ELSEVIER

Neurocomputing 48 (2002) 847–861

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Modified support vector machines in financial time series forecasting

Francis E.H. Tay*, L.J. Cao

Department of Mechanical & Production Engineering, National University of Singapore, 10 Kent Ridge Crescent, 119260 Singapore, Singapore

Received 5 June 2000; revised 13 August 2001

Abstract

This paper proposes a modified version of support vector machines, called C-ascending support vector machine, to model non-stationary financial time series. The C-ascending support vector machines are obtained by a simple modification of the regularized risk function in support vector machines, whereby the recent ε-insensitive errors are penalized more heavily than the distant ε-insensitive errors. This procedure is based on the prior knowledge that in the non-stationary financial time series the dependency between input variables and output variable gradually changes over the time, specifically, the recent past data could provide more important information than the distant past data. In the experiment, C-ascending support vector machines are tested using three real futures collected from the Chicago Mercantile Market. It is shown that the C-ascending support vector machines with the actually ordered sample data consistently forecast better than the standard support vector machines, with the worst performance when the reversely ordered sample data are used. Furthermore, the C-ascending support vector machines use fewer support vectors than those of the standard support vector machines, resulting in a sparser representation of solution.

Keywords: Non-stationary financial time series; Support vector machines; Regularized risk function; Structural risk minimization principle

1. Introduction

The financial market is a complex, evolutionary, and nonlinear dynamical system [3]. The financial time series are inherently noisy, non-stationary, and

* Corresponding author. Tel.: +65-874-6818; fax: +65-779-1459.
E-mail address: mpetayeh@nus.edu.sg (F.E.H. Tay).

0925-2312/02/$ - see front matter
PII: S0925-2312(01)00676-2

deterministically chaotic [23]. This means that the distribution of financial time series is changing over the time. Not only is a single data series non-stationary in the sense of the mean and variance of the series, but the relationship of the data series to other related data series may also be changing. Modeling such dynamical and non-stationary time series is expected to be a challenging task. Over the past few years, neural networks have been successfully used for modeling financial time series ranging from options price [5], corporate bond rating [9], stock index trading [8] to currency exchange [24]. Neural networks are universal function approximators that can map any nonlinear function without a priori assumption about the data [4]. Unlike traditional statistical models, neural networks are data-driven, non-parametric weak models, and they let "the data speak for themselves". So neural networks are less susceptible to the model mis-specification problem than most of the parametric models, and they are more powerful in describing the dynamics of financial time series than traditional statistical models [1,7,24].

Recently, a novel neural network technique, called support vector machines (SVMs), was proposed by Vapnik and his co-workers in 1995 [21]. The SVM is a new way to train polynomial neural networks or radial basis function neural networks based on the structural risk minimization (SRM) principle which seeks to minimize an upper bound of the generalization error rather than minimize the empirical error implemented in other neural networks. This induction principle is based on the fact that the generalization error is bounded by the sum of the empirical error and a confidence interval term that depends on the Vapnik–Chervonenkis (VC) dimension. Established on this principle, SVMs will achieve an optimum network structure by striking the right balance between the empirical error and the VC-confidence interval, eventually resulting in better generalization performance than other neural networks. Another merit of SVMs is that training SVMs is a uniquely solvable quadratic optimization problem, and the complexity of the solution in SVMs depends on the complexity of the desired solution, rather than on the dimensionality of the input space. Originally, SVMs have been developed for pattern recognition problems [6,16,17]. Recently, with the introduction of Vapnik's ε-insensitive loss function, SVMs have been extended to solve nonlinear regression estimation problems, and they exhibit excellent performance [10–12,22].

In the field of financial time series forecasting, numerous studies show that the relationship between input variables and output variable gradually changes over time, and recent data could provide more information than distant data. Therefore, it is advantageous to give more weights on the information provided by the recent data than that of the distant data based on this prior knowledge [2]. In the light of this characteristic, an innovative approach is proposed by Refenes and Bentz [13] which used the discounted least squares (DLS) in the back-propagation neural network to model non-stationary time series. The DLS is obtained by a simple modification of the commonly used least square function whereby the recent errors are penalized more heavily than the distant errors. As it is independent of the actual order of pattern presentation in the training procedure, the DLS is simple to implement. The DLS is reported to be very effective in both a controlled simulation experiment and the estimating of stock returns.

