Expert Systems with Applications 39 (2012) 11607–11615

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Forecasting trends of high-frequency KOSPI200 index data using learning classifiers

Youngdoo Son [a], Dong-jin Noh [b], Jaewook Lee [a,*]

[a] Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, South Korea
[b] Department of Industrial and Management Engineering, POSTECH, San 31, Hyoja-dong, Nam-gu, Pohang 790-784, South Korea

ARTICLE INFO

Keywords:
Financial forecasting
Statistical learning
Binary classification
Market lead-lag relationship
High-frequency trading

ABSTRACT

Recently many statistical learning techniques have been applied to the prediction of financial variables. The aim of this paper is to conduct a comprehensive study of the applications of statistical learning techniques to predict the trend of the return of high-frequency Korea composite stock price index (KOSPI) 200 index data using the information from the one-minute time series of spot index, futures index, and foreign exchange rate. Through experiments, it is observed that the spot index change is better predictable with high-frequency time series data and the futures index information significantly improves the prediction accuracy of the return trends of the spot index for high-frequency index data, while the information of exchange rate does not. Also, dimension reduction process before training helps to increase the accuracy and dramatically for some classifiers. In addition, the trained classifiers with which a virtual trading strategy is applied to, noticeable better profits can be achieved than just a buy-and-hold-like strategy.

1. Introduction

The financial variable prediction has been a long and yet active research theme targeted by many researchers since successful prediction helps to make profits as well as avoid risks. There have been many approaches especially for the stock market. Many people, called chartists, have been trying to predict the stock price or some other financial variables by technical analysis. On the other hand, fundamentalists have used some fundamental values of a firm like sales or earnings and their ratios. In 1960s, the structural models like the capital asset allocation model appeared from academia to predict the return of stocks (Sharpe, 1964). According to the well-known efficient market hypothesis it is argued that the stock price is fully random walk without new unpredictable information, making it almost impossible to predict it. There are, however, several counter-evidences that the stock price process does not follow the random walk leaving aside some controversial issues. Two typical such counter-evidences are the momentum effect and the mean reversion which show that the autocorrelations of the return of a stock are positive in short horizons and negative for long horizons.

Inspired by these empirical findings, during the last decades many statistical learning techniques have been applied to predict various financial variables including stock price in many financial markets (Chang, Wang, & Zhou, 2011; Chen, Shih, & Wu, 2006; Gestel et al., 2001; Kara, Boyacioglu, & Baykan, 2011; Kim, 2003; Refenes, Zapranis, & Francis, 1995; Shazly & Shazly, 1999; Steiner & Wittkemper, 1995; Tay & Cao, 2001; Tsibouris & Zeidenberg, 1995; Wittkemper & Steiner, 1996; Zhang & Wu, 1996). Korean stock index was also investigated by several past studies. Chen et al. (2006) compared the performance of support vector machines and back propagation neural networks in forecasting the six major Asian stock markets and got about 0.55 with support vector machines and 0.56 with artificial neural networks in directional accuracy for Korea composite stock price index (KOSPI). Kim (2003) adopted support vector machines to KOSPI index in forecasting Korean stock markets and reported 0.57 in directional accuracy with support vector machines.

There were also many empirical results that the derivative markets led the cash (or spot) markets. Fleming, Ostdiek, and Whaley (1996) found the returns in cost efficient index derivative markets led those in stock markets. For S&P 500 index, Boyle, Byoun, and Park (2002) found the options market led the cash index. For Korean markets, Kang, Lee, and Lee (2006) analyzed Korean KOSPI 200 index and found that both the futures and options markets led the cash index up to ten minutes. With these results, we can notice that the derivative markets lead the cash index. This suggests the use of independent variables from not only the spot index time series data but also the time series of futures, options, and other financial variables to increase the prediction accuracy of spot index or return.

However, the learning approaches above were mostly based on low- or medium-frequency, weekly or daily data. Two main questions to be addressed when we apply statistical learning techniques to high-frequency data are whether the stock price

* Corresponding author.
E-mail address: jaewook@snu.ac.kr (J. Lee).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.04.015

change is better predictable with high-frequency time series data and whether the futures index or foreign exchange rate data can help the prediction of the spot index in high-frequency time window. To address these, we conduct a comprehensive study of the applications of statistical learning techniques to predict the trend of the return of high-frequency KOSPI 200 index data using the information from the one-minute time series of spot index, futures index, and foreign exchange rate. For the statistical learning techniques, we used four state-of-the-art classifiers: linear regression, logistic regression, artificial neural networks, and support vector machines. The independent variables were extracted from the time series of KOSPI 200 index, KOSPI 200 futures index and the exchange rate of Korean Won and US Dollars. These results help to explain *the tail wagging the dog* phenomenon.

The organization of this paper is as follows. In the next section, we introduce previous works on employed statistical learning techniques. In Section 3, we describe the data and experimental procedure with the way of making independent variables from time series that we used. Then, in Section 4, we explain the experimental results by comparing the classifiers with and without dimension reduction and test the simulated returns of them applied to a virtual trading strategy. Finally, in Section 5, we report conclusions and directions for future works.

2. Review of compared classifiers

In this section, we review the compared state-of-the-art classifiers: linear regression, logistic regression, artificial neural networks, and support vector machines. These classification algorithms are briefly explained below.

