Professor Kyoung-Sook MOON, PhD
E-mail: ksmoon@gachon.ac.kr
Mathematical Finance, Gachon University
Sookyung JUN
E-mail: tnzzzrud@korea.ac.kr
Professor Hongjoong KIM, PhD
E-mail: hongjoong@korea.ac.kr
Mathematics, Korea University

SPEED UP OF THE MAJORITY VOTING ENSEMBLE METHOD FOR THE PREDICTION OF STOCK PRICE DIRECTIONS

Abstract. *The prediction of stock price directions is important in finance. The Majority Voting Ensemble method is superior in prediction accuracy to single classifier models including Logistic Regression, Decision Tree, K-Nearest Neighbors and Support Vector Machine, but the computational cost is very expensive since it considers all the hyperparameters of single classifier models. The current study proposes a revision of the majority voting method to improve the computational efficiency. The proposed method lets each single classifier model find its own hyperparameter values and this modification speeds up the computation by 500 times compared to the standard majority voting method while maintaining the accuracy. The numerical experiments show the ranking of the classifier models in the order of the proposed majority voting, the standard majority voting, and then other single classifier models including the support vector machine. This improvement will allow the majority voting ensemble method to be applied in the financial market in practice. The algorithms are tested on 7 national indices from 3 continents for the past 3 years, and the performance is measured in two criteria, the area under the receiver operating characteristic curve and the percent correctly classified.*

Keywords: *forecasting, machine learning, ensemble method, majority voting, stock price prediction.*

JEL Classification: C40, G17

1. Introduction

The prediction of stock price directions is an important issue in finance for investment banks or individual investors to make a profit or hedge against market risks. Making precise predictions of the stock price direction, however, is

DOI: 10.24818/18423264/52.1.18.13

challenging since the market data is non-stationary and noisy. Many researchers have been working on the stock prediction problems to find satisfactory solutions from huge market data. Fortunately, there have been developed several statistical learning methods as efficient tools for understanding the data and reported that these methods perform well in various areas such as pattern recognition, signal processing, medicine, biology, and engineering. See (Vapnik, 2013; Hastie et al., 2013; James et al., 2013; Raschka, 2015).

Based on stock market data with daily movements, one wants to predict whether the stock price or index will increase or decrease on a future day. This is a classification problem and there have been explored several different statistical learning methods such as Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Ensemble methods. See (Ballings et al., 2015; Tsai et al.,2011; Patel et al., 2015; Kara et al., 2011; Oztekin et al.,2016) and references therein.

As pointed out by (Ballings et al., 2015) and (Tsai et al., 2011), even though ensemble methods are proved to perform well in many areas, there are not many works based on the ensemble methods in stock price prediction problems. One of the reasons is that ensemble methods need huge computational efforts compared to other single classifier models. Authors in(Ballings et al., 2015) studied benchmark statistics of ensemble methods against single classifier models for the stock price direction prediction. A hybrid method of majority voting(Tsai et al., 2011) was studied and authors in(Patel et al., 2015) experimentally showed that the random forest ensemble method outperforms in Indian stock markets.

In this paper, we propose a simple modification of the majority voting ensemble method to reduce its computational cost. The key idea of the proposed method is to allow each single classifier model to seek its own parameters. We compare the performance of the proposed method with well-known machine learning algorithms including LR, DT, KNN, SVM and the standard majority voting ensemble method on the 7 national indices such as S&P 500, NASDAQ (North America), KOSPI 200, NIKKEI 225, HANG SENG (Asia), DAX, FEZ (Europe) for the past 3 years from December 2012 to December 2015. These experimental tests in Section 4 illustrate that this modification reduces work significantly and achieves the computational speed 500 times faster than the standard majority voting method, while maintaining the level of accuracy. This improvement in efficiency will allow the ensemble method to be applied in practice.

The outline of the paper is as follows. In Section 2, single classifier models and standard ensemble methods are briefly described. Majority voting ensemble method and its modification are explained in Section 3. The performance of different statistical learning methods is compared based on real market data in Section 4. The conclusions and future research direction are drawn in Section 5.

