Reproduce Stock Price Prediction Tasks with News Analysis

NYU FRE-7871 Final Project Proposal

Pei-Lun Liao
New York University
pll273@nyu.edu

ABSTRACT

In this project , selected related works will be reproduced with different financial news dataset. The goal is to understand techniques and challenges in stock price prediction.

1 INTRODUCTION

Nowadays, traders apply machine learning technique to extract information from financial news data. The information can be used to boost the accuracy of stock price prediction [4, 5, 12]. The project aims to reproduce selected related works [12]. Hence, people can learn the challenge and technique that are crucial in stock price prediction.

2 APPROACHES

2.1 Data

Data quality is crucial to make the accurate prediction [14]. Unfortunately, it is not easy to find a publicly available financial news dataset. For example, the public financial news dataset published in [4] was currently unavailable due to license issue [10]. Another corpus available online requires membership and subscription fee [11]. Also, it is not practical to crawl dataset from business news media in this one month project. Most of the time will be spent on data collecting, cleaning, and processing instead of understanding the technique that works for stock price prediction.

2.1.1 Webhose.io and Yahoo! Finance. Fortunately, an available financial news dataset was found on Webhose.io [13]. The data was crawled from the Internet from July to October in 2015. 47,851 news articles were collected in machine-readable format. However, there is no paper shows the dataset could help stock price prediction. News articles from Webhose.io come globally. To make sure the articles related to the America stock index, I filtered out the news from other countries and the articles with words number less than 100, and only pick the articles which contain the word "finance" [14].

End of date S & P 500 index from July to October in 2015 was downloaded from Yahoo! Finance [1] . However, we can't find the S & P index by minute and corresponding 500 stock price by minute. Therefore, we could not have the S & P 500 index by minute to predict next 20 minutes stock index as the experiment in AZFinText. What we can do is to predict the close time price and to see if we can improve the return rate. Since we only have a datum per day, it turns out only 76 EOD data was left. It may hurt the performance of a machine learning model.

The source code was released on https://github.com/plliao/FRE-7871-project

2.1.2 Reuters and IEX API. Due to the quality issue in Webhose.io dataset, we need to find another way to do the experiment. Hence, we crawled S & P 500 stock price from IEX API [2] and business news articles from Reuters [3] in the period between July 2nd 2018 to September 30th 2018. Due to the limited available open stock price data, I can only collect 3 months dataset. Also, the number of articles crawled is unknown, and data quality is not guaranteed as well. We will find S & P 500 company names and tickers in an article and map the article to a specific stock. Our goal is predicting the next 20 minutes stock price as described in AZFixText.

2.2 Plan

2.2.1 Stage 1: bag-of-words. The goal in stage 1 is reproducing the experiment result in AZFinText system [12]. AZFinText represented article in the bag-of-words with only proper nouns. The dataset were Yahoo Finance news articles and the S & P 500 index. Reproducing the experiment in AZFinText can prove that the data from Webhose.io also works for stock price prediction. Moreover, we examined whether only using proper nouns helps the performance.

The experiment setup will follow the AZFinText paper. We will represent articles in the bag-of-words and use SVR to predict stock price. Finally, we evaluate the result by the rate of return. The strategy is buying the stock as the predicted price is greater than or equal to 1% movement from the stock price at the time the article was released. Then, sell the stock after 20 minutes. In stage 1, we will have four different results to compare.

- 1. Stock price
- 2. Stock price + News in bag-of-words
- 3. Stock price + News in bag-of-words without stopwords
- 4. Stock price + News in bag-of-words with only proper nouns

The expected performance will be 4 > 3 > 2 > 1 as the paper described.

2.2.2 Stage 2: word and sentence representation. In this stage, We are curious how modern deep learning techniques like word2vec, seq2seq, and CNN-LSTM language models could help improve the performance.

- 5. Stock price + News feature in average word2vec [8, 9]
- 6. Stock price + News feature in Skip-Thought vector[7]
- 7. Stock price + News feature in CNN-LSTM encoding [6] (may try to find pre-trained model)

The performance in stage 2 should be better than the one in stage 1.

