Stock Market Prediction Using Long Short-term Memory

Chi Lin, Jiachao Fang

Introduction

The research of stock index prediction has been conducted for years. However, the scope of the research is large because of the uncertainty and volatility of market. Compared with other data, the stock indices are relatively easy to obtain, which allows us to reduce the burden of finding ground-truth data. In this study, we build a stock index prediction model with better performance and give investors a direction when they make investment decisions.

The prediction accuracy of stock index of existing researches are lower than 70 percentage (Lai 2014), which is relatively low comparing to other machine learning tasks (e.g. image recognition). We assumed there are two main reasons behind this issue: first, there are too many influential factors of stock index, such as politics and economics; second, it lacks of applicable machine learning models for stock index prediction because of the limits of research on neural networks in the past years.

The focus of this work is finding input features and using recurrent neural network. We look for the data that is most likely to affect E-mini Dow ($5) Futures; that is, the major events happened on each day and history of futures index. We train a LSTM model by using these related features. By doing so, our model is sensitive to the time and news that happened in the world. From our study, we find that news data happened more close to predicted index time, index history alone and more input data give better performance.

Related Works

Time series prediction Unlike image or text prediction, stock index is volatile as time change. For example, ImageNet (Deng et al. 2009) is an image dataset organized according to the WordNet hierarchy (Miller 1995). Once the relationship between synonym sets are established, it is rarely changed, so the research of image prediction could ignore the time factor. However, the stock markets are easily affected by multiple factors such as major events happening in the world, confidence of investors and trend of stock index and these factors are related to time. To address the issues, there are some forecasting studies based on time series. Babu proposed an ARIMA-ANN model (Zhang 2003) which combined the linear auto regressive integrated moving average (ARIMA) and nonlinear artificial neural network (ANN) models, because Babu supposed that the existing linear or nonlinear models couldn't make highly accurate predictions from time series data. Sapankevych and Sankar (Sapankevych and Sankar 2009) listed the time series prediction researches using SVM and SVR models. However, recently more and more research is using recurrent neural network (RNN) and one of its special type, LSTM, models to train time-series data. The advantage of RNN model is that it can preserve the previous computations and generate new output by combining current input and previous computations. As a result, in our study we use LSTM models as our prediction method.

Stock market price forecasting Over the years, different models are used to predict the trend of stock markets. Kim (Kim 2003), Cao (Cao and Tay 2003) and Kazem (Kazem et al. 2013) used SVM models to predict stock index. On the other hand, Hsieh (Hsieh, Hsiao, and Yeh 2011) employed RNN model to make prediction. Among these studies, their training data are only from history of stock index, so some studies such as Schumaker (Schumaker and Chen 2009) and Hagenau (Hagenau, Liebmann, and Neumann 2013) tried to use financial news to conduct their studies. As we can see, the mentioned studies used data based on either history stock index or financial news. We suppose these both factors are important to predict stock market price, so in our experiments we use both of them to train our model.

Text mining The Efficient Market Hypothesis (EMH) (Längkvist, Karlsson, and Loutfi 2014) shows that stock prices are largely driven by news rather than stock index history. Text mining techniques can be used to extract and select features of news. Khan et al. (Khan et al. 2010) summarized common steps of text feature extraction are "tokenization, removing stop words, and stemming word". There are six most frequently-used metrics of text feature selections: term frequency reverse document frequency (TF-IDF), information gain (IG), Gini methods, Chi-square, odds ratio, and Chi-square (Khan et al. 2010). Bi-Normal Separation (BNS) brought up by Forman (Forman 2003), which is also a popular metric for text feature selection, for it performed

much better than other metrics. Since stock index is volatile as time change, when doing text feature selection, we should also consider influences of the time, which add much difficulty to our study.

Methods

Feature collection and processing To obtain free and reliable ground-truth data, we use the stock index history dataset provided by Quandl (Qua) and Reddit news dataset issued by Pushshift.io (Pus) . We download history index of E-mini Dow ($5) Futures from 2008-2018. The indices are split day by day. For each day, it shows open, highest, low, last, settle indices in that day. It also lists volume and previous day open interest. We use the last index of each day as our predicted label and other fields as our input features. For daily news data, we collect news articles posted in the news category on Reddit. The collected news range from 2008 to present. For each news, it contains title, created date, number of comments and score. Here, the score means the number of upvotes minus the number of downvotes. The news we download from Pushshift.io (Pus) API is raw data so it has many duplicate and useless data such as same news title show up in multiple times and spam. To address these issue, we first remove the news whose the sum of number of comments and score is less than five. The reason why we do so, it is because for those less value news such as spam, Reddit hides them from the category so the users can't vote them or make comments for them. Hence, the number of comments and the score of the news won't be high. And the reason why we don't remove news based on either the number of comments or the score is because, first, some news even they have high scores but don't have many comments, secondly, some news are controversial so the users discuss a lot about them and give them lot of downvotes.