DOI: 10.24818/18423264/52.1.18.13

2. Methodology

2.1 Logistic Regression

Since Cox (Cox, 1958), Logistic Regression (LR) has been used in many fields including machine learning, medical fields, engineering, economics and finance. See (Pampel, 2000) and (Hosmer et al., 2013). LR models the probability of a binary response based on predictors or features. Given N observed data points, one considers m features, $x_1^i, x_2^i, \cdots, x_m^i$ and associated outcome variables, $z^i = \sum_{j=0}^{m} w_j x_j^i$ satisfying $x_0^i = 1$ and $\sum_{j=0}^{m} w_j = 1$, for each data point, $i = 1, 2, \cdots, N$. The coefficients, $w = (w_0, w_1, \cdots, w_m)$ are then computed by minimizing the following objective function of the prediction

$$J(w) = \sum_{\{1 \le i \le N\}} \left[-y^i \log\left(\varphi(z^i)\right) - (1 - y^i)\log\left(1 - \varphi(z^i)\right)\right] + \frac{1}{2}\lambda \parallel w \parallel^2$$

where y^i is the class label and $\parallel w \parallel = \sqrt{w_0^2 + w_1^2 + \cdots + w_m^2}$.Here the sigmoid function $\varphi(z) = \frac{1}{1+e^{-z}}$ is interpreted as the probability that a particular sample belongs to the expected class. The regularization term in $J(w)$ with the hyperparameter $\lambda \ge 0$ is usedin order to prevent over-fitting (Friedman et al. 2010).In Section 4, we use the *scikit-learn* python package called *LogisticRegression*with the hyperparameter $\lambda \in \{10^{-3}, 10^{-2}, \cdots, 10^4\}$ as in (Raschka, 2015).

2.2 Decision Tree

Decision tree (DT) is a popular classification model with its simple interpretation and robust performance as explained in (Rokach and Maimon, 2014). DT learning creates a model that predicts the value of a target variable based on several input variables by recursive partitioning. A variable is chosen at each step which best splits the set of samples. Different impurity measures or splitting criteria can be used in binary decision trees, such as Gini impurity (which measures how often a randomly chosen element from the set would be incorrectly labeled), information entropy (which measures the change in information entropy from a parent node to the sum of children nodes) or classification error (which measures the minimum of the proportion of the samples). The depth is the parameter for the numerical amount of splitting in the decision trees. The deeper the depth is, the higher accuracy of training set one obtains, though there exists more risk of over-fitting for deeper depth. In Section 4, we use the *scikit-learn* python package called *DecisionTreeClassifier* based on the information entropy with the depth values in $\{1, 2, 3, \cdots, 7\}$ to generate decision trees as in (Raschka, 2015).

DOI: 10.24818/18423264/52.1.18.13

2.3. K-Nearest Neighbors

K-nearest neighbors (KNN) chooses the class label of the new data point by a majority vote among its k nearest neighbors. The k nearest neighbors are determined by the chosen distance metric, the function, $\mathrm{dist}(x^i, x^j) = \sqrt[p]{\sum_n | x_n^i - x_n^j |^p}$ with the parameter $p \in \{1, 2, 3, 4\}$ is used as a distance metric.KNN method is simple and straightforward to implement, but it is sensitive to the local structure of the data and the computational complexity for classifying new samples grows linearly with the number of samples in the training set. The parameter k can be chosen depending on the data and in general larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. In Section 4, the *scikit-learn* python package called *KNeighborsClassifier* with the parameter $k \in \{1, 2, 3, 4\}$ is used for the computation as in (Raschka, 2015).

2.4. Support Vector Machine

Support Vector Machines (SVM) is popular for classification and regression analysis, and it works well in linear and non-linear classifications by implicitly mapping their inputs into high dimensional feature spaces as explained in (Cortes and Vapnik, 1995). SVM minimizes the misclassification error by maximizing the margin, the distance between the separating hyperplane and the training samples that are closest to this hyperplane. If the input data are not linearly classifiable, one need to relax the linear constraints and to control the trade-off between the misclassification errors and error penalties, we introduce a soft margin variable. The hyperparameter C related to the soft margin controls the width of the margin and tune the bias-variance trade-off. The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter. For the classification of the data, one of the most widely used kernel is the Radial Bases Function (RBF), $\kappa(x^i, x^j) = \exp(-\gamma \| x^i - x^j \|^2)$, or the linear function $\kappa(x^i, x^j) = (1 + x^i z^i)(1 + x^j z^j)$, where x^i is the i^{th} feature and z^i is the corresponding outcome, and γ is a free parameter that is to be optimized. In Section 4, SVM is implemented based on the *scikit-learn* python package called *SVC* with the parameters $\gamma \in \{0.3, 0.5, 0.7, 0.9, 1.0\}$ and $C \in \{10^{-3}, 10^{-2}, \cdots, 10^2\}$ as in (Raschka, 2015).