The present study is motivated by the DLS work, and generalizes the idea for SVMs whereby more weights are given to the recent ε-insensitive errors than the distant ε-insensitive errors in the regularized risk function. The regularized term in the regularized risk function is retained, regardless of the empirical error. The objective of this paper is to investigate whether this prior knowledge can also be exploited by SVMs in modeling the non-stationary financial time series. This is important as the SVMs are based on the unique SRM principle which consists of minimizing the sum of the empirical error and the regularized term.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the theory of SVMs for regression estimation. In Section 3, the C-ascending SVMs are described. Section 4 gives the experimental results as well as the data preprocessing technique. Section 5 concludes the work done.

2. Theory of SVMs for regression approximation

For the case of regression approximation, suppose there are a given set of data points $G=\{(x_i,d_i)\}_i^n$ (x_i is the input vector, d_i is the desired value, and n is the total number of data patterns) drawn independently and identically from an unknown function, SVMs approximate the function with three distinct characteristics: (i) SVMs estimate the regression in a set of linear functions, (ii) SVMs define the regression estimation as the problem of risk minimization with respect to the ε-insensitive loss function, and (iii) SVMs minimize the risk based on the SRM principle whereby elements of the structure are defined by the inequality $\|w\|^2 \leqslant$ constant. The linear function is formulated in the high dimensional feature space, with the form of function (1).

$$y=f(x)=w\phi(x)+b, \tag{1}$$

where $\phi(x)$ is the high dimensional feature space, which is nonlinearly mapped from the input space x. Characteristics (ii) and (iii) are reflected in the minimization of the regularized risk function (2) of SVMs, by which the coefficients w and b are estimated. The goal of this risk function is to find a function that has at most ε deviation from the actual values in all the training data points, and at the same time is as flat as possible.

$$R_{\mathrm{SVMs}}(C)=C\frac{1}{n}\sum_{i=1}^{n}L_\varepsilon(d_i,y_i)+\frac{1}{2}\|w\|^2, \tag{2}$$

$$L_\varepsilon(d,y)=\begin{cases}|d-y|-\varepsilon, & |d-y|\geqslant\varepsilon,\\ 0 & \text{otherwise.}\end{cases} \tag{3}$$

The first term $C(1/n)\sum_{i=1}^{n}L_\varepsilon(d_i,y_i)$ is the so-called empirical error (risk), which is measured by the ε-insensitive loss function (3). This loss function provides the advantage of using sparse data points to represent the designed function (1). The second term $\frac{1}{2}\|w\|^2$, on the other hand, is called the regularized term. ε is called the tube size of SVMs, and C is the regularization constant determining the trade-off between the empirical error and the regularized term. They are both user-prescribed parameters and are selected empirically. Introduction of the positive slack variables ζ, ζ^* leads Eq. (2) to the following constrained function [21]:

$$\begin{aligned} &\text{Minimize} \quad R_{\mathrm{SVMs}}(w, \zeta^{(*)}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*) \\ &\text{Subject to} \\ &\qquad d_i - w\phi(x_i) - b_i \leqslant \varepsilon + \zeta_i, \\ &\qquad w\phi(x_i) + b_i - d_i \leqslant \varepsilon + \zeta_i^*, \\ &\qquad \zeta^{(*)} \geqslant 0, \end{aligned} \tag{4}$$

where i represents the data sequence, with $i = n$ being the most recent observation and $i = 1$ being the earliest observation. Finally, by introducing Lagrange multipliers and exploiting the optimality constraints, decision function (1) takes the following form:

