Predicting Stock price trend from news articles

Jungbeom Lee
UCLA
Jungbeol@ucla.edu

ABSTRACT

In this paper, I propose “machine leaning technique” to predict the direction of movement of stock price trend for the United States stock market. The study uses neural network (NN), support vector machine (SVM), and decision tree (Tree) for predicting daily direction of stock market trend. The stock market data is based on the historical data from the index of NASDAQ composition (IXIC) and the index of S&P CBOE (S&P 500) is from January 3, 2000 to July 11, 2018. The predicting models will be made of news articles which are collected from fox.com, nytimes.com, cnn.com, and huffingtonpost.com. Most people receive financial information from news articles. I will evaluate news articles as an input data for predicting stock market price trend with machine learning techniques.

Keywords

Machine learning, news article, stock price trend prediction, support vector machine, neural network, decision tree

1. INTRODUCTION

The efforts to predict stock prices are gigantic in both academic and business fields. It has been interested since the stock market established because it is considered as predicting future with models which are built by many different skills and knowledges. Since machine learning is become successful in predicting field, there are a lot of interesting works to forecast stock price trend with machine learning technique.

People try to predict the value of company with various tools. but it is very difficult to predict how stock prices will change. People can invest with information from business news. For people who is not in the field, it is rare to get secret sources. In this paper, I collect news articles from various news websites by using python web scraper and news parser API, train them with neural network, support vector machine and decision tree classifiers and predict stock price trend with the trained models. The goal for this project is predicting whether the stock market trend goes up or down from the previous day using news articles.

There are many other previous efforts to predict stock price with machine learning method. Kyoung-jae Kim, Ingoo Han proposes hybrid model which combines Genetic algorithms and neural network to predict stock market trend. They use technical indexes and the daily Korea stock price index (KOSPI) as data. The accuracy of the model from the research has range from 58.50% to 65.79% [1]. Another team of Rohit Choudhry, and Kumkum Garg evaluate same hybrid model to compare with neural network model. They use technical indexes and the Indian stock market data for the study. The research shows hybrid model can improve about 4% of the accuracy. In their paper, the average of accuracy of neural network is 56.82% and the average of accuracy of the hybrid model is 60.51% [2]. The other team Jigar Patel, Sahil Shah, Priyank Thakkar, K Kotecha uses trend deterministic data and CNX nifty and S&P BSE. They compare four machine learning models; artificial neural network (86.69%), support vector machine (89.33%), random forest (89.98%), naïve-Bayes (90.19%) [3]. There are few more similar researches working with news articles. Schumaker and Chen use Arizona Financial Text System (AZFinText) which uses a synthesis of linguistic, financial and statistical techniques on financial new articles and it can predict with 71.18% accuracy [4] and it is higher than support vector machine (57.10%) on same datasets [5]. In addition, Hagenau, Liebmann, and Neumann enhance existing text mining methods by using more expressive features to represent text and by employing market feedback as part of our feature selection process. It can predict high as 71.8% [6].

2. Problem Definition and Formalization

Most people who do not involve in financial invest field use daily news articles to predict stock price. People widely believe that news articles affect significantly to stock market trend. In this project, news articles are used as input data to train predicting models with machine learning techniques. The accuracy of models will represent the relationship between news articles and stock market price trends. In addition, it will tell that using newspaper as indicator for stock invest is good idea or not.

To solve this problem, there are two data needed to be prepared: News articles and stock market price trends. Each news article will be sorted by published dates and vectorized for feeding machine learning classifiers. Stock market price trend data is needed to be matched with dates from news articles. The delta between a closing price of each dates and a closing price of previous dates will determine a label for the news article data points. The data will be divided 75/25 to find accuracy of the models. The accuracies will show the result of the study.

3. Data preparation and preprocessing

The study needs to obtain news articles to train machine learning models. I use newspaper API which is convenient python helper library to collect news articles. I produce the sample data by using article API configured to scrap online news articles to get written date and the news articles. The newspaper API scraps the news from the URL address which shows a news article or is newspaper website. I use four different sources such as Fox, NYTimes, CNN, and Huffington Post. I can collect 488 distinct dates since May 23, 2011 to July 07, 2018. There are following reasons that the study uses limited data points. Not every date has published news articles. Many of the articles do not have published dates; in addition, the newspaper API be able to download only what be accessible from newspaper websites, but most newspaper websites do not store old news articles. The collected news articles from different sources are sorted by published dates. Then, all articles are collided to one long text by published dates and some news articles that do not have published dates use an invalid date to be dropped.

