Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Neurocomputing 55 (2003) 307–319

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Financial time series forecasting using support vector machines

Kyoung-jae Kim*

Department of Information Systems, College of Business Administration, Dongguk University, 3-26, Pil-dong, Chung-gu, Seoul 100715, South Korea

Received 28 February 2002; accepted 13 March 2003

Abstract

Support vector machines (SVMs) are promising methods for the prediction of financial time-series because they use a risk function consisting of the empirical error and a regularized term which is derived from the structural risk minimization principle. This study applies SVM to predicting the stock price index. In addition, this study examines the feasibility of applying SVM in financial forecasting by comparing it with back-propagation neural networks and case-based reasoning. The experimental results show that SVM provides a promising alternative to stock market prediction.

Keywords: Support vector machines; Back-propagation neural networks; Case-based reasoning; Financial time series

1. Introduction

Stock market prediction is regarded as a challenging task of financial time-series prediction. There have been many studies using artificial neural networks (ANNs) in this area. A large number of successful applications have shown that ANN can be a very useful tool for time-series modeling and forecasting [24]. The early days of these studies focused on application of ANNs to stock market prediction (for instance [2,6,11,13,19,23]). Recent research tends to hybridize several artificial intelligence (AI) techniques (for instance [10,22]). Some researchers tend to include novel factors in the learning process. Kohara et al. [14] incorporated prior knowledge to improve the

* Tel: +82-2-2260-3324; fax: +82-2-2260-8824.
E-mail address: kkj@kgsm.kaist.ac.kr (K.-j. Kim).

0925-2312/03/$ - see front matter
doi:10.1016/S0925-2312(03)00372-2

performance of stock market prediction. Tsaih et al. [20] integrated the rule-based technique and ANN to predict the direction of the S& P 500 stock index futures on a daily basis.

Quah and Srinivasan [17] proposed an ANN stock selection system to select stocks that are top performers from the market and to avoid selecting under performers. They concluded that the portfolio of the proposed model outperformed the portfolios of the benchmark model in terms of compounded actual returns overtime. Kim and Han [12] proposed a genetic algorithms approach to feature discretization and the determination of connection weights for ANN to predict the stock price index. They suggested that their approach reduced the dimensionality of the feature space and enhanced the prediction performance.

Some of these studies, however, showed that ANN had some limitations in learning the patterns because stock market data has tremendous noise and complex dimensionality. ANN often exhibits inconsistent and unpredictable performance on noisy data. However, back-propagation (BP) neural network, the most popular neural network model, suffers from difficulty in selecting a large number of controlling parameters which include relevant input variables, hidden layer size, learning rate, momentum term.

Recently, a support vector machine (SVM), a novel neural network algorithm, was developed by Vapnik and his colleagues [21]. Many traditional neural network models had implemented the *empirical risk minimization principle*, SVM implements the *structural risk minimization principle*. The former seeks to minimize the mis-classification error or deviation from correct solution of the training data but the latter searches to minimize an upper bound of generalization error. In addition, the solution of SVM may be global optimum while other neural network models may tend to fall into a local optimal solution. Thus, overfitting is unlikely to occur with SVM.

This paper applies SVM to predicting stock price index. In addition, this paper examines the feasibility of applying SVM in financial forecasting by comparing it with ANN and case-based reasoning (CBR).

This paper consists of five sections. Section 2 introduces the basic concept of SVM and their applications in finance. Section 3 proposes a SVM approach to the prediction of stock price index. Section 4 describes research design and experiments. In Section 4, empirical results are summarized and discussed. Section 5 presents the conclusions and limitations of this study.

2. SVMs and their applications in finance

The following presents some basic concepts of SVM theory as described by prior research. A detailed explanation may be found in the references in this paper.

2.1. Basic concepts

SVM uses linear model to implement nonlinear class boundaries through some nonlinear mapping the input vectors x into the high-dimensional feature space. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. Thus, SVM is known as the algorithm that finds a special kind of linear model, the *maximum margin hyperplane*. The maximum margin hyperplane gives the maximum separation between the decision classes. The training examples that are closest to the maximum margin hyperplane are called *support vectors*. All other training examples are irrelevant for defining the binary class boundaries.