$$f(x, a_i^{(*)}) = \sum_{i=1}^{n}(a_i - a_i^*)K(x, x_i) + b. \tag{5}$$

Lagrange multipliers and support vectors. In function (5), a_i, a_i^* are the introduced Lagrange multipliers. They satisfy the equality $a_i a_i^* = 0$, $a_i \geqslant 0$, $a_i^* \geqslant 0$, $i = 1, \ldots, n$, and are obtained by maximizing the dual form of function (4), which has the following form:

$$\begin{aligned} R(a_i^{(*)}) = &\sum_{i=1}^{n} d_i(a_i - a_i^*) - \varepsilon\sum_{i=1}^{n}(a_i + a_i^*) \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i - a_i^*)(a_j - a_j^*)K(x_i, x_j) \end{aligned} \tag{6}$$

with the following constraints:

$$\begin{aligned} &\sum_{i=1}^{n}(a_i - a_i^*) = 0, \\ &0 \leqslant a_i \leqslant C, \quad i = 1, 2, \ldots, n, \\ &0 \leqslant a_i^* \leqslant C, \quad i = 1, 2, \ldots, n. \end{aligned}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions of quadratic programming, only a number of coefficients $(a_i - a_i^*)$ will assume non-zero values, and the data points associated with them could be referred to as support vectors. They are the only elements of the data points that are used in determining the position of the decision function according to function (5). For comparison, the data points with $|a_i - a_i^*| = C$ are called error support vectors, because they are lying outside the boundary of the decision function, and the data points with $0 < |a_i - a_i^*| < C$ are referred to as non-error support vectors, as they exactly lie on the boundary of the decision function.

Kernel function. In (5), $K(x_i,x_j)$ is the kernel function. The value is equal to the inner product of two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$. That is, $K(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$. The elegance of using kernel function lies in the fact that one can deal with feature spaces of arbitrary dimensionality without having to compute the map $\phi(x)$ explicitly. Any function that satisfies Mercer's condition [21] can be used as the kernel function. Common examples of kernel function are the polynomial kernel $K(x,y)=(x.y+1)^d$ and the Gaussian kernel $K(x,y)=\exp(-(x-y)^2/\delta^2)$.

From the implementation point of view, training SVMs is equivalent to solving a linearly constrained quadratic programming (QP) problem with the number of variables equal to the number of training data points. The sequential minimal optimization (SMO) algorithm extended by Scholkopf and Smola [15,14] is very effective in training SVMs for solving the regression estimation problem.

3. *C*-Ascending support vector machines (*C*-ASVMs)

As shown in function (4), the empirical risk function has equal weight C to all the ε-insensitive errors between the predicted and actual values. The regularization constant C determines the trade-off between the empirical risk and the regularized term. Increasing the value of C, the relative importance of the empirical risk with respect to the regularized term grows. For illustration, the empirical risk function is expressed as

$$E_{\mathrm{SVMs}}=C\sum_{i=1}^{n}(\zeta_i+\zeta_i^*). \tag{7}$$

In C-ASVMs, instead of a constant value, the regularization constant C adopts a weight function:

$$E_{c-\mathrm{ASVMs}}=\sum_{i=1}^{n}C_i(\zeta_i+\zeta_i^*), \tag{8}$$

$$C_i=w(i)C, \tag{9}$$

where $w(i)$ is the weight function satisfying $w(i)>w(i-1)$, $i=2,\ldots,n$. As the weights will incline from the distant training data points to the rent training data points, C_i is called ascending regularization constant which will give more weights on the more recent training data points. This raises the following question: what kind of weight function $w(i)$ should be used? In our experiment, the linear weight function and an exponential weight function are investigated. They are described as below.

