Prediction of Stock Price analyzing the online financial news using Naive Bayes classifier and local economic trends

A.S.M Shihavuddin
MSc Student, VIBOT 3rd generation,
Lecturer (On leave), EEE Department,
IUT
Dhaka, Bangladesh
shihav407@yahoo.com

Mir Nahidul Ambia
Lecturer, EEE Department, Eastern
University
Dhaka, Bangladesh
n_shaon@yahoo.com

Mir Mohammad Nazmul Arefin
Lecturer, EEE Department, Stamford
University
Dhaka, Bangladesh
mmn.arefin@gmail.com

MD. Mokarrom Hossain
Lecturer, EEE Department, Stamford University
Dhaka, Bangladesh
m_hossain1718@stamforduniversity.edu.bd

Adnan Anwar
Lecturer, EEE Department, University of Asia Pacific
Dhaka, Bangladesh
adnan@uap-bd.edu

Abstract- **Market and stock exchange news are special messages containing mainly economical and political information. This paper represents data mining algorithms which has been tested on this available information to learn useful trends about the behaviour of the stock markets. The learned trend holds the key to interpret the present and predict the next stock price. This resented work uses Naïve Bayes Algorithm to classify text news related to FTSE100 given on these mentioned websites and the classifier is trained to learn the movement in the stock price (up or down) from the news articles in the web pages of that day. Several heuristics are being used to remove irrelevant parts of the text to get a reasonable performance. This model had demonstrated a statistically significant performance in predicting stock prices compared to other previously developed methods.**

Keywords: Naive bayes classifier, noise in stock market, A priori selection, inductive bias.

I. INTRODUCTION

The approach to classification of markets news may be similar as the approach for determining document relevance. Experts construct a set of keywords which they think are important for moving markets. The occurrences of such a fixed set of several hundreds of keyword records will be counted in every message. The counts are then transformed into weights. Finally, the weights are the input into a prediction engine (e.g. a neural net, a rule based system, or a classifier), which forecasts in which class the analyzed message should be assigned to. As it turns out, the computation of weights using a combination of traditional information retrieval techniques does not necessarily yield accurate forecasts [01], [02]. The weighting scheme consisting of three components (term frequency, document discrimination, and normalization) can algebraically be simplified and is therefore called simple weighting. Leung [01], Peramunetilleke [02] and W¨uthrich [03] show that simple weighting outperforms other information retrieval weighting schemes on various financial classification problems. In papers from Nahm, Mooney [04], [05], [06], [07] a mall number of documents will be manually annotated (we can say indexed) and the obtained index, i.e. a set of keywords, will be induced to a large body of text to construct a large structured database for data mining. Authors are working with documents containing job posting templates. A similar procedure can be found in papers from Macskassy [08], [09]. Key to its approach is the user's specification to label historical documents. These data then form a training corpus, to which inductive algorithms will be applied to build a text classifier. In Lavrenko [10], [11] we can find a method similar to our method. To each trend there exist a set of news that are correlated with this trend. The goal is to learn a language model correlated with the trend and use it later for prediction

In Macskassy [08], [09] the author describes a system, which is not classifying news in good-bad but in important-unimportant. It can provide users with well filtered news. Each user should give a direct statement of his or her interests, which helps to construct a model of importance. A similar approach that uses domain experts to specify a set of keywords has been used also in Peramunetilleke, Wong [12]. The first problem is that the importance of a news story cannot be often evaluated at the time the news story appears but sometime later depending on events (e.g. following messages, movement of markets) that will happen. The next problem is that experts have different opinion about interpretation of news. If experts would have the same opinion it would not be possible to make business with stocks. Most volume of stocks are sold and bought by investment funds managed by experts but to realize such a business it is necessary that one group of experts think it is the best time to sell and other group of experts think it is the best time to buy. The key problem and weak point is the necessity to have the user's specification

978-1-4244-6542-2/$26.00© 2010 IEEE

to label historical documents for training and classifying. The next weak point is that the importance will be measured by the impact of the news story on the price of the investors are reacting with a different delay. In Lavrenko [10], [11] the language model is not given by experts but is derived for each trend similarly as in our approach. The difference is that Lavrenko investigates immediate influence of each news story on price of single stocks during day-trading, has to compute short-time trends and assign to them the corresponding time-stamped news stories. The goal is that having a language model for every trend type, it is possible to monitor a stream of incoming news stories and estimate the corresponding trend to each news story. Then it is possible, for example, to deliver only such news stories to a customer that very probably corresponds to an upward trend. The weak point of this approach is that it is not clear how quickly the market responds to news releases. Lavrenko discuss it but the problem is that is is not possible to isolate market responses for each news story.