For the linearly separable case, a hyperplane separating the binary decision classes in the three-attribute case can be represented as the following equation:

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3, \tag{1}$$

where y is the outcome, x_i are the attribute values, and there are four weights w_i to be learned by the learning algorithm. In Eq. (1), the weights w_i are parameters that determine the hyperplane. The maximum margin hyperplane can be represented as the following equation in terms of the support vectors:

$$y = b + \sum \alpha_i y_i \mathbf{x}(i) \cdot \mathbf{x}, \tag{2}$$

where y_i is the class value of training example $\mathbf{x}(i)$, $\cdot$ represents the dot product. The vector $\mathbf{x}$ represents a test example and the vectors $\mathbf{x}(i)$ are the support vectors. In this equation, b and α_i are parameters that determine the hyperplane. From the implementation point of view, finding the support vectors and determining the parameters b and α_i are equivalent to solving a linearly constrained quadratic programming (QP).

As mentioned above, SVM constructs linear model to implement nonlinear class boundaries through the transforming the inputs into the high-dimensional feature space. For the nonlinearly separable case, a high-dimensional version of Eq. (2) is simply represented as follows:

$$y = b + \sum \alpha_i y_i K(\mathbf{x}(i), \mathbf{x}). \tag{3}$$

The function $K(\mathbf{x}(i), \mathbf{x})$ is defined as the kernel function. There are some different kernels for generating the inner products to construct machines with different types of nonlinear decision surfaces in the input space. Choosing among different kernels the model that minimizes the estimate, one chooses the best model. Common examples of the kernel function are the polynomial kernel $K(x, y) = (xy + 1)^d$ and the Gaussian radial basis function $K(x, y) = \exp(-1/\delta^2 (x - y)^2)$ where d is the degree of the polynomial kernel and δ^2 is the bandwidth of the Gaussian radial basis function kernel.

For the separable case, there is a lower bound 0 on the coefficient α_i in Eq. (3). For the non-separable case, SVM can be generalized by placing an upper bound C on the coefficients α_i in addition to the lower bound [22].

2.2. *Prior applications of SVM in financial time-series forecasting*

As mentioned above, the BP network has been widely used in the area of financial time series forecasting because of its broad applicability to many business problems and preeminent learning ability. However, the BP network has many disadvantages including the need for the determination of the value of controlling parameters and the number of processing elements in the layer, and the danger of overfitting problem.

On the other hand, there are no parameters to tune except the upper bound C for the non-separable cases in linear SVM [8]. In addition, overfitting is unlikely to occur with SVM. Overfitting may be caused by too much flexibility in the decision boundary. But, the maximum hyperplane is relatively stable and gives little flexibility [22].

Although SVM has the above advantages, there is few studies for the application of SVM in financial time-series forecasting. Mukherjee et al. [15] showed the applicability of SVM to time-series forecasting. Recently, Tay and Cao [18] examined the predictability of financial time-series including five time series data with SVMs. They showed that SVMs outperformed the BP networks on the criteria of normalized mean square error, mean absolute error, directional symmetry and weighted directional symmetry. They estimated the future value using the theory of SVM in regression approximation.

3. Research data and experiments

3.1. Research data

The research data used in this study is technical indicators and the direction of change in the daily Korea composite stock price index (KOSPI). Since we attempt to forecast the direction of daily price change in the stock price index, technical indicators are used as input variables. This study selects 12 technical indicators to make up the initial attributes, as determined by the review of domain experts and prior research [12]. The descriptions of initially selected attributes are presented in Table 1.

Table 2 presents the summary statistics for each attribute.

This study is to predict the directions of daily change of the stock price index. They are categorized as "0" or "1" in the research data. "0" means that the next day's index is lower than today's index, and "1" means that the next day's index is higher than today's index. The total number of sample is 2928 trading days, from January 1989 to December 1998. About 20% of the data is used for holdout and 80% for training. The number of the training data is 2347 and that of the holdout data is 581. The holdout data is used to test results with the data that is not utilized to develop the model.

The original data are scaled into the range of $[-1.0, 1.0]$. The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs, then helps to reduce prediction errors.