1. For the linear weight function,

$$w(i)=\frac{i}{n(n+1)/2}. \tag{10}$$

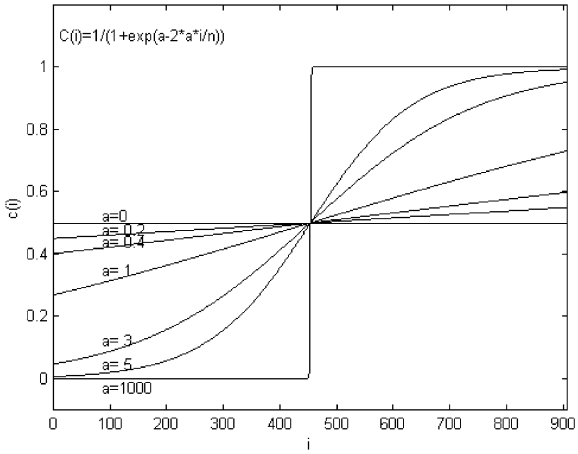


Fig. 1. Weights function of C-ASVMs. In the x-axis, i represents the training data sequence. (a) When $a = 0$, all the weights are equal to 0.5. (b) When $a = 1000$, the first half of the weights are equal to zero, and the second half of the weights are equal to 1. (c) When a increases, the first half of the weights become smaller while the second half of the weights become larger.

That is,

$$C_i = C\frac{i}{n(n+1)/2}. \tag{11}$$

2. For the exponential weight function,

$$w(i) = \frac{1}{1+\exp(a-2ai/n)}. \tag{12}$$

That is,

$$C_i = C\frac{1}{1+\exp(a-2ai/n)}. \tag{13}$$

The exponential weight function is adopted directly from that used in the DLS. a is the parameter to control the ascending rate. The behaviors of this weight function are illustrated in Fig. 1, which can be summarized as follows.

(i) When $a \to 0$ then $\lim_{a\to 0} C_i = \frac{1}{2}C$. In this case, there are the same weights in all the training data points, and $E_{C-\mathrm{ASVMs}} = \frac{1}{2}E_{\mathrm{SVMs}}$.

(ii) When $a \to \infty$, then

$$\lim_{a\to\infty} C_i = \begin{cases} 0, & i < \dfrac{n}{2}, \\ C, & i \geqslant \dfrac{n}{2}. \end{cases}$$

In this case, the weights for the first half of the training data points are reduced to zero, and the weights for the second half of the training data points are equal to 1, and

$$E_{C-\mathrm{ASVMs}} = \begin{cases} 0, & i < \dfrac{n}{2}, \\ E_{\mathrm{SVMs}}, & i \geqslant \dfrac{n}{2}. \end{cases}$$

(iii) $a \in [0, \infty]$ and increases, the weights for the first half of the training data points will become smaller, while the weights for the second half of the training data points will become larger.

Thus, the regularized risk function is calculated as follows:

$$\begin{aligned} &\text{Minimize} \quad R_{c-\mathrm{ASVMs}}(w, \zeta^{(*)}) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} C_i(\zeta_i + \zeta_i^*) \\ &\text{Subject to} \\ &\qquad d_i - w\phi(x_i) - b_i \leqslant \varepsilon + \zeta_i, \\ &\qquad w\phi(x_i) + b_i - d_i \leqslant \varepsilon + \zeta_i^*, \\ &\qquad \zeta^{(*)} \geqslant 0. \end{aligned} \tag{14}$$

The dual function has the original form,

$$R(a_i, a_i^*) = \sum_{i=1}^{n} d_i(a_i - a_i^*) - \varepsilon \sum_{i=1}^{n} (a_i + a_i^*) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} (a_i - a_i^*)(a_j - a_j^*)K(x_i, x_j),$$

but the constraints are changed as follows:

$$\begin{aligned} &\sum_{i=1}^{n} (a_i - a_i^*) = 0, \\ &0 \leqslant a_i \leqslant C_i, \quad i = 1, 2, \ldots, n, \\ &0 \leqslant a_i^* \leqslant C_i, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{15}$$

The SMO algorithm can still be used to optimize the C-ASVMs except that the upper bound value C_i for every training data points is different, and should be adapted according to function (15).

4. Experiment results

4.1. Data set

Three real futures contracts collected from the Chicago Mercantile are examined in the experiment. They are the Standard& Poor 500 stock index futures (CME-SP), United States 30-year government bond (CBOT-US), and German 10-year government bond (EUREX-BUND). Their corresponding time periods used are listed in Table 1, and the daily closing prices are used as the data sets.