2.1. Linear regression

Linear regression, one of the simplest data fitting methods, aims at finding parameters β such that the line $y_{pred} = X\beta$ best fits the dataset where X is a dataset matrix (Du, Han, & Chen, 2004). It can be treated as a least square sense minimization problem as follows:

$$\hat{\beta} = \arg\min_{\beta} \|y_{true} - X\beta\|^2 \tag{1}$$

where $\hat{\beta}$ is a parameter vector that minimizes the difference between y_{true}, the true value of the target, and y_{pred}, the predicted value. The solution of Eq. (1) is then given by

$$\hat{\beta} = (X^T X)^{-1} X^T y_{true}. \tag{2}$$

For classification, there should be a target treating criterion unlike the case of regression since the target value is the class taking discrete values. In this paper, the value of y_{ture} is set to be a target vector of training dataset, consisting of two values, $\{1, -1\}$. After finding $\hat{\beta}$ from Eq. (2), we obtain y_{pred} of the test dataset, which is a vector consisting of continuous values. To get the final predicted target of the test dataset, we apply the following procedure as follows:

$$\text{target}_i = \begin{cases} 1 & \text{if } y_{pred,i} > 0, \\ -1 & \text{otherwise}. \end{cases} \tag{3}$$

2.2. Logistic regression

Logistic regression, similarly to the linear regression, aims at finding the best parameters β that solves the following minimization problem.

$$\hat{\beta} = \arg\min_{\beta} \sum_i \|y_{true,i} - f(x_i^T \beta)\|^2. \tag{4}$$

where x_i is the ith instance of the dataset and $f(\cdot)$ called the logistic function is a real function whose domain is the set of real numbers and range is $(0, 1)$. A widely used logistic function is given by

$$f(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}, \tag{5}$$

Since the logistic function takes the values in the range of $(0, 1)$ and is increasing, it is usually considered as a proxy for probability. For binary classification, it is considered as the probability of one class, say $\mathcal{C}_1$ (denoted by +1) and the probability of the other class, $\mathcal{C}_2$ as $1 - p(\mathcal{C}_1)$ (denoted by -1).

Since Eq. (4) is a nonlinear optimization problem, so there is no closed form solution for Eq. (4) and an iterative method, like *Newton–Rhapson method*, is widely used to find $\hat{\beta}$ (Bishop, 2006). The remaining procedures are almost the same with the linear regression case. With given parameter $\hat{\beta}$, we can find the probability that each test instance belongs to $\mathcal{C}_1$ by calculating $f\left(x_i^T \hat{\beta}\right)$, from which we classify the instances as follows:

$$\text{target}_i = \begin{cases} 1 & \text{if } f\left(x_i^T \hat{\beta}\right) > 0.5, \\ -1 & \text{otherwise}. \end{cases} \tag{6}$$

2.3. Artificial neural networks

Artificial neural networks (ANN) are another widely used classifier which is highly nonlinear. By mimicking a human brain, it consists of perceptrons, each of which has several input values and usually one output value. The output value is determined by a function of input values, called an activation function. The activation function can be a variety of functions such as step function, linear function, sigmoid function and so forth (Grossberg, 1982). In this paper, we used a hyperbolic tangent sigmoid function and a linear function as an activation function. The brief shape of a perceptron is shown on Fig. 1.

In an artificial neural network system, these perceptrons are linked with weighted edges as illustrated in Fig. 2. There are three kinds of layers which contain perceptrons. The input layer is the first layer whose perceptrons get the input values from the instances in a given dataset. The output layer has perceptrons, in many times only one perceptron, which make the overall outputs. Between input layer and output layer, there can be a number of hidden layers. Perceptrons on hidden layer get input values from the previous layer and give output values to the next layer. For supervised learning problems like classification, the weights of edges can be computed with training data using a back-propagation method (Hecht-Nielsen, 1989).

2.4. Support vector machines

Support vector machine (SVM) is a recently emerged nonlinear binary classifier (Boser, Guyon, & Vapnik, 1992; Vapnik, 1995) and successfully extended to regression and clustering problems (Jung,

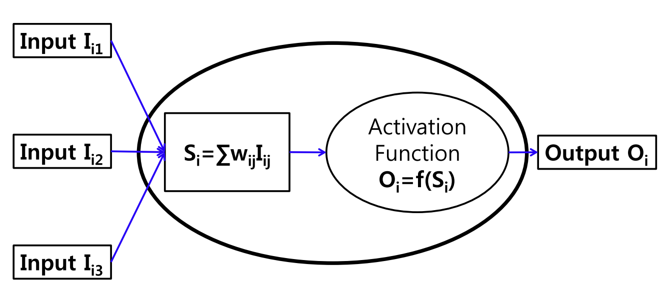

Fig. 1. The shape of a perceptron.

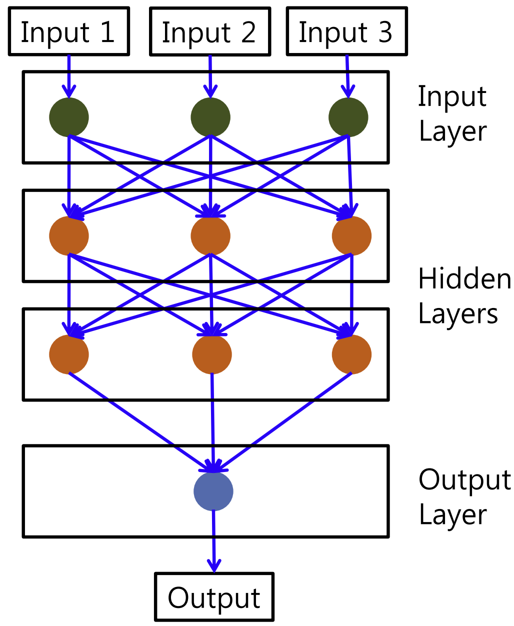

Fig. 2. The structure of an artificial neural network system.