After removing some news, we use SequenceMatcher python class to merge duplicate or similar news titles. We compare the length of the titles and leave the longest one. Also, we leave the sum of number of comments and the sum of score of them. Next, we group the news day by day based on their created time. The created time provided by Pushshift.io (Pus) API is UNIX timestamp format and uses the UTC/GMT time zone, but the E-mini Dow futures uses the Chicago time zone. Also, we need to consider the daylight saving time. Here, we use pytz and datetime python module to separate the news.

As mentioned above, we collect news from 2008-2018 on Reddit. Considering the number of the news for each day increase over years, we choose 10 news with largest score to analyze in order to reduce bias. However, in early years (2008-2011), the number of the news in each day was less than 10, so we discard news data before 2011.

To extract information from the news and for convenience, we only use the news title as feature and process the titles with sentiment analysis.

The new title processing steps are as follows: first, we use nltk python module to tokenize the titles, remove stop & stem words. Then, we use sentiment module of nltk to get 4 sentiments score for each title, which are negative, neutral, positive and compound core respectively. After getting all sentiment scores of every title, we calculate mean values for 4 sentiment scores of 10 selected titles every day, then we have four features: mean negative score, mean neural score, mean positive score and mean compound score.

To summarize, we collect stock index history on Quandl API and news on Reddit API and construct the features through multiple python library. Our features consist of mainly four parts: stock index history, news score, number of comments of each news and news sentiment scores.

Model training We use LSTM model as our predictive methods, given its excellent performance in analyzing time series data. Since the selected features are of different scales and units, we first standardize and normalize all the features. Then we use basic LSTM recurrent network cell, with one hidden layer, to train the model. During the training process, we tune hidden layer units, learning rate, batch size, time step, and iteration numbers to compare different performances. After get the optimal hyperparameters of the model, we retrain a basic LSTM recurrent network with these parameters and get the prediction result. Our test result will be evaluated by mean absolute percentage error (MAPE) defined as eq.1 and root mean squared error (RMSE) defined as eq. 2.

$$\frac{100\%}{n}\Sigma_{t=1}^{n}|\frac{Y_t - F_t}{Y_t}| \tag{1}$$

$$\sqrt{\frac{1}{n}\Sigma_{t=1}^{n}(Y_t - F_t)^2} \tag{2}$$

Experiments

Design We design three control experiments to compare predicted results. All the experiments will use the same model, LSTM with one hidden layer, ten rnn units, and starting with learning rate of 0.0006.

First, we compare the predicted results by using news features extracted by the different time delimitation in first experiment. Since stock market closes at 4:00 pm every day, it is important to decide which time period of news to predict the stock market for next day. Therefore, we collect news and split them day by day using 1:00 pm and 2:00 pm respectively.

For second experiment, we compare the predictive results using different input features. We evaluate the effect of using stock index history alone and the effect of using both stock index history and news.

Last, we compare the predictive results using different lengths of history data. We evaluate the effect of using stock history and news of 300 days and 1000 days to build the model.

Results Our experiment results are showed as table 1.

First, as we can see, the news dataset split by 2:00 PM has higher accuracy. Moreover, RMSE can increase dramatically up to 105 points when using 2:00 PM as splitting point. For example, we can see RMSE decreases from 430.37 to 325.19 when the range of the training data is 1000