The prediction performance P is evaluated using the following equation:

$$P = \frac{1}{m}\sum_{i=1}^{m} R_i \quad (i = 1, 2, \ldots, m) \tag{4}$$

where R_i the prediction result for the ith trading day is defined by

$$R_i = \begin{cases} 1 & \text{if } PO_i = AO_i, \\ 0 & \text{otherwise,} \end{cases}$$

Table 1
Initially selected features and their formulas

Feature name	Description	Formula	Refs.
%K	Stochastic %K. It compares where a security's price closed relative to its price range over a given time period.	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$, where LL_t and HH_t mean lowest low and highest high in the last t days, respectively.	[1]
%D	Stochastic %D. Moving average of %K.	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$	[1]
Slow %D	Stochastic slow %D. Moving average of %D.	$\frac{\sum_{i=0}^{n-1} \%D_{t-i}}{n}$	[9]
Momentum	It measures the amount that a security's price has changed over a given time span.	$C_t - C_{t-4}$	[3]
ROC	Price rate-of-change. It displays the difference between the current price and the price n days ago.	$\frac{C_t}{C_{t-n}} \times 100$	[16]
Williams' %R	Larry William's %R. It is a momentum indicator that measures overbought/oversold levels.	$\frac{H_n - C_t}{H_n - L_n} \times 100$	[1]
A/D Oscillator	Accumulation/distribution oscillator. It is a momentum indicator that associates changes in price.	$\frac{H_t - C_{t-1}}{H_t - L_t}$	[3]
Disparity5	5-day disparity. It means the distance of current price and the moving average of 5 days.	$\frac{C_t}{MA_5} \times 100$	[5]
Disparity10	10-day disparity.	$\frac{C_t}{MA_{10}} \times 100$	[5]
OSCP	Price oscillator. It displays the difference between two moving averages of a security's price.	$\frac{MA_5 - MA_{10}}{MA_5}$	[1]
CCI	Commodity channel index. It measures the variation of a security's price from its statistical mean.	$\frac{(M_t - SM_t)}{(0.015\, D_t)}$ where $M_t = (H_t + L_t + C_t)/3$, $SM_t = \frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$, and $D_t = \frac{\sum_{i=1}^{n} \lvert M_{t-i+1} - SM_t \rvert}{n}$.	[1,3]
RSI	Relative strength index. It is a price following an oscillator that ranges from 0 to 100.	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{t-i}/n)/(\sum_{i=0}^{n-1} Dw_{t-i}/n)}$ where Up_t means upward-price-change and Dw_t means downward-price-change at time t.	[1]

C_t is the closing price at time t, L_t the low price at time t, H_t the high price at time t and, MA_t the moving average of t days.

Table 2
Summary statistics

Feature name	Max	Min	Mean	Standard deviation
%K	100.007	0.000	45.407	33.637
%D	100.000	0.000	45.409	28.518
Slow %D	99.370	0.423	45.397	26.505
Momentum	102.900	−108.780	−0.458	21.317
ROC	119.337	81.992	99.994	3.449
Williams' %R	100.000	−0.107	54.593	33.637
A/D Oscillator	3.730	−0.157	0.447	0.334
Disparity5	110.003	90.077	99.974	1.866
Disparity10	115.682	87.959	99.949	2.682
OSCP	5.975	−7.461	−0.052	1.330
CCI	226.273	−221.448	−5.945	80.731
RSI	100.000	0.000	47.598	29.531

PO_i is the predicted output from the model for the ith trading day, and AO_i is the actual output for the ith trading day, m is the number of the test examples.

3.2. SVM

In this study, the polynomial kernel and the Gaussian radial basis function are used as the kernel function of SVM. Tay and Cao [18] showed that the upper bound C and the kernel parameter δ^2 play an important role in the performance of SVMs. Improper selection of these two parameters can cause the overfitting or the underfitting problems. Since there is few general guidance to determine the parameters of SVM, this study varies the parameters to select optimal values for the best prediction performance. This study uses LIBSVM software system [4] to perform experiments.

3.3. BP

In this study, standard three-layer BP networks and CBR are used as benchmarks. This study varies the number of nodes in the hidden layer and stopping criteria for training. In this study, 6, 12, 24 hidden nodes for each stopping criteria because the BP network does not have a general rule for determining the optimal number of hidden nodes. For the stopping criteria of BP, this study allows 50, 100, 200 learning epochs per one training example since there is little general knowledge for selecting the number of epochs. Thus, this study uses 146 400, 292 800, 565 600 learning epochs for the stopping criteria of BP because this study uses 2928 examples. The learning rate is 0.1 and the momentum term is 0.1. The hidden nodes use the sigmoid transfer function and the output node uses the linear transfer function. This study allows 12 input nodes because 12 input variables are employed.