Lee, & Lee, 2010; Jung, Kim, & Lee, 2011; Lee & Lee, 2005, 2007, 2007, 2010). It first utilizes a nonlinear transfer mapping Φ to map all training data into a high-dimensional feature space. Next, to find an optimal linear classifier of the form

$$f(x_i) = w^T\Phi(x_i) + b. \tag{7}$$

in the mapped high-dimensional feature space, it tries to find the parameters w and b which make the classifier in Eq. (7) optimal in the sense that the margin, the distance between the classifier and the nearest point $\Phi(x_i)$, is maximized. Finding the optimal w and b can be achieved by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{8}$$

subject to $y_i(w^T\Phi(x_i) + b) \geqslant 1$ where y_i is a binary target of the instance x_i. In the case of overlapping or misclassified training instances as in Fig. 3, we can add the penalty term for these misclassification and then the optimization problem in Eq. (8) can be changed into:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \tag{9}$$

subject to $y_i(w^T\Phi(x_i) + b) \geqslant 1 - \xi_i$ where ξ_i is a slack variable to allows soft margins. The solution of Eq. (9) is then given by

$$w = \sum_i \alpha_i^* y_i \Phi(x_i) \tag{10}$$

where α_i^* is the solution of the following quadratic optimization problem, which is dual of the primal problem (9):

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{11}$$

Here $k(x_i,x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ is called a kernel function. There are many candidates for the kernel functions and in this paper, the radial basis kernel function is used (Bishop, 2006).

The nearest points x_i that are nearest to the decision boundary are correspondent with $\alpha_i^* > 0$ and are called support vectors. There are two kinds of support vectors. Ones are support vectors for the class +1 and the others are for the class -1. After finding

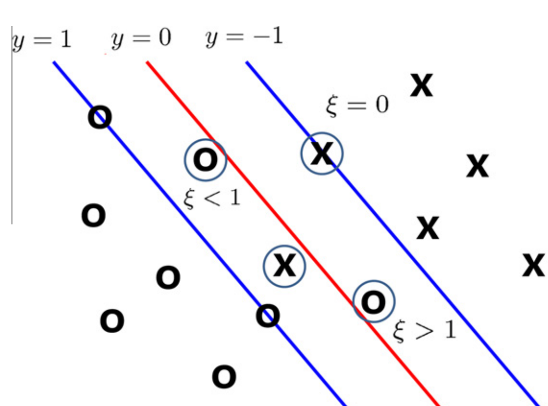

Fig. 3. Misclassification case of support vector machine classifier: points on the blue lines are support vectors and the red line is a decision hyperplane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Extracted independent variables and their description.

Variables	Description	Formula
RDP (-k)	Relative difference prices between k minutes before and now in percentage.	$\frac{p(t)-p(t-k)}{p(t=k)} \times 100$
MA (k)	Moving average of k minutes.	$\frac{\sum_{i=0}^{k-1} p(t-i)}{k}$
EMA (k)	Exponential moving of k minutes.	$\alpha \sum_{i=0}^{k} (1-\alpha)^i p(t-i)$ where $\alpha = \frac{2}{k+1}$
DISP (k)	k-minute disparity: the relative difference in percentage between k-minute moving average and the current price.	$\frac{p(t)-MA(k)}{MA(k)} \times 100$
EDISP (k)	k-minute disparity with exponential moving average: the relative difference in percentage between k-minute exponential moving average and the current price.	$\frac{p(t)-EMA(k)}{EMA(k)} \times 100$
OSCP	Price oscillator: the relative difference 5-min moving average and 10-min between moving average.	$\frac{MA(5)-MA(10)}{MA(5)}$
EOSCP	Price oscillator with exponential moving average: the difference between 5-min exponential moving average and 10-min exponential moving average.	$\frac{EMA(5)-EMA(10)}{EMA(5)}$
RSI (k)	Relative strength index for k-minutes.	$1 - \frac{1}{1+\sum_{i=0}^{k-1}\frac{u(t-i)}{k}\sum_{i=0}^{k-1}\frac{d(t-i)}{k}}$ where $u(t)$ is 1 if there is upward-price change at time t and is 0 otherwise, and $d(t)$ is 1 if there is downward-price change at time t and is 0 otherwise.

optimal w and b with training data, like the other classifiers, we can classify the test instances as follows:

$$\text{target}_i = \begin{cases} 1 & \text{if } f(x_i) > 0, \\ -1 & \text{otherwise}. \end{cases} \quad (12)$$

3. Data description and experimental procedure

3.1. Data description and attribute selection

In our empirical study, we used four weeks high-frequency data consisting of the KOSPI 200 index, KOSPI 200 futures price, and Korean Won/Us Dollar rates from 2011 March 7th to 2011 April 1st where the time interval between observations is one minute. The Korean stock market opens at 9:00 am and closes at 3:00 pm. However, the time interval from 2:50 pm to 3:00 pm is the synchronized bidding time period, so the price does not change. Also, we need 20 min past index price data to make independent variables and 5 min future data to make a target variable. Therefore, the actual datasets we use for experiments have the instances from 9:20 am to 2:45 pm. Hence, we have 1630 instances a week.

