Combining of Random Forest Estimates using LSboost for Stock Market Index Prediction

Nonita Sharma
Department of Computer Science & Engineering
National Institute of Technology Delhi
New Delhi, India
nonitasharma@nitdelhi.ac.in

Akanksha Juneja
Department of Computer Science & Engineering
National Institute of Technology Delhi
New Delhi, India
akankshajuneja@nitdelhi.ac.in

Abstract— **This research work emphases on the prediction of future stock market index values based on historical data. The experimental evaluation is based on historical data of 10 years of two indices, namely, CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets. The predictions are made for 1–10, 15, 30, and 40 days in advance. This work proposes to combine the predictions/estimates of the ensemble of trees in a Random Forest using LSboost (i.e. LS-RF). The prediction performance of the proposed model is compared with that of well-known Support Vector Regression. Technical indicators are selected as inputs to each of the prediction models. The closing value of the stock price is the predicted variable. Results show that the proposed scheme outperforms Support Vector Regression and can be applied successfully for building predictive models for stock prices prediction.**

Keywords— Stock Market Prediction; Regression; Least Square Boost; Random Forest

I. Introduction

Stock price prediction is one of the most important and challenging tasks due to its highly dynamic nature. Efficient-market hypothesis [1] suggests that one cannot rely on the information obtained at the point of investment to achieve returns more than the average market returns. At the same time, technical analysts believe that the trends in the price movement of the stocks can be used to find the predictions of the stock prices. Additionally, several other economic factors, like political factors, company strategies, global market conditions, market trends, repo rate etc., also influence the movement of the stock prices [2]. The stock indices are calculated on the basis of stocks having high market capitalization so as to reflect the overall economy of the country. Further, several technical indicators are also used to obtain statistical figures from the stocks prices.

Moreover, there are several different approaches for time series modeling. Classical statistical models use linear models to make the future predictions of the stock values [3, 4, 5]. Various prediction algorithms using state-of-the-art techniques like Support Vector Regression, Genetic Algorithms, Fuzzy Logic and several other techniques are proposed and proven to give better results than the traditional algorithms [6, 7, 8]. Support Vector Regression (SVR) [9] is so far, the most widely used machine learning algorithm for predicting the stock price values. The study by Patel et al. [10] evaluated the effectiveness of using technical indicators, such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), relative Root Mean Squared Error (rRMME), and Mean Squared Error (MSE) in predicting movements of Nifty and Sensex using SVR.

In this research work, Least Square Boost (LSboost) strategy is applied as a training loss function for combining the estimates of ensemble of multiple trees in random forest [11]. Random forest is a well-defined ensemble learning technique used for supervised learning. In case of stock price prediction, where the dataset is extremely random, one single regression model is not sufficient. Therefore an ensemble of different regression models is used to make predictions. The results obtained by the ensemble of regression models usually outperform the results obtained from a single regression model. Hence, in random forest, input data is fit into random subsets using several Classification and Regression Tree (CART) models and the aggregated result of the forest is used for prediction. Additionally, random forests evaluate the weights of each independent variable to model the dependent variable. However, the training loss function i.e. Mean Squared Error (MSE) used in Random Forest produces poor estimates in certain scenarios [12]. Hence, LSboost is used as a training loss function to improve the error estimates and hence making it more attractive to solve the prediction problem. This paper focuses on the task of predicting future values of the stock market indices. The predictions are made for 1–10, 15, 30 and 40 days in advance.

II. Methodology

The learning algorithm used in this research work is random forest. The time series data is acquired, smoothed and the technical indicators are extracted. Technical indicators are parameters which provide insights to the expected stock price behavior in future. These technical indicators are then used to train the random forest. In this section, the details of each step are discussed.

A. Preprocessing

The preprocessing of the data involves the exponential smoothing of time series historical stock data. In this, recent observations are assigned more weightage and the weightage keeps on reducing as the observations get older. For time series *X*, the exponential smoothing can be iteratively performed as:

$$S_0 = X_0 \tag{1}$$

978-1-5090-4307-1/17/$31.00 ©2017 IEEE

$$for\ t > 0,\ S_t = sf\ * X_t + (1\text{-}\ sf) * X_{t\text{-}1} \qquad (2)$$

Where sf is the smoothing factor, whose value ranges between 0 and 1.This preprocessing removes outliers, noise and missing data from the historical data, making it suitable for identifying the price trend in the stocks values. From the smoothed time series data, the technical indicators are calculated, which is based on the prediction of the target price value i.e. TP_i of the i^{th} day as :

$$TP_i = Sign(CP_{i+d} - CP_i) \qquad (3)$$

Where d is the number of days for which the prediction is to be done. Sign of TP_i determines the price shift, when it is positive, it indicates the positive shift in the prices of stock after d days and vice versa.