Internet provides almost all possible information on the stock market worldwide through various useful websites as Google finance, CNN Money, Bloomberg, Financial Times, Thisismoney, Yahoo finance and many more. The reliability of the information depends on the reputation and the quality of the source sites. The news in the web pages is also related with the time in the publisher country. The web pages published by UK will give report on the local operational time. Despite of all this variance, there is vital information related to the trends in stock price is hidden in these news articles. Using machine learning algorithms this valuable information can be retrieved and used for the financial analysis of the performance of the selected companies.

The work discussed in this report is based on finding the correlation between word attributes and the movement (Up or down) in the stock market. There are large numbers of attributes that can be used to classify texts in the news articles. This attributes are mainly the words that can represents positive, negative or neutral meanings to indicate the possibility of the direction of the stock movement. Naive Bayes classifier is used to handle this large number of attributes using the probability of the each attributes. The system developed learns from the previously given news articles (described as training sets) and creates a vocabulary list of attributes each having its own probabilities for positive or negative examples. This project Collected from April, 2008 to November, 2008 having one closing value report for each working day. In this way, total 170 articles were collected and the learner was trained on 140 data sets. The rest of the data were used for the testing purpose. Data is collected manually.

* As mentioned in the assignment, I took the articles of 31 January, 2008. As the last 5 digit of my student ID number is 79781. The summation of the first four numbers is 7 + 9 corresponding stock during the next one hour. The problem is that it is difficult to decide how much influence each of some hundreds of news stories may have had because learned from all the articles from a single financial website i.e. Thisismoney and the index selected for the learning is FTSE100

II. Bayes Classifier

After analyzing the entire learning algorithm the Naïve Bayes classifier is being selected for the following reasons

- ➢ There is no need to search though all the version space for the optimal hypothesis in this algorithm
- ➢ Prior knowledge can be effectively used in this method. Often in share market some trend can be found to be prominent and good to use as a valid a priori probability which can help the learner to show better performance
- ➢ It uses the probabilistic methods. The probability of all the attribute can create the ground to select the optimal hypothesis based on the training sets
- ➢ Simple to implement and efficiently useable for large number of attributes.
- ➢ Further training can be done for new training instances by updating the probability, received from the previous training

The Bayes Naive classifier selects the most likely classification V_{nb} given the attribute values a_1, a_2, ... , a_n. These results in,

$$V_{nb} = \underset{V_j}{\arg\max}\, P(V_j)\,\Pi\, P(\frac{a_i}{v_j}) \qquad (1)$$

We generally estimate P(ai/vj) using m-estimates:

$$P\left(\frac{a_i}{v_j}\right) = \frac{nc+mp}{c+m} \qquad (2)$$

Where, n = Number of training examples where v = v_j
nc = number of examples where v = v_j and a = a_i
p = a priori estimate for P (a_i / v_j)
m = the equivalent sample size

III. System Learning

Data collection: All the data used in this project are collected from the web page www.thisismoney.com mainly the review report on stock prices of FTSE100. The data is + 7 + 8 = 31, the last digit is 1. That's why I took 31th January to represent my distinctiveness in this report.

Data cleaning: The performance of the system depends on how effectively and efficiently data cleaning have been done. For this project, Data is cleaned in four consecutive steps. First all the numbers from the text have been removed. In the second step, all the words are first kept and anything like colon, comma, bracket, etc are removed. In the 3rd step, all the single character words are cancelled out from the list. In the final step, a most frequent neutral word list is compared with remaining words and they are cancelled out if they match. The stop list is comprised of words which is mainly neutral but frequent (like if, to, the, is, etc) to reduce the number of attributes. After doing all these clearing, one heuristics is used to give more importance to relevant part of the articles. In this case, the system looks for the index title word, as in this project, 'FTSE', 'Footsie'. These paragraphs have greater chance of having texts related to the selected index. Extra waits are given on the attributes found in these paragraphs. In this project, these paragraphs are repeated again in the text so that their frequency is counted twice means they are given double wait than other attributes.. This whole data cleaning system is developed using Matlab.