The study needs to get stock market trend to label the data points. There are many indices for stock market price trend, but I will use few of major stock indices which summarize the performance of major groupings of stocks. For example, IXIC is an index for NASDAQ composite, and SPX is an index for S&P 500. By looking at daily news articles, I hope to predict how the stock price will change: either increase or decrease. Specifically, I will attempt to predict daily changes for the stock market closing price (4 P.M. EST). To obtain stock market price trend data, I use the Alphavantage (https://www.alphavantage.co/) web application to get stock data for IXIC, and SPX. This API let me obtain stock price data for daily information for 18 years. It is sufficient information to process required data for the project.

In association to the news articles, with stock data and transformation, as an input data for machine learning API, I vectorize the long texts which are collected news articles with using Doc2Vec API. Then, I find corresponding stock price delta; setting 1 for increasing delta, and 0 for neutral and decreasing delta.

4. Method description

In this section, I will describe the python APIs that used for the study. For example, they are newspaper API used for getting news articles, The Alphavantage web application which provides stock market data. For machine learning predictors, I use neural networks, support vector machine, and decision tree from scikit-learning libraries.

Data preparation and processing

Newspaper3k

It is inspired by the requests for its simplicity and powered by lxml (python xml file parsing library) for its speed [7]. Newspaper3k is a python library that can scrap and curate articles. Newspaper library take URL as an input. Then, it downloads a HTML file to parse later. The beauty of the library is that it can scrap the article with calling few functions and parse the HTML file. In addition, it also has natural language parsing to extract keywords and summary of articles.

To obtain news articles from websites, I input news website URL. It will download each news article webpage and parse to extract article text part and published date. Then, all extracted data is stored in JSON file. [8]

The library supports multi-thread extracting functions. However, the hardware has limited memory size, so articles are separately extracted and stored by each news website.

Doc2Vec

Doc2Vec [8], proposed initially by Mikolov et al. and supported as a component of Gensim library which is general library for topic modeling for humans, is an extension of his earlier Word2Vec methods that learns word embeddings for text association tasks, extended to entire paragraphs and sentences by associating an additional unique document tag as a label.

The API requires (long one-line text, tag) tuples to train Doc2Vec model. Long one-line text is a collided news article and tag is a published date for the articles. To train Doc2Vec model, I feed the processed tuples to Doc2Vec Neural Network using the CBOW (Continuous Bag Of Words) method with 100 feature dimensions, 300 window, and counts words including unique words.

Doc2Vec transforms documents to vectors after training models. We can extract a vector for associated document from trained Doc2Vec model.

Prediction Classes

I decide to choose classification instead of regression. It is because classification method will be easy to see the error rate such as correctly predicted counts versus wrong prediction counts. This study is hence framed as a binary classification where I classify each document as either 1 - if the stock price moved up, or 0 - if the stock price fell or stay. To get deltas, I use closing market prices because it is the last value of daily period. I use target dates and days before previous date of a target date to increase the accuracy of the predictors. There are some stock market closed dates for holidays and weekends. I use two recursive methods that one can find next market open date and the other can find previous market open date. Prior one is used for finding opening target dates and later one is used for finding opening days before previous date of a target date.

Machine learning predictors

SVM

SVM is an abbreviation of Support Vector Machine. It is linear model in machine learning. It separates data points with hyperplanes to classify each data points. The main idea of SVM is maximizing margin for separating data points to optimize the result of learning. The general linear model is:

$$f(x) = w^T x_i + b \; x \; is \; input \; data,$$

$$where \; w \; is \; weight \; vector, b \; is \; bias \; term.$$

To maximize margin, we need to solve quadratic problem. After making some progression,

$$w \; = \sum \quad \alpha_i y_i x_i \;\; , b = y_k - w^T x_k,$$

$$for \; any \; x_k \; such \; that \; a_k \neq 0, w \; is \; weight.$$

Applying these to general linear model,

$$f(x) = \sum \quad \alpha_i y_i x_i^T x_i + y_k - w^T x_k$$

$$where \; f(x) \; is \; our \; hyper \; plane \; model.$$

To use SVM for the study, I use SVC (support vector classification) class. For this predictor, I use RBF (radical basis function) kernel and choose 7 for the degree to get the results. In addition, support vector machine algorithms are not scale invariants. It is highly recommended to scale the data according to scikit-learning organization which provides the libraries [9].