Table 1
Three futures contracts

Names	Time periods
CME-SP	24/05/1989–08/10/1993
CBOT-US	23/05/1991–20/10/1995
EUREX-BUND	31/05/1991–25/10/1995

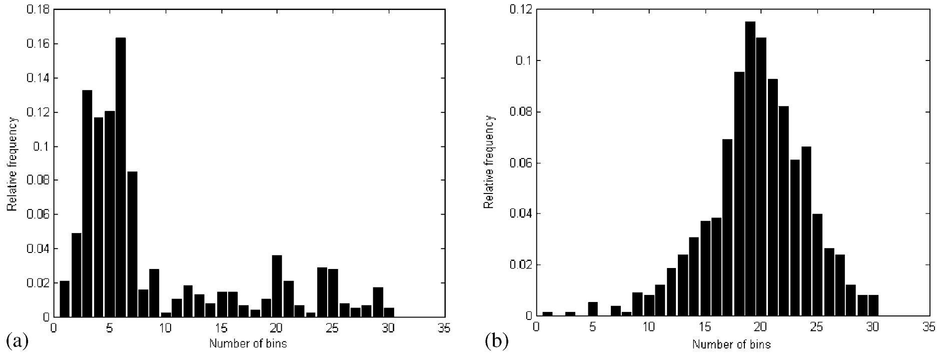


Fig. 2. Histogram. (a) CME-SP daily closing price and (b) RDP + 5. RDP + 5 values have a more symmetrical and normal distribution.

The original closing price is transformed into a 5-day relative difference in percentage of price (RDP). As interpreted by Thomason [18–20], there are four advantages in applying this transformation. The most prominent advantage is that the distribution of the transformed data will become more symmetrical and closer to normal as illustrated in Fig. 2. The modification in the trend of the data distribution will improve the predictive power of the neural network.

The input variables are constructed from four lagged RDP values based on 5-day periods (RDP-5, RDP-10, RDP-15, RDP-20), and one transformed closing price (EMA15) which is obtained by subtracting a 15-day exponential moving average from the closing price. The subtraction is performed to eliminate the trend in price. The output variable RDP + 5 is obtained by firstly smoothening the closing price with a 3-day exponential moving average. The calculations for all the indicators are given in Table 2.

The long left tail in Fig. 2b indicates that there are outliers in the data set. Since outliers may make it difficult or time-consuming to arrive at an effective solution for SVMs, RDP values beyond the limits of ± 2 standard deviations are selected as outliers. They are replaced with the closest marginal values. The other preprocessing technique is data scaling. All the data points are scaled into the range of $[-0.9, 0.9]$ as the data points include both positive and negative values. Finally, all of the three data sets are partitioned into three parts according to the time sequence. The first part is used for training, the second part used for validation is to select optimal parameters for the SVMs. The last part is used for the purpose of

Table 2
Input variables and output variable[a]

	Indicator	Calculation
Input variables	EMA15	$P(i) - \overline{EMA_{15}(i)}$
	RDP-5	$(p(i) - p(i-5))/p(i-5) * 100$
	RDP-10	$(p(i) - p(i-10))/p(i-10) * 100$
	RDP-15	$(p(i) - p(i-15))/p(i-15) * 100$
	RDP-20	$(p(i) - p(i-20))/p(i-20) * 100$
Output variable	RDP + 5	$(\overline{p(i+5)} - \overline{p(i)})/\overline{p(i)} * 100$ $\overline{p(i)} = \overline{EMA_3(i)}$

[a]*Note*: $EMA_n(i)$ is the n-day exponential moving average of the ith day; $p(i)$ is the closing price of the ith day.

testing. There are a total of 907 data patterns in the training set, 200 data patterns in both the validation set and the test set in each data set.