2.5.Ensemble method

Ensemble methods such as Majority Voting (MV) and Random Forest (RF) use multiple learning methods to obtain better performance than single classifiers. See (Seni and Elder, 2010) and(James et al., 2013). Empirically, ensembles turn out to yield better results when there is significant diversity among the models. Figure 1 shows the experimental results from single classifier methods (LR, DT, KNN,

DOI: 10.24818/18423264/52.1.18.13

SVM) and ensemble methods (MV, RF) applied to 7 national indices for the past 3 years from 2012 to 2015. The average area under the receiver operating characteristic curve(AUC) are compared. The values of features and variables for Figure 1 are described in detail in Section 4.

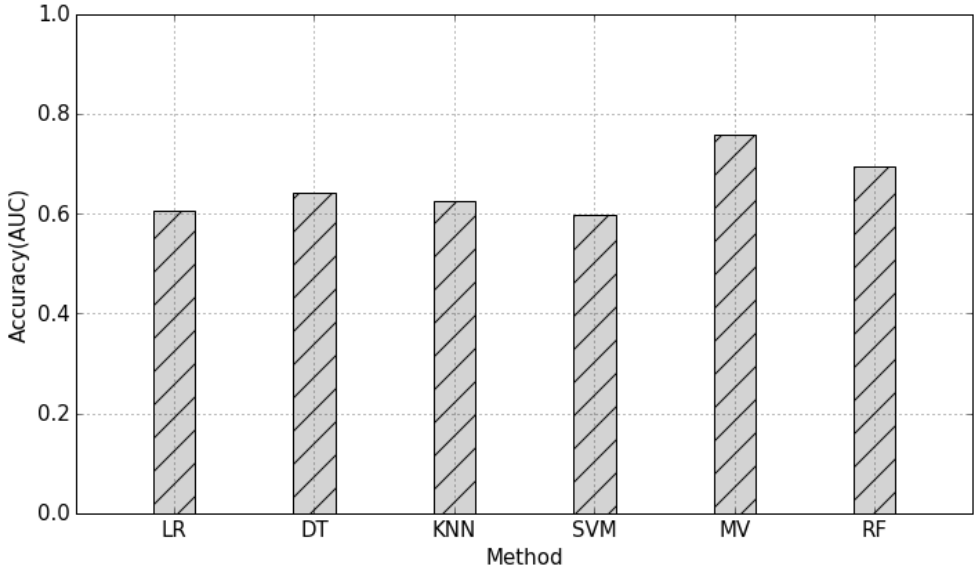

Figure 1.The accuracy of single classifiers (LR, DT, KNN, SVM) and ensemble methods (MV, RF) when the numbers of training and test data are 200 and 50, respectively

As Figure 1 shows and (Ballings et al., 2015) (and references there-in) points out, the ensemble methods like MV and RF are more powerful than single classifier models. In this experiment, the ensemble method based on MV is slightly better than that based on RF. Even though the regression ensemble method such as RF may be used for the prediction of the stock price direction, the classification such as MV seems to be more appropriate for such a prediction. Therefore, we focus on the classification approach based on MV in this study. Evaluating the prediction of ensembles, however, typically requires more computation than single classifier models. The computational work from MV measured by CPU time (second) increases around 650 times than that from SVM in this study as seen in Section 3. In order to overcome this slow computation, a revised MV method called 'Regional Majority Voting (RMV)' is proposed in Section 3.

The hyperparameters of classifier methods implemented in this study are summarized in Table 1 as in (Raschka, 2015).

DOI: 10.24818/18423264/52.1.18.13

Table 1.The values of hyperparameters for the classifier models.

Method	hyperparameters		Values of hyperparameters used in the computation
Logistic Regression	λ		$10^{-3}, 10^{-2}, \cdots, 10^{4}$
Decision Tree	Depth		$1, 2, 3, \cdots, 7$
KNN	p		$1, 2, 3, 4$
	k		$1, 2, 3, 4$
SVM	Kernel = linear	C	$10^{-3}, 10^{-2}, \cdots, 10^{2}$
	Kernel = RBF	C	$10^{-3}, 10^{-2}, \cdots, 10^{2}$
		γ	$0.3, 0.5, 0.7, 0.9, 1.0$
Majority Voting	All the parameters above		

Table 2.CPU time for single methods and ensemble methods

	Logistic Regression	Decision Tree	KNN	SNM	Majority Voting
CPU time	0.35 sec	0.03 sec	0.73 sec	6.84 sec	4436.90 ec

3. Regional Majority Voting

The Majority Voting explores a set of classifiers to combine predictions from the individual methods anticipating improvements in prediction. The current study considers 4 single classifier models in machine learning including LR, DT, KNN, and SVM. The hyperparameters for MV consist of all the hyperparameters from each individual classifier model. See Table 1. That is, the hyperparameters of MV are the product topology of each hyperparameter from single classifier models. Thus, the current study considers 32,256 number of hyperparameters and select the label that has been predicted by the majority of classifiers, which leads to huge computational costs as shown in Table 2 or as pointed in (Raschka, 2015). Table 2 shows the average computational cost when the experiments in Section 4 are performed. Even though the MV ensemble method performs better than single classifier models in general, it requires more computational time.