Neural Network

Neural network models can be representative as MLP (multi-layer perceptron). MLP is a supervised learning algorithm that learns a function by training on a dataset:

$$f(\cdot): R^m \rightarrow R^o$$

where m is the number of dimensions for input and o is the number of dimensions for output.

Given a set of features (X = x_1, x_2, …, x_m) and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. [9]

With the popularity of Artificial Neural Network (ANN) algorithms for performing classification tasks, we also decided to explore its use for this task. Our architecture is that of a fully-connected multilayer perceptron, with an input layer of 100 dimensions (the document vector), and six hidden layers, and a one-dimension output layer that produces the predicted classification. The number of hidden layers and the number of neurons along with the training rate in the respective layers were empirically derived from initial experimentation, and we went with the tanh function as the activation function for the neurons in the network. For practical usage, it is also highly recommended to scale the data.

Decision tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. [9]

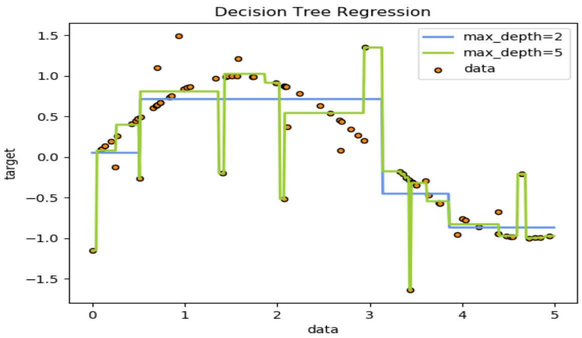

Figure 1. decision tree example [9]

For example, decision trees learn from data points to find an approximation curve with a set of if-else decision rules. Decision trees have some advantages and dis-advantages. It is simple to understand and easy to implement. It requires little data preparation such as data normalization or dummy values inserting/removing. However, decision trees always experience hardship between underfitting and overfitting. Deeper trees will cause overfitting. For this study, I use max depth as seven to avoid underfitting and overfitting.

5. Experiments design and Evaluation

Experimental Design

The study is made of five steps to accomplish the goal. They are data collecting, data processing, training the machine learning models, predicting with the models, and evaluating the results. Most time-consuming part is data collecting. It depends on the number of news which are collected. The total time to process whole steps will be few minutes except data collecting part. It takes between 7 and 20 minutes to collect the news articles that are used for the studies.

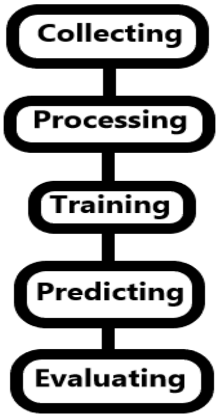

Figure 2. The five steps of the experiences

The newspaper3K library is used for data collecting. Because of the hardware memory issue, I separately collect news articles by sources which are multiple news websites. Each collecting program load the news article URLs with provided news website URL. Then, it iterates each article to download and parse. Every article that is scrapped by the methods will be stored as a JSON file. In the stored JSON files, there will be each news article and published dates.

The most challenging part is data preprocessing and processing. After this procedure, news articles and daily stock price trend will be vectorized to feed machine learning classifiers as training data. The collected news articles which do not have published dates will be set as minimum date that does not have meaning to be removed later. Then, news articles are sorted and collide by published dates. The preprocessed documents become vectors by Doc2Vec library as tagged by published dates. These vectors are needed to pair with daily stock market trend. After the pairing, the pairs are divided into 75/25 % as training data and test data.

For this experiment, I use three machine learning classifiers; support vector machine, neural network, and decision tree. After processing the data, it is simple as calling a method to training the classifiers.

For evaluation, it is needed to find optimization points for all classifiers. Then, I compare accuracy between one day delta for daily stock price trend and two-day delta for daily stock price trend. Lastly, all classifiers are compared for the results.

Evaluation

To evaluate the trained machine learning models, I use accuracy. The accuracy is the ratio of corrected predicted points over all predicted points.