4.2. Experimental results

In this investigation, the Gaussian function is used as the kernel function of SVMs, which is inspired by the empirical findings that Gaussian kernels tend to give good performance under general smoothness assumptions, and therefore should be considered especially if no additional knowledge of the data is available [14]. As there is no structured way to choose the optimal parameters of SVMs, the values of the kernel parameter δ^2, C and ε that produce the best result on the validation set are used for the standard SVMs. The validation set is also used to choose the best combination of δ^2, C, ε, and the optimal control rate a in the C-ASVMs. These values could vary in futures due to different characteristics of futures. The SMO for solving the regression problem is implemented in this experiment, and the program is developed using VC^{++} language.

Furthermore, the C-ASVMs of the same settings with the reversed order of training data points which will put more weights on the more distant training data points (referred to as C-RSVMs) and the standard SVMs without using the most recent 200 training data points (referred to as SVMs-200) are also investigated. If the performance of the C-RSVMs and SVMs-200 is inferior to that of SVMs, the significance of the recent training data points will be made clearer.

The prediction performance is evaluated based on the criteria of the normalized mean squared error (NMSE) on the test set, which is calculated as follows:

$$NMSE = \frac{1}{n^* \delta^2} \sum_{i=1}^{n} (d_i - y_i)^2,$$

$$\delta^2 = \frac{1}{n-1} \sum_{i=1}^{n} (d_i - \bar{d})^2, \tag{16}$$

where $\bar{d}$ is the mean of the actual values.

Table 3
The converged NMSE in all SVMs

Methods	SVMs	C-ASVMs		C-RSVMs		SVMs-200
		Linear	Exponential	Linear	Exponential	
CME-SP	0.9353	0.8915	0.8850	0.9738	1.0198	0.9420
CBOT-US	1.0115	0.9978	0.9881	1.0423	1.0502	1.0375
EUREX-BUND	1.1114	1.0869	1.0694	1.1531	1.1867	1.1484

The best results in all types of SVMs are given in Table 3. From the table, it can be observed that the C-ASVMs have smaller values of NMSE than those of the standard SVMs. Also, the C-ASVMs with the use of the exponential weight function perform slightly better than those of using the linear weight function. Moreover, by paying little attention on the information provided by the recent training data points, both the C-RSVMs and SVMs-200 have worse performance than the standard SVMs, with the worst performance in the C-RSVMs by using the exponential weight function. All the results demonstrate the fact that in the non-stationary financial time series the recent training data points are more significant than the distant training data points. And by incorporating this prior knowledge into SVMs, the C-ASVMs are more effective in forecasting financial time series than the standard SVMs.

Fig. 3a gives the predicted and actual values of RDP+5 for the first 100 training data points in CME-SP. In C-ASVMs and C-RSVMs, only the exponential weight function is illustrated in this figure. It can be observed that the C-RSVMs fit best in the first 100 training data points, because they put more weights on the distant training data points which will be learned better than the recent training data points. Fig. 3b gives the results of the last 100 training data points. In this case, the C-ASVMs fit best since more weights are placed on the recent training data points in this method. Fig. 4 gives the predicted and actual values for the first 100 test data points. It is obvious that the C-ASVMs forecast more closely to the actual values and capture turning points better than both the standard SVMs and C-RSVMs. The same observations were made when the methods were applied to CBOT-US and EUREX-BUND.

The number of support vectors is also studied. Fig. 5 gives a comparison of non-error support vectors in the C-ASVMs and the standard SVMs. It can be found that the total number of non-error support vectors in the two methods is comparable while the corresponding data points are mostly different. In the C-ASVMs, most of the non-error support vectors are distributed in the recent training data points because the recent training data points have been penalized more heavily than the distant training data points. Fig. 6 shows the error support vectors which are different in the C-ASVMs and the standard SVMs. Error support vectors which are the same in the two methods are not shown in this figure. Compared to the standard SVMs, C-ASVMs have less error support vectors in the distant training data points. The reason can be explained by the fact that it is easier for the distant