3 EXPERIMENT

3.1 Data processing

The articles published before the stock close time are collected. We selected the last 5 articles per day with at least 100 words each article. If the article doesn't contain the word "finance" or the article does not come from America, we filter them out. After that, we remove stop words, punctuation, number, low-frequency words, and high-frequency words in the article. The threshold of the low and high frequency are 10 and 100. We merge 5 articles bag-of-words into 1 article and normalize it with L2 norm. For the price features, I use the previous 5 days open and close price difference as features. The idea is trying to learn the trend by days. Since we only have 76 days, it is not likely to reproduce the result in AZFinText [12].

3.2 Tasks

We have three different tasks. The classification task is to predict if the close price goes up. The regression problem is to predict the price difference. The final one is applying a simple strategy to see what we can gain. The strategy is trading as the predicted price difference is positive. We use logistic regression as our classification model, ridge regression for regression task.

- Classification task: close price goes up or down
- Regression task: close and open price difference
- Trading strategy task: compute return using the prediction price

The reason for not applying the SVR model as in AZFinText is that SVR always picks the same support vectors among different features. The insufficient data may cause it. The result is not comparable among them. Also, the reason not using the same strategy as in AZFinText is for all models we could not have 1 % return. Hence, we have 0 return rate for all models and features. So the result is not comparable as well. Hence, we start with a more straightforward strategy.

3.3 Evaluation

Since the data is time series, we can't apply standard K-fold cross-validation to our model because we can have the forward bias. Hence, we use 5 fold time series cross-validation instead. We built word dictionary and scaled our features only on the observable dataset. Then, we predict the dataset on the next fold. If there are new words in the article, we ignore them. We evaluate accuracy for the classification task, mean square error for the regression task and return rate in percentage for trading strategy task.

4 RESULT

4.1 Stage 1: bag-of-words

We evaluate the performance with different features. Table 1 shows the result for using only price index features. Table 2 and table 3 shows the result for using price index features and text features. The result shows it hard to predict the future with insufficient data, and the model is not likely to make money with less than 1 % return rate. Sometimes we overfit like the case in the period of 09/04 - 09/22 and 09/23 - 10/08 in table 1. However, generally speaking, the text features could help the performance a little bit like the table 4 shows. Nonetheless, we only have 76 data. The result was not convincible.

We also find the important terms in our article by checking the weight of coefficient in our models in table 5. It gives us the explanation that why the performance between the text features with and without stopwords is similar because they have the similar vital terms in their models. However, the terms do not make much sense to me. The found words are general and not obvious to say it can make the impact on the stock price.

5 FUTURE WORK

The future works are cleaning the Reuters dataset and do the same experiment again. Also, I would like to try the technique described in AZFinText by only selecting noun words in an article as features and begin the stage 2 in my plan.

REFERENCES

[1] 2015. Yahoo! Finance. (2015). https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC

[2] 2018. IEX API. (2018). https://iextrading.com/developer/docs/#chart

[3] 2018. Reuters. (2018). https://www.reuters.com/resources/archive/us/

[4] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using Structured Events to Predict Stock Price Movement: An Empirical Investigation. In *EMNLP*.

[5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep Learning for Event-driven Stock Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 2327–2333. http://dl.acm.org/citation.cfm?id=2832415.2832572

[6] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2016. Unsupervised Learning of Sentence Representations using Convolutional Neural Networks. (11 2016).

[7] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 3294–3302. http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[10] Philippe RÃ¡my. 2014. Financial News Dataset from Bloomberg and Reuters. https://github.com/philipperemy/financial-news-dataset. (2014).

[11] Evan Sandhaus. 2008. The New York Times Annotated Corpus. https://catalog.ldc.upenn.edu/LDC2008T19. (2008).