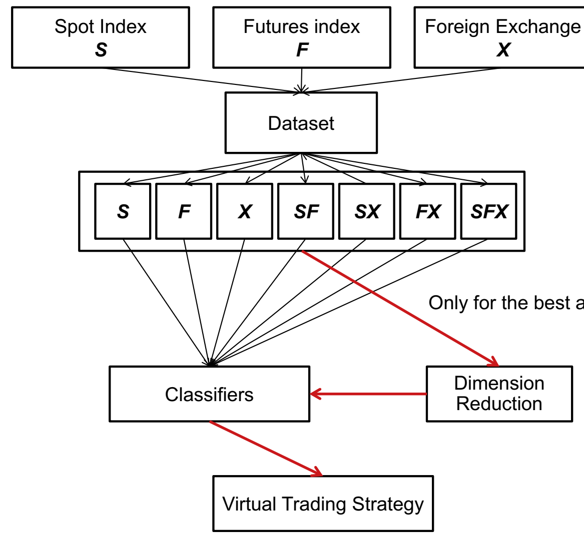

Fig. 5. Total procedure of the algorithm: red lines are only applied for the best independent variable set. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The independent variables, most of which are also employed in Chen et al. (2006) and Kim (2003), are 23 variables extracted from the time series of the spot index: RDP (-1), RDP (-2), RDP (-3), RDP (-5), RDP (-10), RDP (-20), MA (5), MA (10), MA (20), EMA (5), EMA (10), EMA (20), DISP (5), DISP (10), DISP (20), EDISP (5), EDISP (10), EDISP (20), OSCP, EOSCP, RSI (5), RSI (10), and RSI (20). The detailed explanation of these variables is shown in Table 1. The set of these 23 variables is named S (Spot) set. The number of variables from the futures time series is 24 with 23 for the same with the spot index and one for the original futures price itself. The set of these variables is named F (Futures) sets. The variables from the rate of Korean Won and US Dollars are the same with those from the futures and they are named X (foreign eXchange). The target variable is a binary variable that if RDP + 5 of the spot index is positive or zero, then +1, otherwise −1. Therefore, the instance at time T has the information from 20 min before to T as independent variables and uses the spot index value five minutes after T as dependent target variable.

We divide the total dataset into three overlapping datasets: each one has one week of training data and another one week of test data and it rolls over. Hence the first one has the training data from 2011 Mar 7th to 11th and the test data from 2011 Mar 14th to 18th. The second one has the training data from 2011 Mar 14th to 18th and the test data from 2011 Mar 21st to 25th. The last one has the training data from 2011 Mar 21st to 25th and the test data from 2011 Mar 28th to Apr 1st. The dataset separation is shown on Fig. 4.

3.2. Experimental procedure

In our experiments, we use four classifiers described in Section 2: linear regression, logistic regression, artificial neural networks,

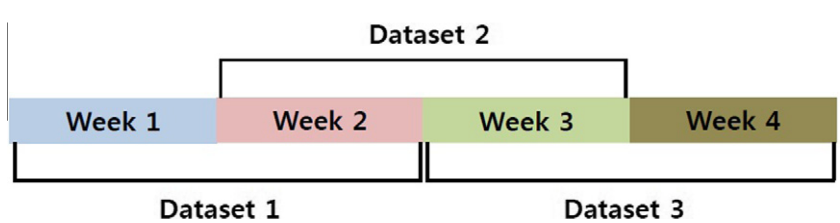

Fig. 4. The dataset separation with rolling over.

and support vector machines. These classifiers are applied to each of three datasets with seven kinds of independent variable attribute set: using S only, F only, X only, S and F, S and X, F and X, and all variables of S, F, and X. For linear regression, we decide that the predicted value is +1 if the regressed value is positive and −1 if the regressed value is negative. For logistic regression, we follow the usual process for binary classification. For artificial neural networks, we use as a training process a back propagation algorithm and used as an activation function a hyperbolic tangent sigmoid function except for the edges from the last hidden layer to the output layer where we use a simple linear function. For a fair comparison we considered all the case where the number of hidden layers is from 1 to 40, each of which has three perceptrons. For support vector machines, we used the Gaussian radial basis function kernel given by $k(x_i, x_j) = \exp(-q\|x_i - x_j\|^2)$ where q is a scaling factor. For a fair comparison we considered all the case where the scaling factor q of the kernel is from 0.01 to 1000.

These classifiers are applied with and without dimension reduction. For the dimension reduction process, we used a PCA (principal component analysis) algorithm with the variable set which shows the best performances. The reduced dimension is from 1 to 30 and the classifiers are applied to this dimension reduced data. Also, we compared the performance of a virtual trading strategy using the best result of each classifier.

Our total procedure are shown in Fig. 5. The red lines are only for the best independent variable set. To verify the performance of the classifiers, we compared the accuracy of them with the baseline accuracy; the accuracy of trivial classifier which selects the mode between two classes. The baseline accuracy of the first test data is 0.5288, of the second is 0.5184, and of the last is 0.5239.

4. Experimental results

In this section, we present our experimental results for which the four classifiers are applied to 3 datasets with 7 different set of attributes. We also compared the performance of these classifiers with and without dimension reduction. Finally we performed a virtual trading for each classifier to assess its profitability.

4.1. Results without dimension reduction

Firstly, the results of linear regression are shown in Table 2 where the best result of each dataset is boldfaced. It is observed that using S only attributes or X only attributes leads to usually worse accuracy than the others whereas the high prediction accuracy appears stably when S&F attributes are used. As is shown in the results of SX, FX, SFX, the exchange rate information is not that useful for our prediction task. In average, the best result was obtained from independent variable sets of S&F and the accuracy was 58.4%. It is about 6.3% better than the case when only independent variables of the set S were used.

Secondly, the results of logistic regression are shown in Table 3 where the best result of each dataset is also boldfaced. Similarly to linear regression case, the stably good performances were obtained when S&F attributes are used. Also in average, the prediction accuracy with S&F attributes is about 6.7% higher than that with only S attributes. In this case, the worse result was obtained when S&X attributes are used which has accuracy of 51.5%, not so different from baseline accuracy.

Thirdly, for artificial neural network classifier, we controlled the number of hidden layers as parameters from 1 to 40. The good results are mostly obtained from the small number of hidden layers because the large number of hidden layers can make overfitting problem and Table 4 shows the results of artificial neural network classifier where the number of hidden layers is from 1 to 4. The best result of each dataset and each number of hidden layers is boldfaced. As the number of hidden layers changes, the independent variables sets leading to the best result also changed a little bit (sometimes using only F set caused the best result and sometimes using all of S, F, and X sets did). However, in the view of the average of three datasets, using both S set and F set caused the better results stably. This result coincides with those of linear

Table 2
Prediction accuracy results of linear regression with the test datasets.