3.4. CBR

For CBR, the nearest-neighbor method is used to retrieve relevant cases. This method is a popular retrieval method because it can be easily applied to numeric data such as financial data. This study varies the number of nearest neighbor from 1 to 5. An evaluation function of the nearest-neighbor method is Euclidean distance and the function is represented as follows:

$$D_{\mathrm{IR}} = \sqrt{\sum_{i=1}^{n} w_i (f_i^{\mathrm{I}} - f_i^{\mathrm{R}})^2}, \tag{5}$$

where D_{IR} is a distance between f_i^{I} and f_i^{R}, f_i^{I} and f_i^{R} are the values for attribute f_i in the input and retrieved cases, n is the number of attributes, and w_i is the importance weighting of the attribute f_i.

4. Experimental results

One of the advantages of linear SVM is that there is no parameter to tune except the constant C. But the upper bound C on the coefficient α_i affects prediction performance for the cases where the training data is not separable by a linear SVM [8]. For the nonlinear SVM, there is an additional parameter, the kernel parameter, to tune. First, this study uses two kernel functions including the Gaussian radial basis function and the polynomial function. The polynomial function, however, takes a longer time in the training of SVM and provides worse results than the Gaussian radial basis function in preliminary test. Thus, this study uses the Gaussian radial basis function as the kernel function of SVMs.

This study compares the prediction performance with respect to various kernel parameters and constants. According to Tay and Cao [18], an appropriate range for δ^2 was between 1 and 100. In addition, they proposed that an appropriate range for C was between 10 and 100. Table 3 presents the prediction performance of SVMs with various parameters.

In Table 3, the best prediction performance of the holdout data is recorded when δ^2 is 25 and C is 78. The range of the prediction performance is between 50.0861% and 57.8313%. Fig. 1 gives the results of SVMs with various C where δ^2 is fixed at 25.

Tay and Cao [18] suggested that too small a value for C caused under-fit the training data while too large a value of C caused over-fit the training data. It can be observed that the prediction performance on the training data increases with C in this study. The prediction performance on the holdout data increases when C increases from 1 to 78 but decreases when C is 100. The results partly support the conclusions of Tay and Cao [18].

Fig. 2 presents the results of SVMs with various δ^2 where C is chosen as 78.

According to Tay and Cao [18], a small value of δ^2 would over-fit the training data while a large value of δ^2 would under-fit the training data. The prediction performance

Table 3
The prediction performance of various parameters in SVMs

C	Training data		Holdout data	
	Number of hit/total number	Hit ratio	Number of hit/total number	Hit ratio
(a) $\delta^2 = 1$				
1	1358/1637	82.9566	305/581	52.4957
10	1611/1637	98.4117	296/581	50.9466
33	1634/1637	99.8167	291/581	50.0861
55	1637/1637	100	295/581	50.7745
78	1637/1637	100	293/581	50.4303
100	1637/1637	100	293/581	50.4303
(b) $\delta^2 = 25$				
1	966/1637	59.0104	319/581	54.9053
10	1007/1637	61.515	331/581	56.9707
33	1037/1637	63.3476	330/581	56.7986
55	1048/1637	64.0195	334/581	57.4871
78	1060/1637	64.7526	336/581	57.8313
100	1076/1637	65.73	332/581	57.1429
(c) $\delta^2 = 50$				
1	954/1637	58.2773	331/581	56.9707
10	970/1637	59.2547	325/581	55.938
33	993/1637	60.6597	335/581	57.6592
55	1011/1637	61.7593	324/581	55.7659
78	1017/1637	62.1258	322/581	55.4217
100	1020/1637	62.3091	326/581	56.1102
(d) $\delta^2 = 75$				
1	951/1637	58.0941	323/581	55.5938
10	973/1637	59.438	323/581	55.5938
33	975/1637	59.5602	325/581	55.938
55	982/1637	59.9878	331/581	56.9707
78	985/1637	60.171	333/581	57.315
100	990/1637	60.4765	333/581	57.315
(e) $\delta^2 = 100$				
1	938/1637	57.2999	320/581	55.0775
10	962/1637	58.766	317/581	54.5611
33	971/1637	59.3158	322/581	55.4217
55	974/1637	59.4991	324/581	55.7659
78	984/1637	60.11	325/581	55.938
100	980/1637	59.8656	329/581	56.6265

on the training data decreases with δ^2 in this study. But Fig. 2 shows the prediction performance on the holdout data is stable and insensitive in the range of δ^2 from 25 to 100. These results also support the conclusions of Tay and Cao [18].

