information, use the 1-hot encoding to encode the event features.

4. MODEL SELECTION

4.1 SVM

We use SVC class from sklearn package as our classifier. To explore the optimal hyper parameter of the SVM classifier we use Grid Search techniques on the dev set. The hyper parameter includes the kernel function, the regularization term C, the γ coefficient in the RBF kernel functions, and so on. We choose the set of parameters which performs best on the dev set as our optimal parameter.

4.2 Artificial Neural Network

Since the relation between news and stock movement is very complex and may involve hidden factors, we use ANN to address the problem, in hope that the network structure of ANN can capture the hidden relations.

We use the recently open sourced Tensorflow developed by Google to build our Neural Network. The network structure is shown in Figure 1 (adopted from Ding et al. 14).

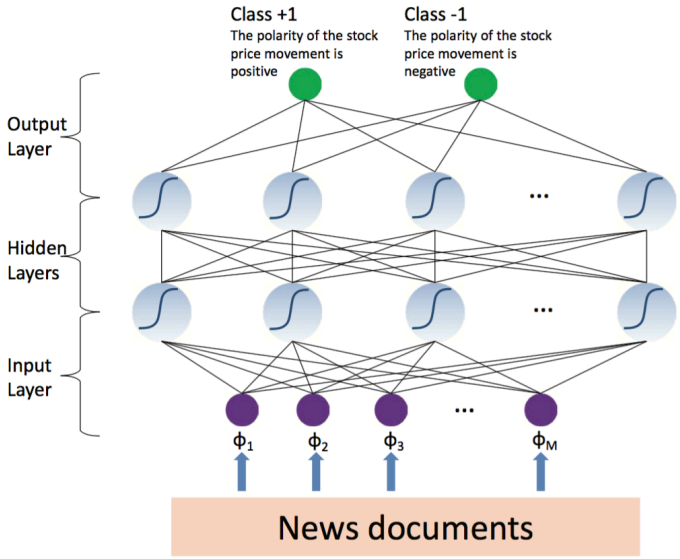

Figure 1 Structure of the deep neural network model

We set the number of nodes in layer 1 and layer 2 to both 1024. The output is used as input to the softmax function and then the cross entropy is calculated as the loss function.

5. EXPERIMENT RESULT

The experiments are run for Google Inc. using Bloomberg news and the experiment result is shown in Table 2.
Baseline accuracy is measured when the majority vote is used.

Table 2 Accuracy of different settings for Google Inc.

	Baseline	SVM + BoW	SVM + Event	NN + BoW	NN + Event
Train	52.05%	99.73%	80.27%	52.05%	63.29%
Dev	58.91%	50.39%	58.91%	58.91%	52.71%
Test	50.69%	50.00%	47.92%	49.31%	50.00%

From the table it is disappointed that our model does not even beat the baseline. The accuracy for Google is 67.86% in Ding's paper.

The result of the experiment is not satisfying due to several reasons:

1) The algorithm used to classify the news by company is not good enough so training dataset is not big enough to provide enough information;

2) The TFIDF feature and Event feature are not good enough. For IFIDF feature, we just remove the stop words and maybe we can also try normalized values and restrict the number of features to the top K to reduce the feature sparseness. For Event feature, although we use backoff to reduce the sparseness of the dataset, we did not do word stemming and aggregation. The event feature may be still too sparse.

3) For the hyper parameter tuning, we can search more precisely in the parameter space and try different network structures (different number of nodes/different number of hidden layers), use different activation functions and/or use dropout to improve the robustness of the neural network.

What's more, maybe it's better to predict the S&P 500 Index other than an individual company since there are more information. This will be done in later work.

6. CONCLUSION

Stock market prediction is a very challenging task. With the development of new technology and theory the possibility of higher accuracy prediction will increase. The value of this task is to help stake holders gain more insight from a flood of information on the Internet and utilize it to make better decisions.

Despite that the experiment result is not quite satisfying, it servers as a basic framework for my later research. In the whole experiment replication process, I realized how hard it is to reconstruct someone else's result. A good knowledge of the theory is far from enough. It requires patience to find the good feature, choose the proper model and tune the parameters.

7. ACKNOWLEDGEMENT

Thank Lilin for providing the Event Extraction Wrapper API (OpenIENER) for Python; The datasets are downloaded from Ding's website.

8. REFERENCE

[Ding *et al.*, 2014] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. In Proc. of *EMNLP*, pages 1415-1425, Doha, Qatar, October 2014. Association for Computational Linguistics.