Inductive bias: the develop method while learning uses the following assumptions:

1. The probabilities of one text position are independent of the words that occur in other positions, given the documents classification V_j.
2. The probability of encountering a specific word W_k is independent of the specific word position being considered. The attributes are independent and identically distributed.
3. The writers tends to mention the name of the company or index when they writing about it.
4. The probability of the stock price to go down is higher when the company is doing badly for the last few months and its stock price is going down more frequent than average. The probability of the stock price to go up is higher when the company is doing better for the last few months and its stock price is going up more frequent than average.

The effect of data cleaning on Inductive bias: The data cleaning done in this project, is affecting the inductive bias of the system in a negligible amount. First, it removes numbers, and most frequent neutral words. Then it gives more wait on the attributes found in the paragraph which have name of the subject company in it. Thus the data cleaning process is not removing any irrelevant section of the test but trying to give more importance on the relevant documents. This method seems is adding new bias to the system, and that is; writers use the name of the company when they write about it. This bias can be accepted as it is very obvious that, while writing about a company in the news, the writer will use the name of the company to elaborate on the point.

Naïve Bayes Learning: The cleaned training sets are a set of text documents with their target values. V is the set of all possible values. This function learns the probability terms $P(w_k/v_j)$, describing the probability that a randomly drawn word from a document in class V_j will be the English word w_k. in this case the V_j have two possible values as up or down. The steps involved are;

1. Collects all word that occurs in the examples.
   Vocabulary: The set of all distinct words occurring in any text documents from the training sets.
2. Calculates the required $P(w_k/v_j)$ probability terms for each target value V_j (up or down) in V
   - Docs_j = the subset from training sets for which target value is V_j
   - Text_j = a single document created by concentrating all members o f doc_j
   - n = total number of distinct positions in Text_j
   - For each word w_k in Vocabulary, n_k = number of times work w_k occurs in Text_j
   - For each word w_k in Vocabulary,

$$P\left(\frac{w_k}{v_j}\right)=\frac{n_k+1}{n+|Vocabulary|} \qquad (3)$$

To get better performance, the words, taken from the days in which the share price has more movement than normal, are given more weights in the counting. The stock price movement of any day and average movement of that day are compared. This ratio gives the individual day's importance level. The span of this possible relative values is narrowed down by a Moderator value m and importance factor is calculated using the following equation. This Importance factor is then multiplied with the counting of all attributes found in the text of that individual day (All these values are calculated earlier and saved in the database to make the testing process faster).

$$IF=\frac{RSM+MV}{1+MV} \qquad (8)$$

Where, IF = Importance factor,
RSM = Relative stock movement
MV = Moderator value

Sample of the created vocabulary matrix is given in the table bellow:

Table 1: Example of vocabulary matrix*.

A priori Selection: In this project the A priory probability is selected using the overall year's performance and also the last week's trends as shown bellow,

$$PP(V_j)=\frac{1*POY(V_j)+.05*POW(V_j)}{1.5} \qquad (4)$$

Where, $PP(V_j)$ = A priory probability of V_j
$POY(V_j)$ = Percentage of occurring V_j in last 1 year
$POW(V_j)$ = Percentage of occurring V_j in last 1 week

Testing classifier: The learned vocabulary is tested on the news articles. The testing returns the estimated target value for the given document. A_i denotes the word found in the ith position within Doc.

- Positions = all word position in Doc that contains word found in Vocabulary

- Returns V_{NB} where

$$V_{nb} = \underset{V_j}{\arg\max} P(V_j)\Pi P(\frac{a_i}{v_j}) \quad (5)$$

For the simplicity of computation and to avoid the computation of very small probabilities logarithm is being used to find the final decision.