Figs. 3 and 4 present the results of the best SVM model for the training and the holdout data, respectively.

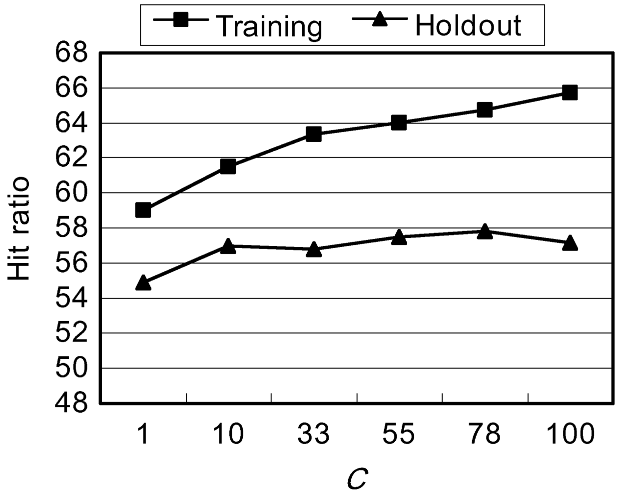


Fig. 1. The results of SVMs with various C where δ^2 is fixed at 25.

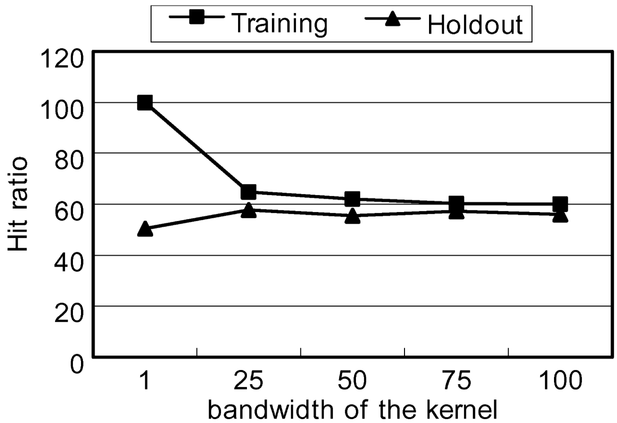
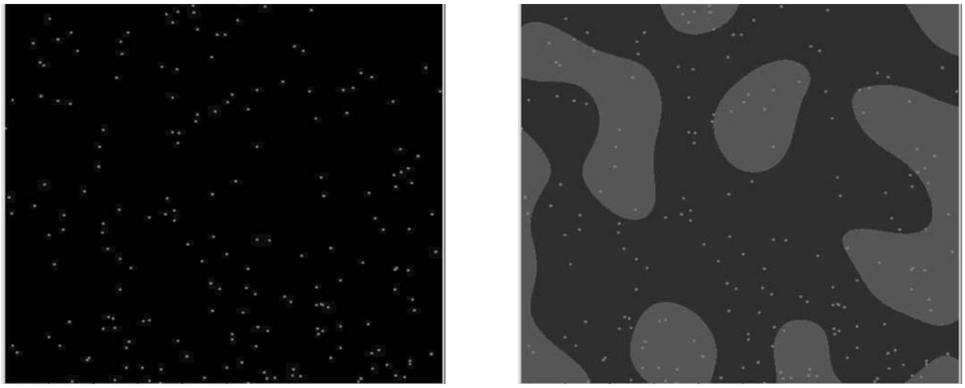


Fig. 2. The results of SVMs with various δ^2 where C is fixed at 78.

Figs. 3(a) and 4(a) represent data patterns before SVM is employed. Two different colors of circles are two classes of the training and the holdout examples. Figs. 3(b) and 4(b) show the results after SVM is implemented. The two classes are represented by green and red bullets.