The current study proposes to distribute this finding procedure into each individual method. That is, once each method in the single classifier model finds its own optimal hyperparameters independently, those values selected from each method are directly used in MV, from which the computation can be simplified. Thus, the hyperparameter of the proposed revision of MV is the union, instead of the product topology, of each hyperparameter of single classifier models.Mathematically such an approach may put the hyperparameter set at a local critical point, not the global critical point. Sufficiently many computational simulations below in Section 4 show that the parameters found in this way are

DOI: 10.24818/18423264/52.1.18.13

quite effective and support numerically that the predictions based on this modification MV are almost as good as those from the standard MV. This MV using the hyperparameters proposed from each single classifier method is denoted Regional the Majority Voting (RMV) considering each single method as a region for voting.

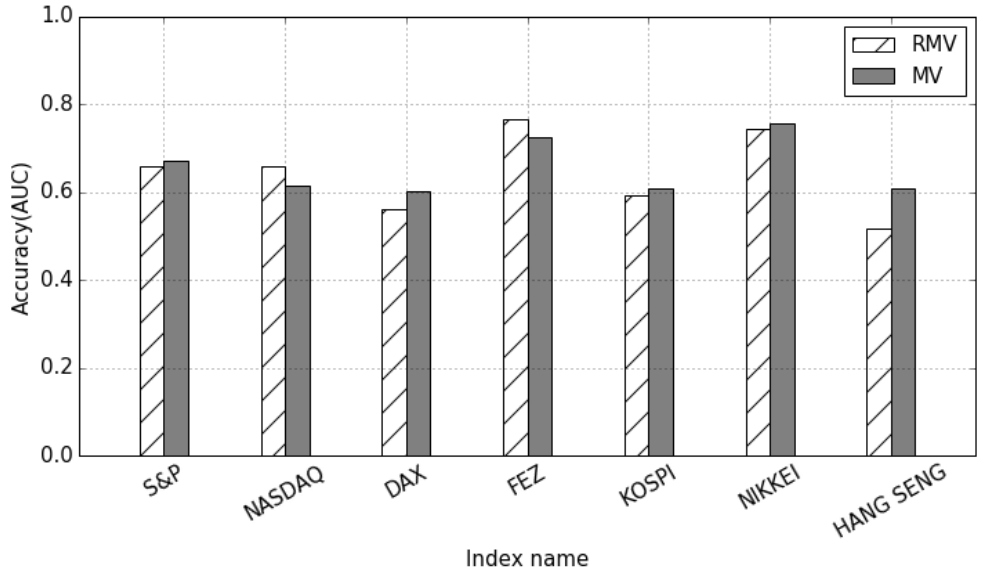

Figure 2.The accuracies for RMV and MV Ensemble methods

Figure 2 presents the accuracies of RMV and MV when the numbers of training and test sets are 200 and 50, respectively. The figure shows that the accuracies from the RMV and MV methods are similar. In some cases, RMV performs, on the contrary, better than the standard MV. But, the computational costs for these methods are quite different. Table 3 shows that the computational time is reduced by a factor of 500 with RMV. Figure 3 summarizes the architecture of the RMV ensemble method.

Table 3. The CPU time for Majority voting and Regional Majority Voting

	Majority Voting	Regional Majority Voting	Ratio
CPU time	4436.90 sec	8.85 sec	$\frac{4436.90}{8.85} = 501.34$