As, $\log(A.B) = \log A + \log B$ (6)

Also, $\log(A.B) > \log(C.D) if A.B > C.D$ (7)

The number of attributes can be reducing by removing the almost equal probable attributes for both up and down examples. They can be called as the neutral words and have negligible impact on the decisions. It reduces the computational complexity without reducing the system performance.

The price of the next day has little but important relevance with what happened today. This relevance can be checked using the same Naïve Bayes classifier developed here with some effective modification. The first modification would be to use the stock price of the next day as the target values in the same training sets. For example, if we are taking a text of Monday, the target value would be the stock price movement in the Tuesday. The prior probability can be selected as the same way as before. This prediction for the future value can be considered a good one if it have anything more than 50% accuracy constantly. The selection of prior probability done earlier can have good effect on the performance, when predicting stock movement; as it takes into account both the yearly and the weekly trends. The selection of the training sets is also crucial for this type of assignment. The articles written by experts analyzing the company performance and commenting on the possible future situation can be more applicable for satisfactory learning of the crucial attributes. The system was checked on the existing data sets and the results are discussed in the next sections. As the second modification, the selection of the training sets should be filter to get more relevant test elaborating about the trends or the future; and written as well as by experts. The naïve bayes classifier is reasonably effective for finding out the hidden characteristics or in other words the possible best hypothesis without searching the version space. As used in text classification the naïve learner can try to learn any possible information or any possible correlation with the today's news and tomorrow's stock price.

IV. Result and Analysis

The classifier is trained by data sets collected from April, 2008 to mid October, 2008 of total 140 and tested on the rest of the data sets of October and November making a set of 35. The number of training examples is varied and the performance is measured at different training sets. The found results are given below:

Table 2: The testing result of the system

No. of Test	Test 1	Test 2	Test 3
Training in total	91	119	152
Vocabulary length	4711	5380	5961
Training down	46	59	76
Training up	45	60	76
Words in down	22622	28647	36102
Words in up	21382	27315	33488
Accuracy on test	58.62%	66.62%	79.4%
Accuracy training	100%	100%	100%

Table 3: The testing result of the system (Adding the mentioned heuristics)

No. of Test	Test 1	Test 2	Test 3
Training in total	91	119	152
Vocabulary length	4711	5380	5961
Training down	46	59	76
Training up	45	60	76
Words in down	22622	28647	36102
Words in up	21382	27315	33488
Accuracy on test	59.91%	79.12%	87.22%
Accuracy training	99.4%	98.12%	94.61%

The change of accuracy for different number of training sets in normal Bayes classifier and modified Bayes classifier is shown below,

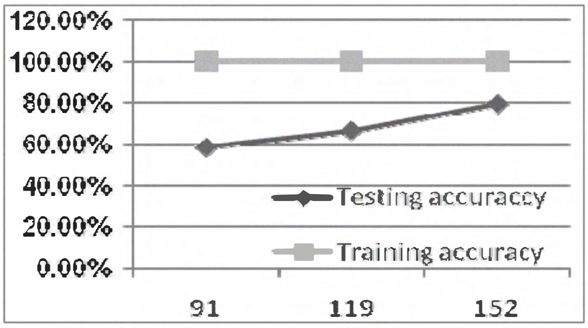

Figure 1: Change in accuracy at different the number of training sets in normal Bayes classifier

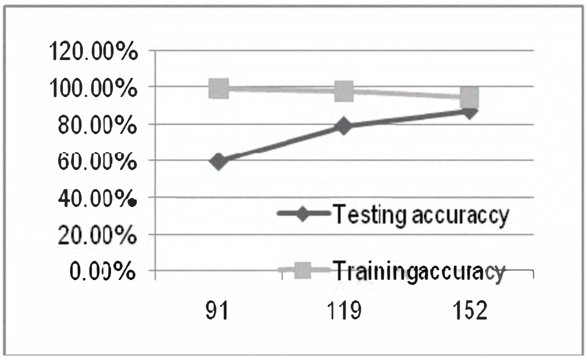

Figure 2: Change in accuracy at different the number of training sets in modified Bayes classifier.

The graphs show in both the cases that the accuracy of right classification is increasing with the increase in the number of training sets. In accordance with the current number of training sets the performance is quiet satisfactory. The other published papers shows maximum 90% accuracy on test sets, which is clearly attainable with this system by slight modification, more sophisticating data cleaning and increased number of reliable less noisy training sets.