B. Features

Feature extraction is done on the basis of the technical indicators calculated from smoothed time series data, aiming to forecast the stock price movement. These are extensively used by analysts to check for the direction of stock price movements. The indicators used in case of CNX Nifty data are opening price, high price, low price, shares traded, and turn over (Rs. Cr) and the dependent/predicted variable is the closing price. While in case of S&P BSE Sensex data, opening, high, and low prices are used as indicators for predicting the closing price of the stocks.

C. Proposed Method - Least Square Boost based Random Forest (LS-RF)

CARTs are widely used for several analysis and prediction applications. But the trees that are built to learn extremely irregular patterns incline to over fit the training sets. Noisy data or an outlier may cause the tree to grow in an entirely different manner as the decision trees are very simple and predictive models having high variance and low bias. Hence, this problem is overcome by Random Forest by training several CARTs on multiple function space at the cost of slightly increased bias. This implies that none of the decision trees in the random forest gets the whole training data. The training data is recursively divided into partitions. The splitting is done using mean square error as measures of impurity. Once all the decision trees are formed, their predictions are combined to give a final prediction. LSboost is a way of combining outputs of multiple CART learners in order to achieve enhanced performance. Additionally, it is used to reduce the variance as well as over fitting of the decision tree.

The tree ensemble model used in our proposed technique is linear model in which the regression/dependent variable m_i is determined from the set of predictor variables x_j, which can be represented as:

$$m_i = \sum_j \theta_j x_{ij} \qquad (4)$$

Where θ_j are the values determined from the data. Let r_i be the actual value of the dependent variable for i^{th} sample. Tree ensemble model is a set of CART trees, where the sum of the predictions of multiple trees is taken as:

$$y_i = \sum^{k}_{i=1} mi\ xij,\ \forall j, \forall m_k \in M \qquad (5)$$

Where k represents the total number of trees in the random forest and M is the functional space. The performance of the model is determined from the objective function, which is determined from the training loss $L(\theta)$ and regularization term $\tau(\theta)$

$$Obj(\theta) = \sum^{k}_{i=1} L(\theta)_i + \sum^{k}_{i=1} \tau(\theta)_i \qquad (6)$$

The training loss $L(\theta)$ measures the predictive accuracy of the model, it uses the logistic loss for logistic regression.

$$L(\theta) = \sum_{I} [y_i \ln(1+ e^{-y_i})] + [(1 - y_i) \ln(1+ e^{y_i})] \qquad (7)$$

Regularization term $\tau(\theta)$, helps to avoid the overfitting problem and is added to control the complexity of the problem. The LS-RF aims to find k CARTs where each CART predicts y_i values corresponding to which $Obj(\theta)$ is minimum and squared error is minimum. Let **T** denote the outcome of the random forest (i.e. ensemble of k CARTs).

Algorithm: LS-RF

```
Input: Dataset D (set of predictor variables and regression/dependent variable)
for $t$ = 1 to $k$
    $d \subset D$
    $T_t \leftarrow$ CART (d) where CART (d) = $argmin (\sum^{|d|}_{i=1}(y_i - r_i)^2)$
end for
return T
```

III. Experimental Results

In this work, a LSboost-based Random Forest regression model is proposed for stock market price prediction. This section presents the details of the dataset utilized, the experimental setup, and discussion of the results obtained on applying the proposed method (LS-RF) on the given dataset. Also, the performance of the proposed method is compared with that of the well-known Support Vector Regression on the same dataset and experimental setup.

A. Datasets

This study uses total ten years of historical data from Jan 2006 to Dec 2015 of two stock market indices CNX Nifty and S&P BSE Sensex which are highly voluminous. All the data is obtained from http://www.nseindia.com/ and http://www.bseindia.com/websites.

B. Experimental Setup

All experiments are carried out using Windows 10 environment over Intel core i7 machine with 8 GB RAM and a processor speed of 3 GHz. Matlab 2016 is utilized for experiments. In LS-RF, the number of trees in the ensemble is 100.