DOI: 10.24818/18423264/52.1.18.13

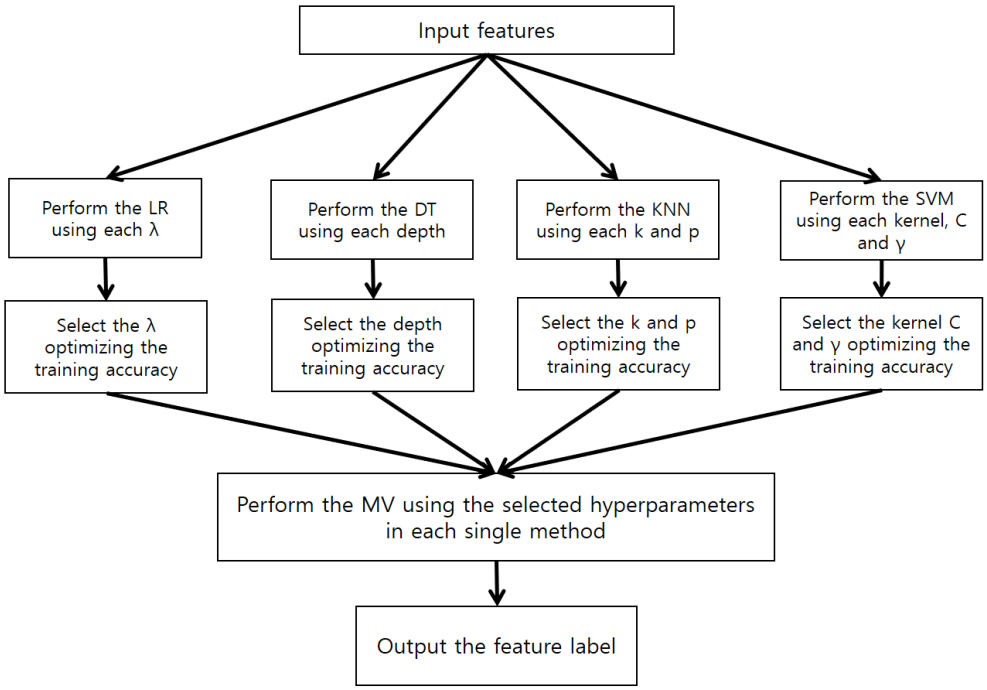

Figure 3.The architecture of the RMV ensemble method.

4. Numerical Results

4.1. Data and variables

The algorithms are tested on 7 national indices from 3 different continents for the past 3 years (from December 2012 to December 2015): S&P 500, NASDAQ (North America), KOSPI 200, NIKKEI 225, HANG SENG (Asia), DAX, FEZ (Europe). The features used in the training and tests are summarized in Table 4. The original stock or index values are represented by $\tilde{S}_j$. Here $H(x)$ denotes Heaviside function, $H(x) = 1$ if x is positive or $H(x) = 0$ otherwise. Since financial data is quite oscillatory, in order to help to capture meaningful information from the data we smoothen the data with the window size d in Table 4.

Table 4.Summary of features used in classifier models

Feature	Expression
Uniformly smoothed value (S_j)	$S_j = \frac{1}{d}\sum_{i=0}^{d-1} \tilde{S}_{j-i}$

DOI: 10.24818/18423264/52.1.18.13

Exponentially smoothed value (E_j)	$E_j = \frac{\sum_{i=0}^{d-1} e^{-i} \tilde{S}_{j-i}}{\sum_{i=0}^{d-1} e^{-i}}$
Difference (Δ_j)	$\Delta_j = S_j - S_{j-d}$
Range (G_j)	$G_j = \max_{0 \le i \le d-1} S_{j-i} - \min_{0 \le i \le d-1} S_{j-i}$
Mean (M_j)	$M_j = \frac{1}{d} \sum_{i=0}^{d-1} S_{j-i}$
Standard deviation (σ_j)	$\sigma_j = \frac{1}{d-1} \sum_{i=0}^{d-1} S_{j-i} - M_j^2$
The number of increases (N_j^+)	$N_j^+ = \sum_{i=0}^{d-1} H(S_{j-i} - S_{j-i-1})$
The number of decreases (N_j^-)	$N_j^- = \sum_{i=0}^{d-1} H(S_{j-i-1} - S_{j-i})$
The area of increases (A_j^+)	$A_j^+ = \sum_{i=0}^{d-1} S_{j-i}^+$ where $S_j^+ = \begin{cases} S_j & \text{if } S_j - S_{j-1} > 0 \\ 0 & \text{otherwise} \end{cases}$
The area of decreases (A_j^-)	$A_j^- = \sum_{i=0}^{d-1} S_{j-i}^-$ where $S_j^- = \begin{cases} S_j & \text{if } S_j - S_{j-1} < 0 \\ 0 & \text{otherwise} \end{cases}$
Correlation ($corr_j$)	$corr_j$ = correlation between $\{S_{j-i} : 0 \le i \le d-1\}$ and $\{S_{j-i-d} : 0 \le i \le d-1\}$
L1 norm ($L1_j$)	$L1_j = \frac{1}{d} \sum_{i=0}^{d-1} (S_{j-i} - S_{j-i-d})$
L2 norm ($L2_j$)	$L2_j = \frac{1}{d} \sqrt{\sum_{i=0}^{d-1} (S_{j-i} - S_{j-i-d})^2}$

The single classifier methods such as LR, DT, KNN, and SVM and the ensemble methods such as MV and RMV are used in this study. The parameters used in classifier models are summarized in Table 5.