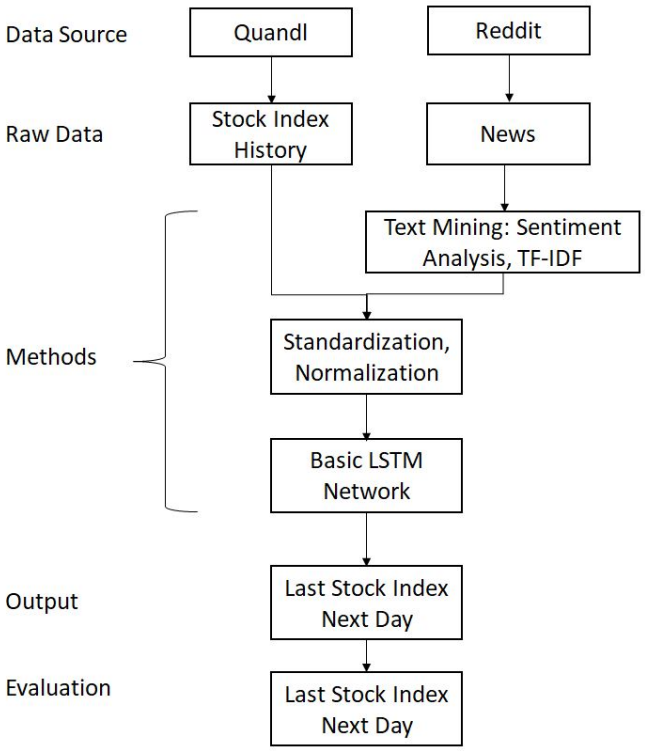

Figure 1: Flow Chart

-1300. The result shows that when closer the time of the news is to the closing time of the stock market, it impacts the last price of the stock market more.

For second experiment, we find that using index history alone actually has better predictive results comparing to using both index history and news sentiments. This phenomenon is prominent in the 1:00 PM dataset, which shows that news sentiments do not help in stock prediction. It contradicts to our previous assumption that stock index is driven by news.

Last, we notice that when more history data is used to train the LSTM model the predictive results are better, which corresponds with our assumptions at the beginning. It also proves the excellent performance of LSTM in predicting time series data.

Conclusions

In our study, we develop a systematic methodology to collect and process the index history and news posts in Reddit. We use Quandl to download E-mini Dow ($5) Futures and Pushshift.io to download Reddit news posts. To process raw news data, we merge some of them and group them by proper time zone. Later, we calculate the sentiment scores of news titles by date. After getting all the required features, we process them by feature engineering techniques, such as standardization and normalization. Using sentiment scores of news posts together with the index history as input features, we train a LSTM model and predict future indices.

For experiment results, our most interesting finding is that feeding predictive model with the history of the stock index and the news can not lead to better performance comparing to using stock history index alone. Also, we find that trained the predictive model with more data and the news which is close to predicted time can give better results.

Directions of Future Research

Because of the limit of time, there is still room for improvement on our research. First, we conclude that the influence of the news on the predicted index becomes larger when the news happened closer to the predicted index time. However, in our experiments, we test only 1:00 PM and 2:00 PM datasets, which probably can not represent all. Therefore, more time demarcations need to be tested to prove the conclusion. On the other hand, we find that the features selected from the news title can not improve the predicted results. Nevertheless, this can be caused by the features we extract from the news title, which may not represent the news title itself. Therefore, we need to try to use different feature extracting methods to find the features which can truly represent the news.

References

Cao, L.-J., and Tay, F. E. H. 2003. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks* 14(6):1506–1518.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical im-

Table 1: Experiment Results

	Training Data Range	1000-1300	300-1300	1000-1300	300-1300
	Begin at	1:00 PM		2:00 PM	
	Testing Data Range	1300-1353			
History	MAPE	1.17%	0.99%	1.32%	1%
	RMSE	345.59	297.51	390.33	294.19
History + Sentiment	MAPE	1.47%	1.24%	1.11%	0.98%
	RMSE	430.37	364.85	325.19	296.68

age database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research* 3(Mar):1289–1305.

Hagenau, M.; Liebmann, M.; and Neumann, D. 2013. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems* 55(3):685–697.

Hsieh, T.-J.; Hsiao, H.-F.; and Yeh, W.-C. 2011. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied soft computing* 11(2):2510–2525.

Kazem, A.; Sharifi, E.; Hussain, F. K.; Saberi, M.; and Hussain, O. K. 2013. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied soft computing* 13(2):947–958.

Khan, A.; Baharudin, B.; Lee, L. H.; and Khan, K. 2010. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology* 1(1):4–20.

Kim, K.-j. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55(1-2):307–319.

Lai, P.-f. B.-W. C. H. 2014. Performance of stock market prediction. *Public Finance Quarterly* 4:471.

Längkvist, M.; Karlsson, L.; and Loutfi, A. 2014. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters* 42:11–24.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

The pushshift.io reddit api. https://github.com/pushshift/api. Accessed: 2018-04-01.

Financial, economic and alternative data quandl. https://www.quandl.com. Accessed: 2018-04-01.

Sapankevych, N. I., and Sankar, R. 2009. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine* 4(2).

Schumaker, R. P., and Chen, H. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)* 27(2):12.

Zhang, G. P. 2003. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing* 50:159–175.