For support vector machine classier, there are two parameters that are evaluated in this study; degree and kernel function. I use radial basis function as kernel function to test multiple degrees. Then, I test which kernel function is more suitable for the prepared data set.

In table 1, the kernel function for support vector machine is radial basis function. The table shows degree 7 gives best result as 62.53%. From degree 3, the accuracy is getting higher until degree 7. Then, it meets suboptimal point degree 11 and it goes down again. I also tested degree 6 and it provides identical accuracy.

Degrees					
Deg 3	5	7	9	11	13
60.82%	58.76%	60.82%	57.73%	60.82%	57.73%
57.73%	60.82%	63.91%	63.91%	60.82%	58.76%
62.88%	59.79%	62.88%	59.79%	60.82%	57.73%
60.47%	59.79%	62.53%	60.47%	60.82%	58.07%

Table 1. degrees for SVM (RBF)

In addition, I also try with linear kernel function to check if degree 7 is optimal point. In table 2, we can see degree 7 is still best point.

Degrees			
Deg 3	5	7	9
59.79%	59.79%	61.85%	56.70%
58.76%	62.88%	59.79%	63.91%
61.85%	62.88%	63.91%	59.79%
60.13%	61.85%	61.85%	60.13%

Table 2. degrees for SVM (Linear)

In table 3, I compare two kernel functions; linear and radial basis function. Radial basis function gives better result (62.53%) than linear function (61.85%).

Kernel functions	
LINEAR	**RBF**
61.85 %	60.82 %
59.79 %	63.91 %
63.91 %	62.88 %
61.85 %	62.53 %

Table 3. kernel functions for SVM

For support vector machine classifier, the parameters for kernel function and degree are radial basis function and seven to get best result.

Neural network classier can take many parameters and I manually set hidden layer size, activation function, and solver. Hidden layer size means the number of neurons in hidden layers. Activation function means activation function for hidden layer. Solver means the solver for weight optimization. There are three available solver options such as ‘lbfgs’, ‘sgd’, and ‘adam’. ‘lbfgs’ is an optimizer in the family of quasi-Newton methods. ‘sgd’ refers to stochastic gradient descent. ‘adam’ refers to a stochastic gradient-based optimizer [9]. Since ‘adam’ is good for relatively large datasets and ‘lbfgs’ performs for small datasets, I choose ‘lbgfs’ for the classifier. First, I will compare the accuracies among different activation functions. Then, I will find best hidden layer size.

In table 4. The hidden layer size is 2 which is minimal value. There are three activation functions are tested; ‘logistic’, ‘tanh’, and ‘relu’. ‘logistic’ is the logistic sigmoid function, returns f(x) = 1 / (1 + exp(-x)). ‘tanh’ is the hyperbolic tan function, returns f(x) = tanh(x). ‘relu’ is the rectified linear unit function, returns f(x) = max(0, x) [9].

Activation functions		
logistic	tanh	relu
58.76%	59.79%	52.57%
42.26%	50.51%	52.57%
52.57%	58.57%	48.45%
51.19%	56.35%	51.19%

Table 4. activation functions for NN

The best activation function is ‘tanh’ function for the neural network classifier. ‘tanh’ provides significantly higher accuracy than other two’s accuracies which are 51.19%. In table 5, the activation function is set as ‘tanh’ and search best hidden layer size in the range between 2 and 8.

Hidden layer size			
Degree 2	4	6	8
59.79%	50.51%	57.73%	51.54%
50.51%	56.70%	55.67%	50.51%
58.57%	47.42%	56.70%	43.29%
56.35%	51.54%	56.70%	48.44%

Table 5. hidden layer size for NN

In table 5. Hidden layer size is 6 to perform best result in the range. I also try hidden layer size 5 and 7. They give about 2% lower accuracies. 56.70% (hidden layer 6) is best.

Last machine learning classifier for the study is decision tree. For decision tree, I modify ‘max_depth’ parameter for regulation that can avoid overfitting. ‘max_depth’ means the maximum depth of the tree.

Maximum depth of the tree					
3	**5**	**7**	**9**	**11**	**Max**
49.48%	49.48%	47.42%	56.70%	44.32%	50.51%
51.54%	48.45%	43.29%	52.57%	48.45%	46.39%
55.67%	47.42%	50.51%	50.51%	54.63%	54.63%
52.23%	48.45%	47.07%	53.26%	49.13%	50.51%

Table 6. maximum depth of DT

After trying several times, I can find out maximum depth of the decision tree classifier is 9. However, it is 53.26% which is not distinct among all other results.