DOI: 10.24818/18423264/52.1.18.13

Table 5.The summary of parameters and variables used in classifier models

Parameter	Value
Threshold	1% per 30 prediction length
Prediction length	$10, 20, 30, \cdots, 90$ days
Window size	$10, 20, 30, \cdots, 90$ days
The numbers of training and test set	(train, test) = (200, 50), (400, 100)

The performance is measured in two ways, the area under the receiver operating characteristic curve (AUC) and the percent correctly classified (PCC):

$$\mathrm{AUC} = \int_{-\infty}^{\infty} \mathrm{TPR(T)FPR'(T)dT}$$

and

$$\mathrm{PCC} = \frac{\mathrm{T}}{\mathrm{N}}$$

Here $\mathrm{TPR(T)} = \int_{\mathrm{T}}^{\infty} \mathrm{f(x)dx}$ and $\mathrm{FPR(T)} = \int_{\mathrm{T}}^{\infty} \mathrm{g(x)dx}$, where f(x) and g(x) are the probability density functions of the true positive rate and the false positive rate, respectively. N is the number of all elements and T is the number of correctly predicted elements. The machine learning algorithms above are applied to the indices, S&P 500 and NASDAQ from North America, KOSPI 200, NIKKEI 225, and HANG SENG from Asia, and DAX and FEZ from Europe. Since there are not much differences between AUC and PCC as seen in Figure 4, the AUC is used as accuracy measure for the examples below.

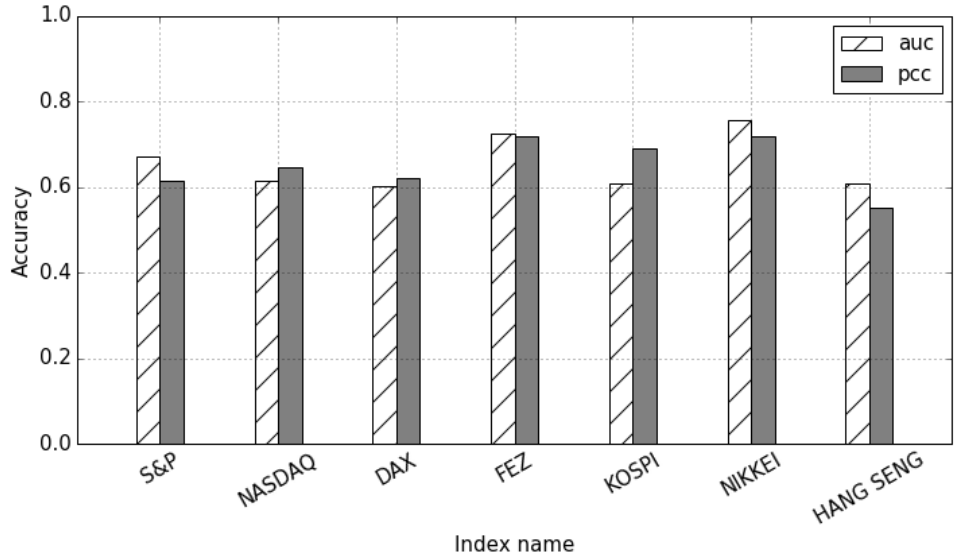

Figure 4.The average AUC and PCC values of Regional Majority Voting for various indices

DOI: 10.24818/18423264/52.1.18.13

4.2 Effects from different machine learning algorithms

Figure 5 compares the accuracies of different classifier methods. It shows the averages of AUC values for each index from 200 training sets and 50 test sets. The RMV ensemble method performs better than the other single methods.