C. Evaluation Measures

Four well-known metrics, namely, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), relative Root Mean Squared Error (rRMSE) and Mean Squared Error (MSE) are used to evaluate the performance of the regression models. Mathematical notations of these evaluation measures are shown in Eqs. 8-11 as follows:

$$MAPE = 1/n \sum^{n}_{i=1} (|A_i - P_i|/|A_i|) \times 100 \qquad (8)$$

$$MAE = 1/n \sum^{n}_{i=1} (|A_i - P_i|/|A_i|) \qquad (9)$$

$$rRMSE = sqrt(1/n \sum^{n}_{i=1} ((A_i - P_i)/A_i)^2) \qquad (10)$$

$$MSE = 1/n \sum^{n}_{i=1} ((A_i - P_i)/A_i)^2) \qquad (11)$$

where A_i and P_i are the actual and the predicted values for the i^{th} day. n is the total number of days for which prediction is made.

IV. Results and Discussion

The results obtained for the proposed LS-RF method and SVR method are shown in Table I and Table II for Nifty CNX data and S&P BSE Sensex data respectively, over four performance measures i.e. MAPE, MAE, rRMSE, and MSE. Both the Tables show performance for predictions made 1-10, 15, 30, and 40 days in advance.

TABLE I. Prediction performance for NIFTY CNX data

Prediction Model	**Error Measures**			
	MAPE	**MAE**	**rRMSE**	**MSE**
1 day ahead of time				
LS-RF	0.2573	0.0025	0.0025	420.1312
SVR	2.0085	0.0201	0.0201	25581.5150
2 days ahead of time				
LS-RF	0.3978	0.0039	0.0042	1089.2167
SVR	1.1800	0.0118	0.0144	13165.9144
3 days ahead of time				
LS-RF	0.4096	0.0041	0.0043	1105.0762
SVR	1.1891	0.0119	0.0137	11720.7024
4 days ahead of time				
LS-RF	0.4738	0.0047	0.0049	1494.0803
SVR	1.1357	0.0114	0.0128	10216.8944
5 days ahead of time				
LS-RF	0.5709	0.0057	0.0062	2250.4284
SVR	0.9969	0.0099	0.0116	8396.6995
6 days ahead of time				
LS-RF	0.6390	0.0064	0.0069	2799.2322
SVR	1.0341	0.0103	0.0117	8432.2734
7 days ahead of time				
LS-RF	0.5629	0.0056	0.0064	2408.5767
SVR	1.1199	0.0112	0.0125	9412.0410
8 days ahead of time				
LS-RF	0.5557	0.0056	0.0062	2287.5788
SVR	1.0663	0.0107	0.0119	8572.0829
9 days ahead of time				
LS-RF	0.7181	0.0071	0.0089	4618.7898
SVR	1.1203	0.0112	0.0124	9149.6747
10 days ahead of time				
LS-RF	0.8130	0.0081	0.0099	5736.0129
SVR	1.1524	0.0115	0.0126	9414.8301
15 days ahead of time				
LS-RF	0.7052	0.0071	0.0089	4532.1186
SVR	0.9881	0.0099	0.0115	7684.6964
30 days ahead of time				
LS-RF	1.0111	0.0101	0.0181	16565.9667
SVR	1.1252	0.0112	0.0126	8987.9900
40 days ahead of time				
LS-RF	1.3063	0.0131	0.0228	25937.6635
SVR	1.1657	0.0117	0.0132	9589.6024

TABLE II. Prediction performance for S&P BSE SENSEX data

Prediction Model	**Error Measures**			
	MAPE	**MAE**	**rRMSE**	**MSE**
1 day ahead of time				
LS-RF	0.0240	0.0002	0.0002	39.3513
SVR	4.5435	0.0454	0.0454	1412799.3827
2 days ahead of time				
LS-RF	0.2495	0.0025	0.0034	7424.2006
SVR	3.8135	0.0381	0.0388	1018519.6376
3 days ahead of time				
LS-RF	0.1726	0.0017	0.0027	4957.2829
SVR	3.8199	0.0382	0.0387	999443.3589
4 days ahead of time				
LS-RF	0.1980	0.0020	0.0027	4932.4472
SVR	3.7748	0.0377	0.0381	963306.0500
5 days ahead of time				
LS-RF	0.2173	0.0022	0.0028	5016.9331
SVR	3.6304	0.0363	0.0367	885782.6023
6 days ahead of time				
LS-RF	0.2654	0.0027	0.0033	6830.4108
SVR	3.6537	0.0365	0.0369	885410.3178
7 days ahead of time				
LS-RF	0.2497	0.0025	0.0031	6067.8055
SVR	3.7087	0.0371	0.0374	902542.8814
8 days ahead of time				
LS-RF	0.3105	0.0031	0.0039	9433.4056
SVR	3.6813	0.0368	0.0371	882470.2856
9 days ahead of time				
LS-RF	0.3962	0.0040	0.0051	16420.2627
SVR	3.7595	0.0376	0.0379	916365.3814
10 days ahead of time				
LS-RF	0.4143	0.0041	0.0052	16821.2848
SVR	3.7965	0.0380	0.0383	929402.3120
15 days ahead of time				
LS-RF	0.5745	0.0057	0.0071	30133.2620
SVR	3.6091	0.0361	0.0365	829977.9021
30 days ahead of time				