Features	S	F	X	SF	SX	FX	SFX
Dataset 1	0.523	**0.601**	0.530	0.592	0.510	0.583	0.567
Dataset 2	0.511	0.564	0.523	**0.587**	0.536	0.554	0.589
Dataset 3	0.528	0.539	0.511	**0.572**	0.510	0.521	0.561
Average	0.521	0.568	0.521	**0.584**	0.519	0.553	0.572

Table 3
Prediction accuracy results of logistic regression with the test datasets.

Features	S	F	X	SF	SX	FX	SFX
Dataset 1	0.523	**0.602**	0.529	0.591	0.502	0.546	0.580
Dataset 2	0.510	0.558	0.509	**0.591**	0.517	0.554	0.548
Dataset 3	0.525	0.532	0.534	**0.579**	0.527	0.519	0.492
Average	0.520	0.564	0.524	**0.587**	0.515	0.540	0.540

Table 4
Prediction accuracy results of artificial neural networks with the test datasets.

Features	S	F	X	SF	SX	FX	SFX
# of hidden layers	Dataset 1						
1	0.526	0.593	0.450	**0.594**	0.558	0.567	0.491
2	0.519	0.537	0.520	**0.559**	0.536	0.467	0.504
3	0.533	0.550	0.471	**0.618**	0.507	0.468	0.504
4	0.520	**0.599**	0.524	0.596	0.471	0.592	0.545
	Dataset 2						
1	0.509	0.543	0.503	0.588	0.518	0.535	**0.596**
2	0.519	0.561	0.504	0.590	0.552	0.523	**0.601**
3	0.493	0.574	0.491	**0.587**	0.507	0.563	0.558
4	0.542	0.516	0.511	**0.580**	0.512	0.571	0.574
	Dataset 3						
1	0.508	0.531	0.478	0.540	0.518	0.528	**0.564**
2	0.525	0.523	0.551	**0.585**	0.502	0.545	0.571
3	0.523	0.541	0.518	0.510	0.533	0.511	**0.559**
4	0.517	0.555	0.501	**0.571**	0.501	0.516	0.568
	Average						
1	0.514	0.556	0.477	**0.574**	0.531	0.543	0.550
2	0.521	0.540	0.525	**0.578**	0.530	0.512	0.558
3	0.516	0.555	0.493	**0.571**	0.516	0.514	0.545
4	0.526	0.557	0.512	**0.582**	0.495	0.560	0.563

Table 5
Prediction accuracy results of support vector machine with the test datasets.

Features	S	F	X	SF	SX	FX	SFX
Scaling factor	Dataset 1						
25	0.517	0.604	0.535	**0.625**	0.525	0.601	0.616
33	0.52	0.605	0.527	**0.629**	0.536	0.592	0.614
	Dataset 2						
25	0.518	0.560	0.525	0.593	0.533	0.563	**0.600**
33	0.517	0.554	0.516	0.586	0.526	0.562	**0.592**
	Dataset 3						
25	0.518	0.564	0.503	**0.616**	0.507	0.528	0.579
33	0.515	0.548	0.515	**0.610**	0.514	0.537	0.587
	Average						
25	0.517	0.576	0.521	**0.611**	0.522	0.564	0.598
33	0.517	0.569	0.519	**0.608**	0.526	0.564	0.598

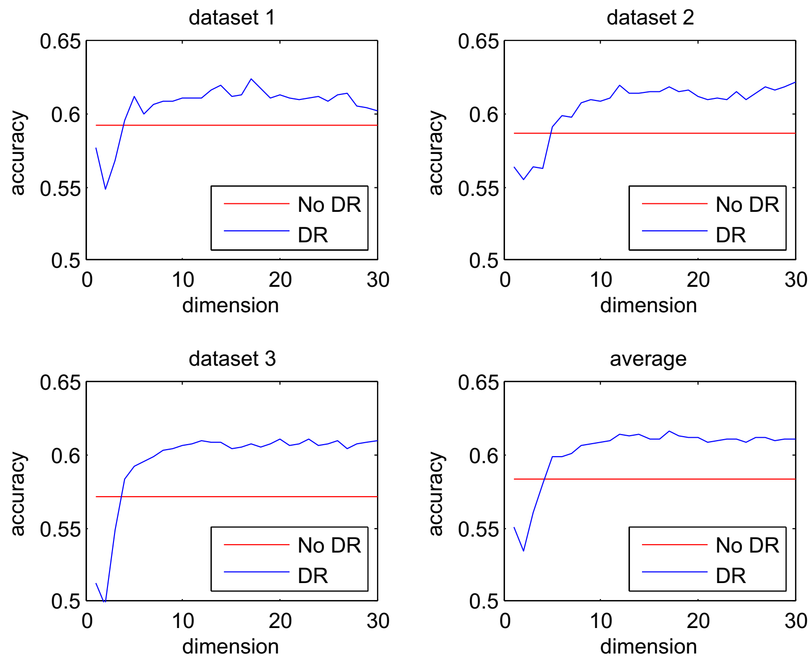

Fig. 6. Prediction accuracy of the linear regression after dimension reduction (DR).