Above tests use 2-day delta. To get the stock price trend, I subtract the stock price of two previous date of the target date from the stock price of the target date. The purpose of using 2-day delta is increasing the accuracies. There are two assumptions. stock market brokers can access the information immediately and it takes about one day to publish as a news article. Table 7 will show this assumption holds or not.

1-day delta and 2-day delta

SVM	NN	DT	SVM	NN	DT
63.91%	50.51%	51.54%	60.82%	57.73%	56.70%
63.91%	51.54%	59.79%	63.91%	55.67%	52.57%
65.97%	44.32%	53.60%	62.88%	56.70%	50.51%
64.59%	48.79%	54.97%	62.53%	56.70%	53.26%

Table 7. comparison for different deltas for stock trend

In table 7, left half uses 1-day delta and right half uses 2-day delta. For support vector machine and decision tree, 1-day delta provides higher accuracy. For Neural network, 2-day delta works well. The accuracy is increased 7.91% from 48.79% to 56.70%. However, the assumptions are not true because it improves only neural network classifier and it worsens other classifiers.

Above tests are based on the index of NASDAQ composition (IXIC). To validate the tests and trained machine learning models, I will use the index of S&P CBOE (S&P 500) for stock price trend instead of the index of NASDAQ composition (IXIC). The parameters of models are set for the best result from previous tests. For support vector machine classifier, it uses 1-day delta, the kernel function is radial basis function, and degree is 7. For neural network classifier, it uses 2-day delta, activation function is tanh function, and hidden layer size is 6. For the decision tree classifier, it uses 1-day delta, and the maximum depth of the decision tree is 9.

Comparison between IXIC and S&P 500

SVM	NN	DT	SVM	NN	DT
63.91%	57.73%	51.54%	53.60%	54.63%	48.45%
63.91%	55.67%	59.79%	59.79%	52.57%	55.67%
65.97%	56.70%	53.60%	54.63%	53.60%	42.26%
64.59%	56.70%	54.97%	56.00%	53.60%	48.79%

Table 8. Different stock trend sources

Table 8 shows that the results of different stock price trend data are not remarkable as comparing with previous results. The predictors with news articles perform worse for S&P 500 than IXIC.

6. Conclusion

The goal of the study is that people can use news articles to predict stock market price trend. In conclusion, it is not safe to use news article as the source to predict stock market price trend. The accuracies of the study are 64.59% from support vector machine model, 56.70% from neural network model, and 54.97% from decision tree model. Since a single wrong prediction will cause huge damage on financial status, so the accuracy of 64.59% is not credible. The accuracy of other researches that uses news articles with advanced models is also highest as only 71.80% [6]. Although these models are claimed as profitable on simulation, it will be hard to be applied to normal people who are not in financial field. Similarly, other sources such as technical indexes are led to the range of accuracies from 58.50% to 65.79%. On the other hand, trend deterministic data give remarkable results which are over 86%.

7. REFERENCES

[1] Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications,19*(2), 125-132. doi:10.1016/s0957-4174(00)00027-0

[2] Choudhry, R., & Garg, K. (2008). A Hybrid Machine Learning System for Stock Market Forecasting. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:2, No:3, 2008*

[3] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications,42*(1), 259-268. doi:10.1016/j.eswa.2014.07.040

[4] Schumaker, R. P., & Chen, H. (2009). A quantitative stock prediction system based on financial news. *Information Processing & Management,45*(5), 571-583. doi:10.1016/j.ipm.2009.05.001

[5] Schumaker, R. P. and Chen, H. 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Trans. Inform. Syst. 27, 2, Article 12 (February 2009), 19 pages. DOI = 10.1145/1462198.1462204 http://doi.acm.org/10.1145/1462198.1462204

[7] Newspaper3k: Article scraping & curation¶. (n.d.). Retrieved from http://newspaper.readthedocs.io/en/latest/

[8] The Github repository for the research. (n.d.). Retrieved from https://github.com/nanaya07/Stock-trend-prediction-news-machine-learning/

[9] Scikit-learning. (n.d.). Retrieved from http://scikit-learn.org

▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