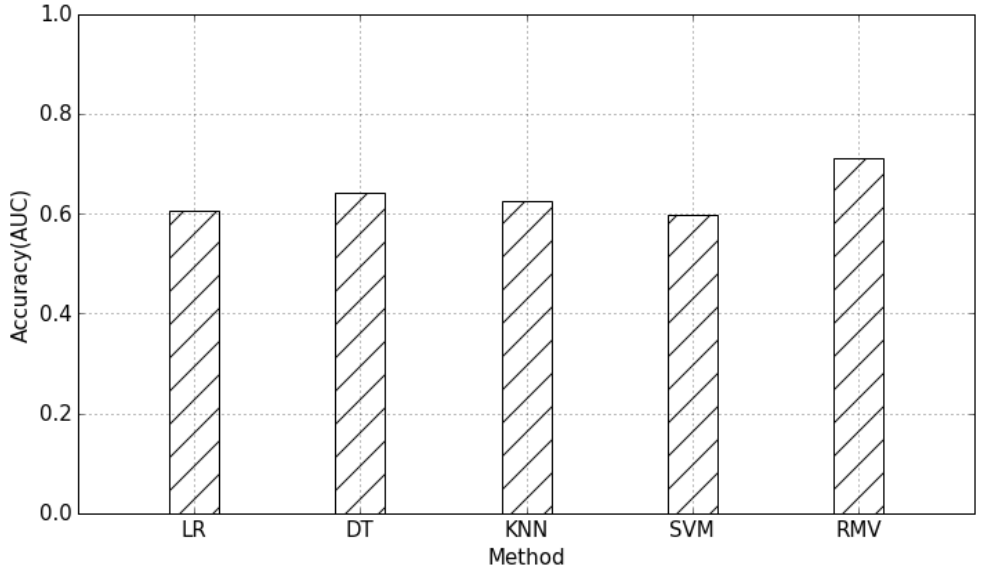

Figure 5.The accuracies for RMV and single methods with window size 20, when the numbers of training and test sets are 200 and 50, respectively

Figure 6 shows the averages of AUC values vs the averages of CPU times for single classifiers (LR, DT, KNN, SVM) and ensemble methods (RMV and MV). While both MV and RMV improve the accuracies over the single classifier methods, MV, contrary to RMV, requires a lot of computational efforts to get this difference.

DOI: 10.24818/18423264/52.1.18.13

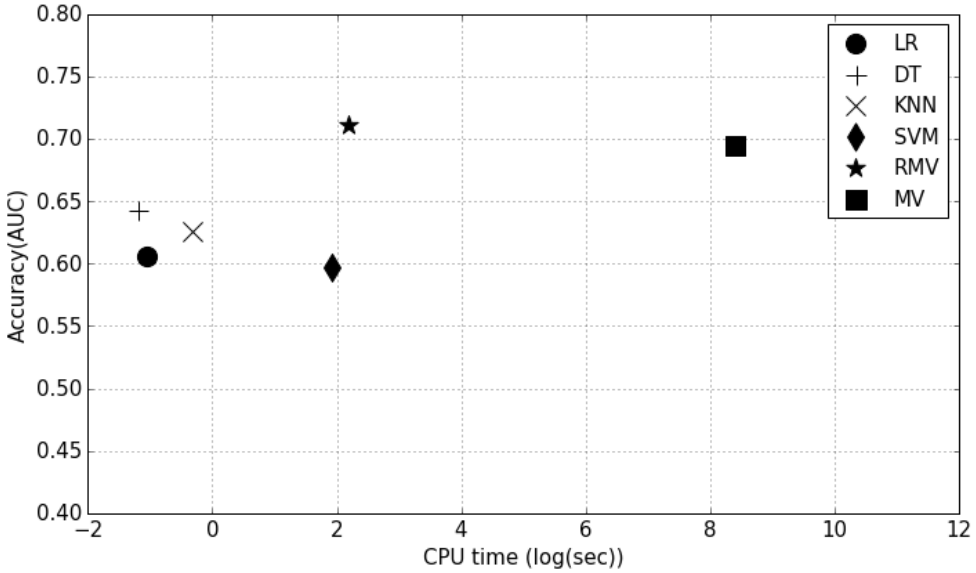

Figure 6.The accuracy and CPU time for each method

4.3. Effects from prediction length

Figure 7 shows the accuracies of SVM, MV, and RMV for different prediction lengths. The figure also shows that the ensemble methods by MV and RMV perform better than SVM and that the accuracies of MV and RMV are close. This again conforms that RMV is superior to MV when considering the difference in computational costs.