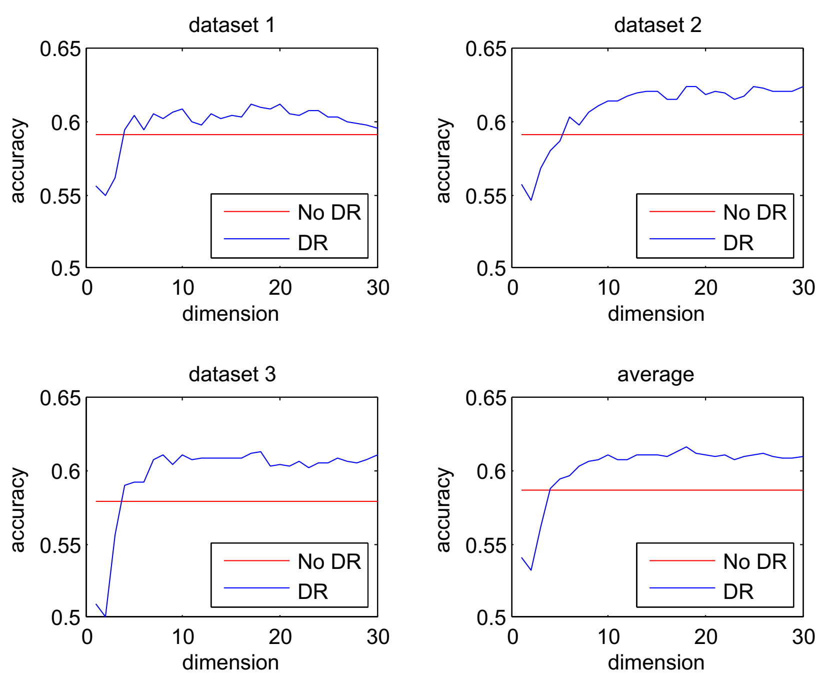

Fig. 7. Prediction accuracy of the logistic regression after dimension reduction (DR).

regression and logistic regression. The best results in average were obtained when the number of hidden layers is four and the independent variables were from the variable set S&F and the average accuracy at this condition is 58.2%.

Finally, for support vector machines, we controlled ten scaling factor values: 0.01, 0.1, 1, 10, 25, 33, 50, 75, 100, and 1000. Since the results when the scaling factors are 25 or 33 showed better performance than other scaling factors, we presented the results of the SVMs only with the scaling factors of 25, 33 in Table 5 where the best results are boldfaced in Table 5. The best results are obtained mostly when S&F attributes were used although in the case of a dataset 2 we observed slightly better results when all sets of S, F, and X were used. The best results anyway have in common that they all use the information of spot and futures price simultaneously. In average, the best results were obtained when the scaling factor of the radial basis kernel function was 25 and S&F attributes were used. This result coincides with the results of the other three classifiers. The best accuracy in average is 61.1%.

In summary, we observe that the stock price change is the best result of the spot index prediction when we use the independent variables from the time series of both the spot index and its futures and this result has much better accuracy than the prediction result

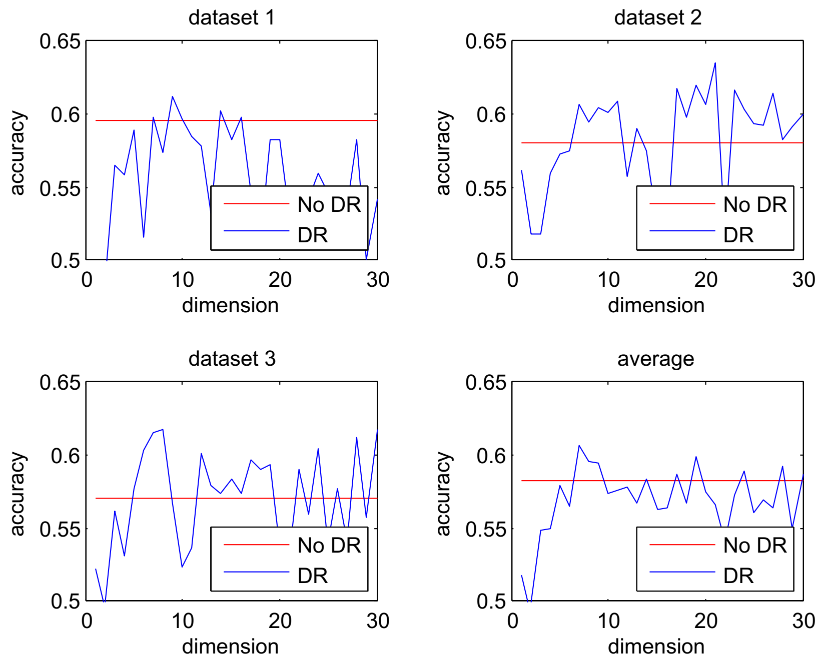

Fig. 8. Prediction accuracy of the ANN classifier after dimension reduction (DR).

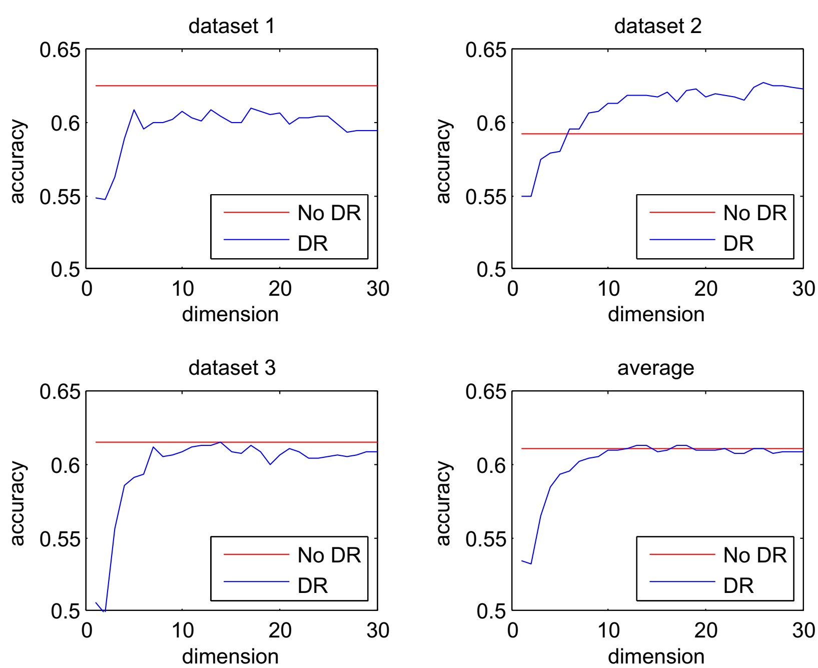

Fig. 9. Prediction accuracy of the SVM classifier after dimension reduction (DR).