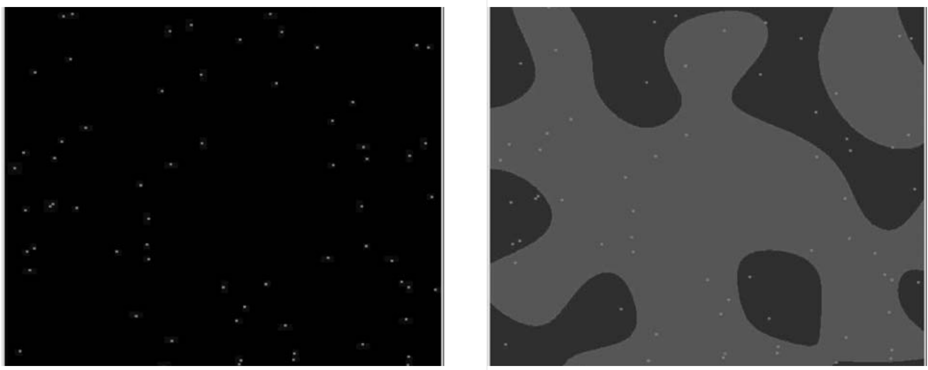
In addition, this study compares the best SVM model with BP and CBR. Table 4 gives the prediction performance of various BP models.

In Table 4, the best prediction performance for the holdout data is produced when the number of hidden processing elements are 24 and the stopping criteria is 146 400 or



(a) Before SVM is implemented (b) After SVM is implemented

Fig. 3. Graphical interpretation of the results of SVM for the training data: (a) before SVM is implemented and (b) after SVM is implemented.



(a) Before SVM is implemented (b) After SVM is implemented

Fig. 4. Graphical interpretation of the results of SVM for the holdout data: (a) before SVM is implemented and (b) after SVM is implemented.

292 800 learning epochs. The prediction performance of the holdout data is 54.7332% and that of the training data is 58.5217%.

For CBR, this study varies the number of retrieved cases for the new problem. The range of the number of retrieved cases is between 1 and 5. However, the prediction performances of these five experiments produce same results. Thus, this study uses the prediction performance when the number of retrieved cases is 1. The prediction accuracy of the holdout data is 51.9793%. For CBR, the performance of the training

Table 4
The results of various BP models

Stopping criteria (epoch)	Number of hidden nodes	Prediction performance for the training data (%)	Prediction performance for the holdout data (%)
146 400	6	58.1552	52.8399
	12	58.6439	53.3563
	24	58.5217	54.7332
292 800	6	58.1552	52.8399
	12	58.6439	53.3563
	24	58.5217	54.7332
565 600	6	58.1552	52.8399
	12	58.1552	52.8399
	24	58.1552	52.8399

Table 5
The best prediction performances of SVM, BP, and CBR (hit ratio: %)

	SVM	BP	CBR
Training data	64.7526	58.5217	
Holdout data	57.8313	54.7332	51.9793

Table 6
McNemar values (p values) for the pairwise comparison of performance

	BP	CBR
SVM	1642 (0.200)	4.654 (0.031)
BP		0.886 (0.347)

data is ignored because the retrieved case and the new case are the same in the training data. Table 5 compares the best prediction performances of SVM, BP, and CBR.

In Table 5, SVM outperforms BPN and CBR by 3.0981% and 5.852% for the holdout data, respectively. For the training data, SVM has higher prediction accuracy than BPN by 6.2309%. The results indicate the feasibility of SVM in financial time series forecasting and are compatible with the conclusions of Tay and Cao [18].

The McNemar tests are performed to examine whether SVM significantly outperforms the other two models. This test is a nonparametric test for two related samples and may be used with nominal data. The test is particularly useful with before-after measurement of the same subjects [7]. Table 6 shows the results of the McNemar test to compare the prediction performance of the holdout data.

As shown in Table 6, SVM performs better than CBR at 5% statistical significance level. However, SVM does not significantly outperform BP. In addition, Table 6 also shows that BP and CBR do not significantly outperform each other.

5. Conclusions

This study used SVM to predict future direction of stock price index. In this study, the effect of the value of the upper bound C and the kernel parameter δ^2 in SVM was investigated. The experimental result showed that the prediction performances of SVMs are sensitive to the value of these parameters. Thus, it is important to find the optimal value of the parameters.