[12] Robert P. Schumaker and Hsinchun Chen. 2009. A quantitative stock prediction system based on financial news. *Information Processing & Management* 45, 5 (2009), 571 – 583. https://doi.org/10.1016/j.ipm.2009.05.001

[13] Webhose.io. 2018. Webhose.io. Available at: https://webhose.io/datasets/ accessed on 2018-09-17. (2018).

[14] Jinjian Zhai, Nicholas Cohen, and Anand Atreya. 2011. CS224N Final Project: Sentiment analysis of news articles for financial signal prediction. (March 2011).

Prediction date	Word size	Classification (acc)	Regression (MSE)	Trading (return %)
07/31 - 08/17	-	0.687 / 0.666	119.07 / 206.77	0.172 / -0.037
08/18 - 09/03	-	0.678 / 0.833	143.70 / 1168.6	0.096 / 0.706
09/04 - 09/22	-	0.7 / 0.416	376.43 / 537.69	0.463 / -0.180
09/23 - 10/08	-	0.653 / 0.5	393.41 / 643.12	0.352/ 0.113
10/09 - 10/26	-	0.593 / 0.833	429.95 / 152.88	0.302 / 0.449

Table 1: Experiment result with S & P 500 index features

Prediction date	Word size	Classification (acc)	Regression (MSE)	Trading (return %)
07/31 - 08/17	680	0.875 / 0.666	55.419 / 215.57	0.475 / -0.018
08/18 - 09/03	1207	0.785 / 0.833	64.944 / 1188.86	0.521 / 0.896
09/04 - 09/22	1583	0.85 / 0.416	133.75 / 507.64	0.900 / 0.213
09/23 - 10/08	1852	0.865 / 0.5	128.37 / 695.36	0.937 / 0.113
10/09 - 10/26	2188	0.765 / 0.75	143.76 / 132.99	0.904 / 0.502

Table 2: Experiment result with S & P 500 index features and bag-of-words text features with stopwords

Prediction date	Word size	Classification (acc)	Regression (MSE)	Trading (return %)
07/31 - 08/17	520	0.875 / 0.666	47.378 / 226.24	0.438 / -0.018
08/18 - 09/03	1043	0.821 / 0.833	62.046 / 1166.6	0.556 / 0.896
09/04 - 09/22	1427	0.825 / 0.416	132.15 / 517.17	0.900 / 0.213
09/23 - 10/08	1700	0.865 / 0.5	128.08 / 694.24	0.937 / 0.113
10/09 - 10/26	2055	0.765 / 0.75	143.42 / 126.87	0.865 / 0.502

Table 3: Experiment result with S & P 500 index features and bag-of-words text features without stopwords

10/09 - 10/26	Classification (acc)	Regression (MSE)	Trading (return %)
Index feature	0.593 / 0.833	429.95 / 152.88	0.302 / 0.449
Text with stopwords	0.765 / 0.75	143.76 / 132.99	0.904 / 0.502
Text without stopwords	0.765 / 0.75	143.42 / 126.87	0.865 / 0.502

Table 4: Experiment result with different features on the last fold of validation

Model	stopwords	positive terms	negative terms
Logistic Regression	V	['ahead' 'equity' 'half' 'income' 'lse']	['cents' 'crude' 'customers' 'debt' 'dollar']
Logistic Regression		['ahead' 'equity' 'half' 'income' 'lse']	['cents' 'crude' 'customers' 'dollar' 'japanese']
Ridge Regression	V	['acquire' 'credit' 'deadline' 'half' 'index']	['against' 'care' 'court' 'dollar' 'ecb']
Ridge Regression		['acquire' 'credit' 'deadline' 'half' 'index']	['care' 'court' 'dollar' 'ecb' 'japanese']
Naive Bayes	V	['information' 'oil' 'per' 'technology' 'your']	['debt' 'oil' 'per' 'rate' 'your']
Naive Bayes		['information', 'lse', 'oil', 'rate', 'technology']	['china', 'debt', 'dollar', 'oil', 'rate']

Table 5: Terms importance analysis