Table 6
Prediction accuracy of four classifiers after dimension reduction (DR).

Classifier	Linear reg.	Logistic reg.	ANN	SVM
Dataset 1	0.624	0.612	0.612	0.611
Dataset 2	0.622	0.625	0.635	0.628
Dataset 3	0.611	0.613	0.618	0.616
Average	0.619	0.616	0.621	0.618

Table 7
Virtual trading returns (in percentage).

Classifier	Base	Linear reg.	Logistic reg.	ANN	SVM
Dataset 1	6.28	79.44	75.70	73.52	69.92
Dataset 2	3.54	31.80	31.93	35.47	33.67
Dataset 3	6.88	29.42	29.89	30.85	29.80
Average	5.57	46.88	45.84	46.61	44.46

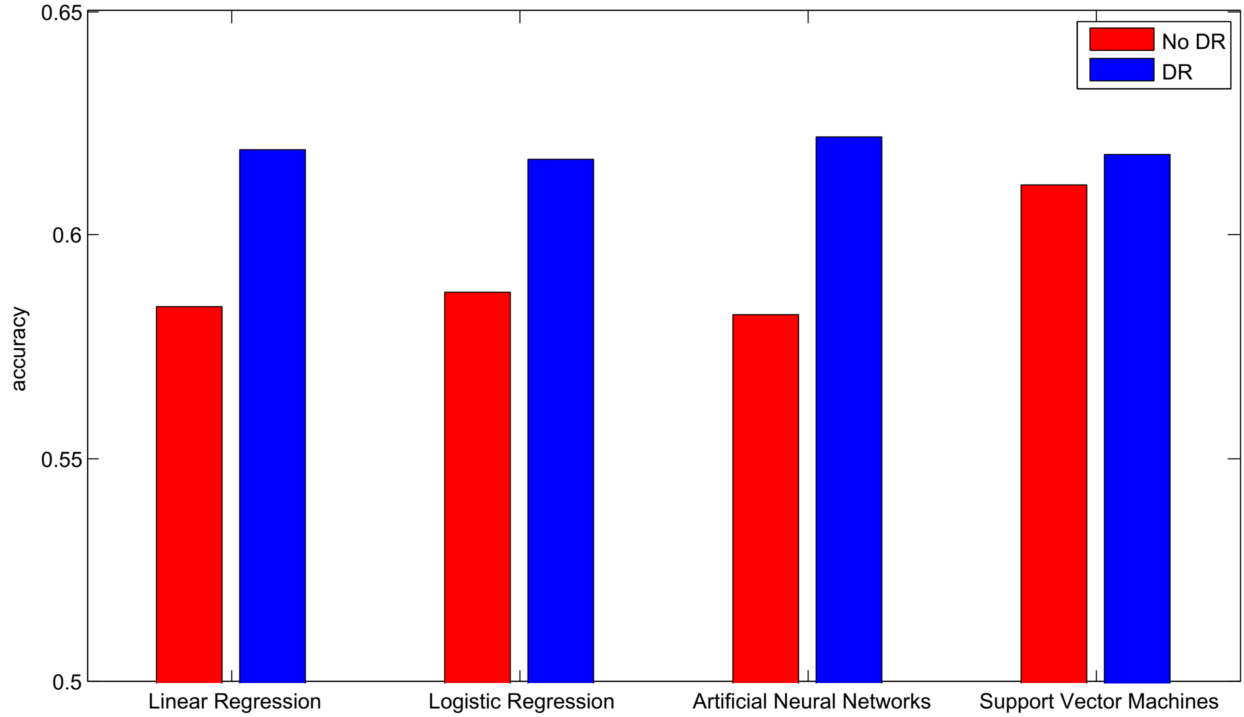

Fig. 10. Prediction accuracy of four classifiers without dimension reduction (No-DR) and with dimension reduction (DR).

with the spot index information only. This supports that KOSPI 200 index futures lead the cash index. Also the support vector machines showed quite higher prediction accuracy than the other three classifiers where there were no significant differences in accuracy among these three classifiers.

4.2. Results after dimension reduction

Since the variables come from the time series manipulation, they might be highly correlated. To overcome this we adopted the principal component analysis process to reduce the dimension of the independent variables of S&F. We reduced the dimension of data from 1 to 30 and selected one that showed the best performance for each classifier and each dataset. Firstly, Figs. 6 and 7 show the results of classification accuracy of the linear regression and the logistic regression for each reduced dimension from 1 to 30 both for the case of separated datasets and its average. The red line is classification accuracy without dimension reduction. It is observed that in both classifiers, accuracy can be noticeably improved with dimension reduction.

Secondly, Fig. 8 shows the results of classification accuracy of the artificial neural networks with the reduced dimensions. The number of hidden layer is taken as four, the value of which showed in average the best performance before the dimension reduction. It is also observed that accuracy can be improved with dimension reduction for all the datasets as well as the average, but unlike the two above classifiers, the improvement for dataset1 is not significant. Finally, Fig. 9 shows the results of classification accuracy of the SVM with the reduced dimensions. Unlike the other three classifiers, it is observed that there is no reduced dimension that shows better performance for dataset1 and dataset3 while reduced dimension significantly improved prediction accuracy for dataset2. Table 6 shows the best result of each of four classifiers. Most of the results with dimension reduction are better than those without dimension reduction, actually all exceed 60%, thereby improving the prediction accuracy. One noticeable point is that after the dimension reduction, there are no much differences among four classifiers. See Fig. 10.