In addition, this study compared SVM with BPN and CBR. The experimental results showed that SVM outperformed BPN and CBR. The results may be attributable to the fact that SVM implements the structural risk minimization principle and this leads to better generalization than conventional techniques. Finally, this study concluded that SVM provides a promising alternative for financial time-series forecasting.

There will be other research issues which enhance the prediction performance of SVM if they are investigated. The prediction performance may be increased if the optimum parameters of SVM are selected and this remains a very interesting topic for further study. The generalizability of SVMs also should be tested further by applying them to other time-series.

Acknowledgements

This work was supported by the Dongguk University Research Fund.

References

[1] S.B. Achelis, Technical Analysis from A to Z, Probus Publishing, Chicago, 1995.
[2] H. Ahmadi, Testability of the arbitrage pricing theory by neural networks, in: Proceedings of the International Conference on Neural Networks, San Diego, CA, 1990, pp. 385–393.
[3] J. Chang, Y. Jung, K. Yeon, J. Jun, D. Shin, H. Kim, Technical Indicators and Analysis Methods, Jinritamgu Publishing, Seoul, 1996.
[4] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2001, Available at http://www.csie.edu.tw/∼cjlin/papers/libsvm.pdf.
[5] J. Choi, Technical Indicators, Jinritamgu Publishing, Seoul, 1995.
[6] J.H. Choi, M.K. Lee, M.W. Rhee, Trading S& P 500 stock index futures using a neural network, in: Proceedings of the Annual International Conference on Artificial Intelligence Applications on Wall Street, New York, 1995, pp. 63–72.
[7] D.R. Cooper, C.W. Emory, Business Research Methods, Irwin, Chicago, 1995.
[8] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Networks 10 (5) (1999) 1048–1054.
[9] E. Gifford, Investor's Guide to Technical Analysis: Predicting Price Action in the Markets, Pitman Publishing, London, 1995.
[10] Y. Hiemstra, Modeling structured nonlinear knowledge to predict stock market returns, in: R.R. Trippi (Ed.), Chaos & Nonlinear Dynamics in the Financial Markets: Theory, Evidence and Applications, Irwin, Chicago, IL, 1995, pp. 163–175.
[11] K. Kamijo, T. Tanigawa, Stock price pattern recognition: a recurrent neural network approach, in: Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, 1990, pp. 215–221.

[12] K. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Syst. Appl. 19 (2) (2000) 125–132.
[13] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka, Stock market prediction system with modular neural network, in: Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, 1990, pp. 1–6.
[14] K. Kohara, T. Ishikawa, Y. Fukuhara, Y. Nakamura, Stock price prediction using prior knowledge and neural networks, Int. J. Intell. Syst. Accounting Finance Manage. 6 (1) (1997) 11–22.
[15] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Amelia Island, FL, 1997, pp. 511–520.
[16] J.J. Murphy, Technical Analysis of the Futures Markets: A Comprehensive Guide to Trading Methods and Applications, Prentice-Hall, New York, 1986.
[17] T.-S. Quah, B. Srinivasan, Improving returns on stock investment through neural network selection, Expert Syst. Appl. 17 (1999) 295–301.
[18] F.E.H. Tay, L. Cao, Application of support vector machines in financial time series forecasting, Omega 29 (2001) 309–317.
[19] R.R. Trippi, D. DeSieno, Trading equity index futures with a neural network, J. Portfolio Manage. 19 (1992) 27–33.
[20] R. Tsaih, Y. Hsu, C.C. Lai, Forecasting S& P 500 stock index futures with a hybrid AI system, Decision Support Syst. 23 (2) (1998) 161–174.
[21] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[22] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, San Francisco, CA, 2000.
[23] Y. Yoon, G. Swales, Predicting stock price performance: a neural network approach, in: Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Hawaii, 1991, pp. 156–162.
[24] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, Int. J. Forecasting 14 (1998) 35–62.

Kyoung-jae Kim received his M.S. and Ph.D. degrees in Management Information Systems from the Graduate School of Management at the Korea Advanced Institute of Science and Technology and his B.A. degree from the Chung-Ang University. He is currently a faculty member of the Department of Information Systems at the Dongguk University. His research interests include data mining, knowledge management, and intelligent agents.