DOI: 10.24818/18423264/52.1.18.13

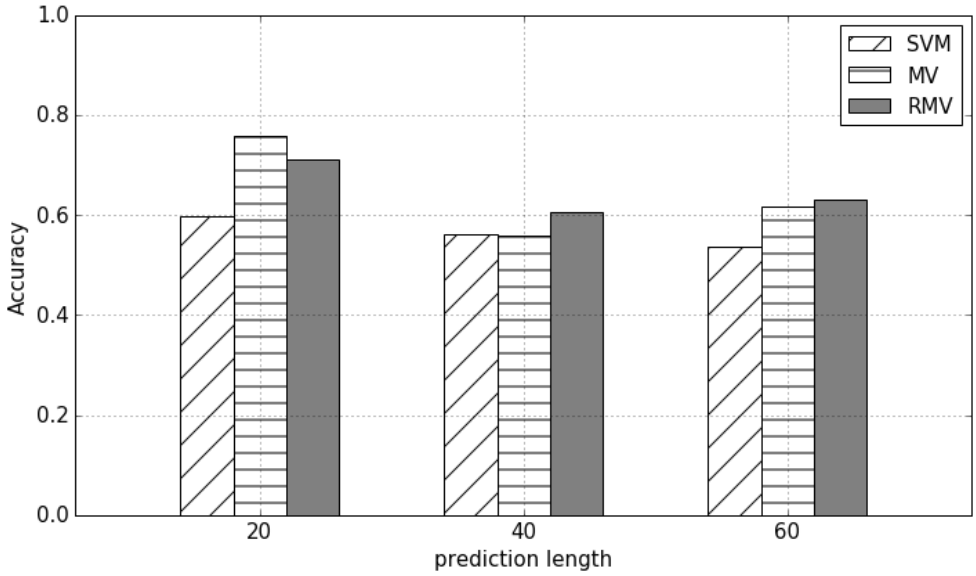

Figure 7. The accuracy for different prediction lengths when the numbers of training and test sets are 200 and 50, respectively

5. Conclusions

The ensemble method shows better prediction results compared to single classifier models but its heavy computational cost lowers its usage in practice. It is the novelty of this current study to clarify the weakness of the majority voting ensemble method for the first time especially in terms of the computational costs and to propose a numerical alternative to avoid it. The proposed method reduces its computational costs to about one hundredthwhile preserving the accuracy. This will allow the ensemble method to be practically applied to various fields including many problems in the financial markets.

Future research directions include analysis of mathematical aspects of machine learning algorithms, and applications to high frequency trading and multi-scale prediction with adaptive time scale. Also, it is interesting to study how to combine the prediction result with the efficient trading strategy for stock or currency market.

Acknowledgements

The research of Kim was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A1A2058271). This work of Moon was supported by the Gachon University research fund of 2017.

DOI: 10.24818/18423264/52.1.18.13

REFERENCES

[1]**Ballings, M., den Poel, D.V., Hespeels, N., Gryp, R. (2015),*Evaluating Multipleclassifiers for Stock Price Direction Prediction;*** Expert System with Applications, 42: 7046—7056;
[2]**Cortes, C., Vapnik, V. (1995),*Support-vector Networks;*** Machine Learning, 20: 273—297;
[3]**Cox, D.R. (1958),*The Regression Analysis f Binary Sequences;*** Journal of the Royal Statistical Society Series B, 20:215--242;
[4]**Friedman, J., Hastie, T., Tibshirani, R. (2010),*Regularization Paths for Generalized Linear Models via Coordinate Descent;*** Journal of Statistical Software, 33: 1—22;
[5]**Hastie, T., Tibshirani, R., Friedman, J. (2013), *The Elements of Statistical Learning:Data Mining, Inference, and Prediction,*** *Springer*, ISBN-13: 978-0387848570;
[6]**Hosmer, D.W., Lemeshow, S., Sturdivant, R.X. (2013),*Applied Logistic Regression*;** *Wiley*,ISBN-13: 978-0471356325;
[7]**James, G., Witten, D., Hastie, T., Tibshirani, R. (2013), *An Introduction Tostatistical Learning with Applications in R;*** *Springer*, ISBN-13: 978-1461471370;
[8]**Kara, Y., Boyacioglu, M.A., Baykan, O.K. (2011),*Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange;*** Expert Systems with Applications, 38: 5311—5319;
[9]**Oztekin, A., Kizilaslan, R., Freund, S., Iseri, A. (2016),*A Data Analytic Approach To Forecasting Daily Stock Returns in an Emerging Market;*** European Journal of Operational Research, 253: 697—710;
[10]**Pampel, F.C. (2000), *Logistic Regression: A Primer Quantitative Applications in the Social Sciences*;** SAGE Publications, Inc, **ISBN-13:** 978-0761920106;
[11]**Patel, J., Shah, S., Thakkar, P., Kotecha, K. (2015), *Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Prediction and Machine Learning Techniques;*** Expert Systems with Applications, 42: 259—268;
[12]**Raschka, S. (2015), *Python Machine Learning*;** Packt Publishing Ltd, ISBN-13: 978-1783555130;
[13]**Rokach, L., Maimon, O. (2014),*Data Mining with Decision Trees: Theory and Applications*;** World Scientific Publishing Company, ISBN-13: 978-9814590075;
[14]**Seni, G., Elder, J. (2010),*Ensemble Methods in Data Mining: Improving Accuracy through Combining Predictions;*** Morgan and Claypool Publishers, ISBN-13: 978-1608452842;
[15]**Tsai, C.F., Lin, Y.C., Yen, D.C., Chen, Y.M. (2011),*Predicting Stock Returns by Classifier Ensembles;*** Applied Soft Computing, 11: 2452—2459;
[16]**Vapnik, V.N. (2013), *The Nature of Statistical Learning Theory;*** *Springer*, ISBN-13: 978-0387987804.

DOI: 10.24818/18423264/52.1.18.13