In summary, there is a significant improvement in accuracy for each classifier when we preprocessed the dataset with dimension reduction techniques except the SVM in which case there is a slight improvement in accuracy. Also the spot index changes for high-frequency one-minute data are better predictable than for medium-frequency daily data reported in Chen et al. (2006) and Kim (2003).

4.3. Virtual trading

Finally, we tested the simulated returns of a virtual trading strategy utilizing the four classifiers that showed the best performance in average. The strategy is as follows: if the classifier decides that the target of the instance at T is 1, then buys at T and sell at $T+5$. If the classifier decides that the target is -1, then short-sells at T and buys back at $T+5$. The simulated virtual trading returns in percentage are shown in Table 7. The rather high returns in this result cannot be exactly realized since the transaction costs are not considered here. The base trading result is just a buy and hold strategy that chooses a better overall return of the two cases when a classifier decides all instances as one class, 1 or -1. It is observed that all the classifiers make much better results than the base classifier while there are no significant differences in the rate of returns among the classifiers.

5. Conclusion and future work

In this paper, we tried to predict the price change of KOSPI 200 index in five minutes in the form of binary classification. We used four state-of-the-art classifiers and observed that the support vector machines classifier overwhelmed the others when no dimension reduction is applied. We found that the use of spot index price and futures price information caused the best prediction accuracy result while the information of exchange rate did not, which implies that the futures information is very helpful in predicting the trends of the spot index, or the futures leads the cash. Also, dimension reduction for example with PCA algorithm helped to increase the accuracy of the classification. The achieved accuracy for the high-frequency one-minute data is shown to be better than that of medium-frequency daily data, suggesting better predictability of high-frequency data. After dimension reduction, the employed four classifiers did not show any noticeable different performances.

When a virtual trading strategy is applied to the high-frequency data using the trained classifiers, noticeable better profits are achieved than just a buy-and-hold-like strategy, although the transaction cost may pose some handicaps in directly applying these results to real trading. Thus the size of price change as well as some guides for handling transactions should be considered for the statistical learning approach to be applied in real trading and are left for the future works.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2010-0025530) and (No. 2011-0017657).

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual ACM workshop on computational learning theory* (pp. 144–152).

Boyle, P. P., Byoun, S., & Park, H. Y. (2002). The lead-lag relation between spot and option markets and implied volatility in option prices. *Research in Finance, 19*, 269–284.

Chang, P.-C., Wang, D.-D., & Zhou, C.-L. (2011). A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications, 39*, 611–620.

Chen, W.-H., Shih, J.-Y., & Wu, S. (2006). Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journal of Electronic Finance, 1*(1), 49–67.

Du, W., Han, Y. S., & Chen, S. (2004). Privacy-preserving multivariate statistical analysis: linear regression and classification. *Electrical Engineering and Computer Science, Paper*, 12.

Fleming, J., Ostdiek, B., & Whaley, R. E. (1996). Trading costs and the relative rates of price discovery in stock, futures, and option markets. *The Journal of Futures Markets, 16*(4), 353–387.

Gestel, T. V., Suykens, J. A. K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., et al. (2001). Financial time-series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks, 12*(4), 809–821.

Grossberg, S. (1982). *Studies of Mind and Brain*. Reidel.

Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. In *International joint conference on neural networks* (pp. 593–605).

Jung, K.-H., Kim, N., & Lee, J. (2011). Dynamic pattern denoising method using multi-basin system with kernels. *Pattern Recognition, 44*, 1698–1707.

Jung, K.-H., Lee, D., & Lee, J. (2010). Fast support-based clustering method for large-scale problems. *Pattern Recognition, 43*, 1975–1983.

Kang, J., Lee, C. J., & Lee, S. (2006). An empirical investigation of the lead-lag relations of returns and volatilities among the KOPSI200 spot, futures, and options markets and their explanations. *Journal of Emerging Market Finance, 5*(3), 235–261.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. 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.

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

Lee, J., & Lee, D. (2005). An improved cluster labeling method for support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*, 461–464.

Lee, D., & Lee, J. (2007). Domain described support vector classifier for multi-class classification problems. *Pattern Recognition, 40*, 41–51.

Lee, D., & Lee, J. (2007). Equilibrium-based support vector machine for semi-supervised classification. *IEEE Transactions on Neural Networks, 18*, 578–583.

Lee, D., & Lee, J. (2010). Dynamic dissimilarity measure for support-based clustering. *IEEE Transactions on Knowledge and Data Engineering, 22*, 900–905.

Refenes, A.-P., Zapranis, A. D., & Francis, G. (1995). Modeling stock returns in the frame work of APT: A comparative study with regression models. *Neural Networks in the Capital Markets*, 101–125.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance, 19*, 425–442.

Shazly, M. R. E., & Shazly, M. E. E. (1999). Forecasting currency prices using genetically evolved neural network architecture. *International Review of Financial Analysis, 8*(1), 67–82.

Steiner, M., & Wittkemper, H.-G. (1995). Neural networks as an alternative stock market model. *Neural Networks in the Capital Markets*, 135–147.

Tay, F. E. H., & Cao, L. (2001). Application of support vector machines in financial time-series forecasting. *Omega, 29*, 309–317.

Tsibouris, G., & Zeidenberg, M. (1995). Testing the efficient markets hypothesis with gradient descent algorithms. *Neural Networks in the Capital Markets*, 127–136.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

Wittkemper, H.-G., & Steiner, M. (1996). Using neural networks to forecast the systematic risk of stocks. *European Journal of Operational Research, 90*, 577–588.

Zhang, Y., & Wu, L. (1996). Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications, 36*, 8849–8854.