LS-RF	0.7594	0.0076	0.0110	66525.4960
SVR	3.5440	0.0354	0.0361	795074.2902
40 days ahead of time				
LS-RF	0.9770	0.0098	0.0135	99866.9207
SVR	3.4667	0.0347	0.0354	746928.4644

From Tables I-II it can be observed that, on the whole, the proposed LS-RF prediction model outperforms the well-known SVR prediction model for both the datasets, in terms of all the four performance evaluation metrics. This may be attributed to the fact that LS-RF builds and learns an ensemble of trees (forest) whose predictions are combined using least square boosting method. However, SVR, even though it is a popular and powerful approach, is based on building a single regression model.

CONCLUSION

In this paper, the focus is to predict the future values of stock market indices based on the previous stock values using regression. Experiments are carried out on ten years of historical data (January 2006 to December 2016) of two indices namely CNX Nifty and S&P BSE Sensex from Indian stock markets. The predictions are made for 1–10, 15, 30, and 40 days in advance.

The proposal of least square boost based random forest is a significant research contribution of this paper as this scheme provides a new way of combining the estimates of ensemble of trees for prediction models. To accomplish this, machine learning methods are carried out in two stages. First stage uses Random Forest regression to predict future values of statistical parameters which are fed as the inputs to the prediction models. In the second stage, Least Square Boosting is used as a training loss function to improve the error estimates. The experimental results are encouraging and establish the usefulness of the proposed approach.

In the stock market, the term ‘stop-loss order’ is used by investors to state the order to close a position whenever loss reaches a threshold. In this research work, predictions are made for 1–10, 15, 30, and 40 days in advance. This can enable investors by predicting the ‘stop-loss order’ for 1–10 days or 15 or 30 or 40 days. As the proposed method achieves reduced prediction error, investors can ensure booking lesser amount of loss or more amount of profit.

In future, the proposed prediction model may also be implemented in other areas like GDP forecasting, energy consumption forecasting, or weather forecasting.

REFERENCES

[1] Fama and F. Eugene, "Random walks in stock market prices," Financial analysts journal 51, vol. 1, pp. 75-80, 1995.

[2] Miao, Kai, C. Fang and . Z. G. Zhao., "Stock Price Forecast Based on Bacterial Colony RBF Neural Network [J]," 2007.

[3] Bollerslev and Tim, "Generalized autoregressive conditional heteroskedasticity," Journal of econometrics 31, vol. 3, pp. 307-327, 1986.

[4] Hsieh and A. David, "Chaos and nonlinear dynamics: application to financial markets.," The journal of finance 46, vol. 5, pp. 1839-1877, 1991.

[5] Rao, . T. Subba and M. M. Gabr, An introduction to bispectral analysis and bilinear time series models, vol. 24, Springer Science & Business Media, 2012.

[6] Hadavandi, Esmaeil, S. Hassan and G. Arash , "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting," Knowledge-Based Systems 23, vol. 8, pp. 800-808, 2010.

[7] Lee, Yi-Shian and Lee-Ing Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," Knowledge-Based Systems 24, vol. 1, pp. 66-72, 2011.

[8] M. H. Zarandi, H. Esmaeil and I. B. Turks, "A hybrid fuzzy intelligent agent-based system for stock price prediction," International Journal of Intelligent Systems 27, vol. 11, pp. 947-969, 2012.

[9] Welling and Max, "Support vector regression," 2004.

[10] J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," Expert Systems with Applications 42, vol. 4, pp. 2162-2172, 2015.

[11] Breiman and Leo, "Random forests," in Machine learning 45, 2001.

[12] Strobl, Carolin and Z. . Achim, "Danger: high power!–exploring the statistical properties of a test for random forest variable importance," 2008.