The system was checked for predicting next day stock price by the previously mentioned method. If was trained on 152 training sets and tested on 35 test sets. For now, it is giving 50% accuracy in the Naïve bayes classifier, which means it, is not learning. More training using mentioned methods have a great chance of getting accuracy more than 50% Doing reasonable selection of the training data, and using large number of training sets the system have a greater chance of learning the future stock price. The used reasonable heuristics are useful in this case as well.

The training set used in the work is for only 7 months from single website for only closing reports on FTSE100. The size is not enough for the Naïve Bayes classifier to give its best result; but it can certainly give a satisfactory result to analyze the performance of the system. The accuracy will be increasing with the increase in training sets. The used number of sets is reasonably enough for learning the trends with less confidence. The target here is to check if the system is learning or not. Then with more time and effort, satisfactory level of data can be collected and feed into the system to learn.

The noise can certainly confuse the learner to find the right hypothesis. The noise of the web pages or in other words, the irrelevant news can increase the number of attributes having more neutral words. The stock price noise can have very bad impact on the performance, because the learner will then be confused and the useful attributes will be becoming more and more neutral means the probability for up and down will be close to equal on these attributes.

V. Conclusion

Text mining respects that complete understanding of natural language text is not immediately attainable and focuses on extracting a small amount of information from text with high reliability. The performance of the Naïve Bayes learner greatly depends on the frequency of the data instances, attribute values and target value for accuracy prediction. The increase of the training examples can certainly increase the performance of the system. Also better algorithm for noise cleaning and using expert's comments in the training sets can boost the performance accuracy even more

REFERENCE:

[01] Leung, S.: Automatic Stock Market Prediction from World Wide Web Data. MPhil thesis, The Hong Kong University of Science and Technology, 1997.

[02] Peramunetilleke, D.: A System for Exchange Rate Forecasting from News Headlines. MPhil thesis, The Hong Kong University of Science and Technology, 1997.

[03] Meretakis, D. W¨uthrich, B.: Classification as Mining and Use of Labeled Itemsets.

[04] Nahm, U. Y., Mooney, R.J.: Using Information Extraction to Aid the Discovery to Prediction Rules from Text.

[05] Nahm, U. Y., Mooney, R.J.: A Mutually Beneficial Integration of Data Mining and Information Extraction. In: Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-2000), pp.627-632, Austin, 2001.

[06] Nahm, U. Y., Mooney, R.J.: Mining Soft-Matching Rules from Textual Data. In the Proceedings of the 17th International Joint Conference on Artificial Intelligence IJCAI-01, 2001.

[07] Nahm, U. Y., Mooney, R.J.: Text Mining with Information Extraction. AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases, Stanford, 2002.

[08] Macskassy, S.A., Provost,F.: Intelligent Information Triage. In: Proceedings of SIGIR'01, September 2001, New Orleans, USA.

[09] Macsakssy, S.A., Hirsh, H., Provost, F., Sankaranarayanan, R., Dhar, V.: Information Triage using Prospective Criteria. In: Proceedings of User Modeling Workshop: Machine Learning, Information Retrieval and User Learning, 2001.

[10] Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., Allan, J.: Language models for financial news recommendation. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 389-396, 2000.

[11] Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., Allan, J.: Mining of Concurrent Text and Time Series.

[12] Peramunetilleke, D., Wong, R.K.: Currency Exchange Rate Forecasting from News Headlines. In: Proceedings of the Thirteenth Australasian Database Conference ADC2002, Melbourne, Australia, 2002.

TABLE 1: EXAMPLE OF VOCABULARY MATRIX

Vocabulary index	Word Attributes	Appeared in down	Appeared in up	Probability down	Probability up	log_prob. Down	log_prob up
2.00	'shares'	490	440	0.0124	0.0112	-4.39	-4.49
3.00	'big'	56	33	0.0015	0.0009	-6.53	-7.04
4.00	'oil'	125	136	0.0032	0.0035	-5.75	-5.66
5.00	'explorers'	7	5	0.0002	0.0002	-8.49	-8.71