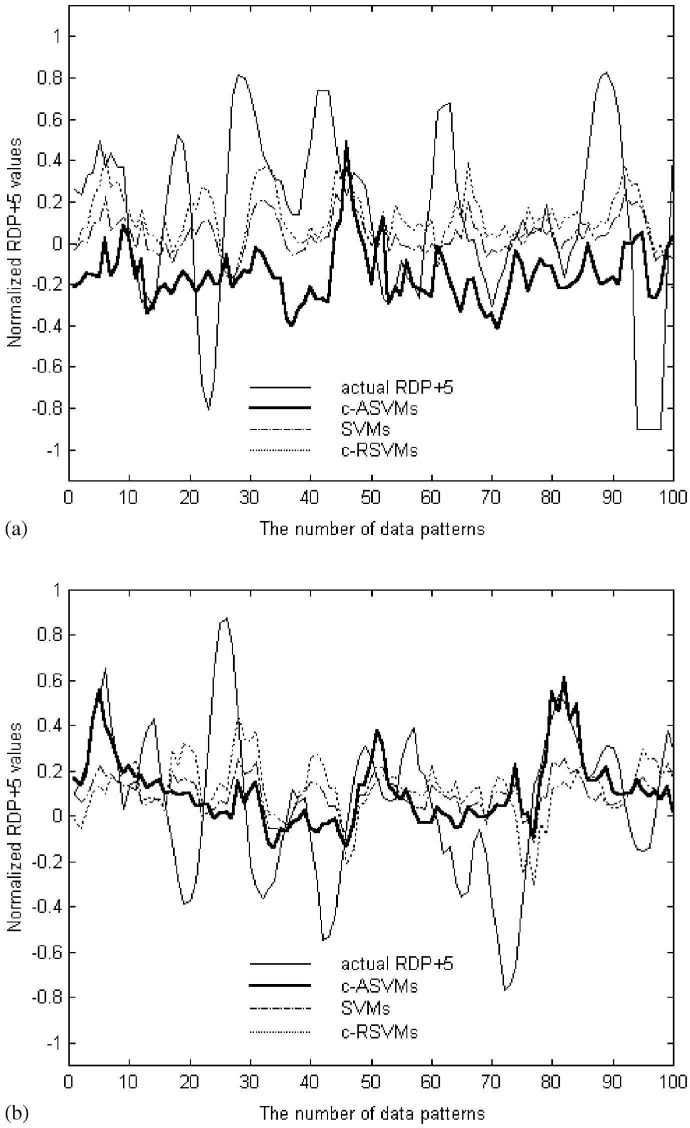


Fig. 3. The predicted and actual RDP + 5 on the first 100 training data points (a) and the last 100 training data points (b) in CME-SP. (a) C-RSVMs perform best. (b) C-ASVMs perform best.

training data points to converge to non-support vectors resulting from using small values of C. Thus, the solution of C-ASVMs is much sparser than that of the standard SVMs. The result is consistent with the theory of support vector error bound that the number of support vectors is an indication of the generalization performance of SVMs [16]. Usually, the fewer the number of support vectors, the higher the generalization performance of SVMs.

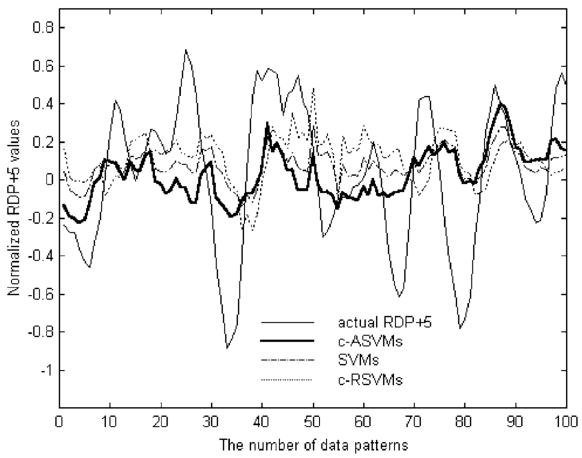


Fig. 4. The predicted and actual RDP + 5 on the test set in CME-SP. C-ASVMs perform best.

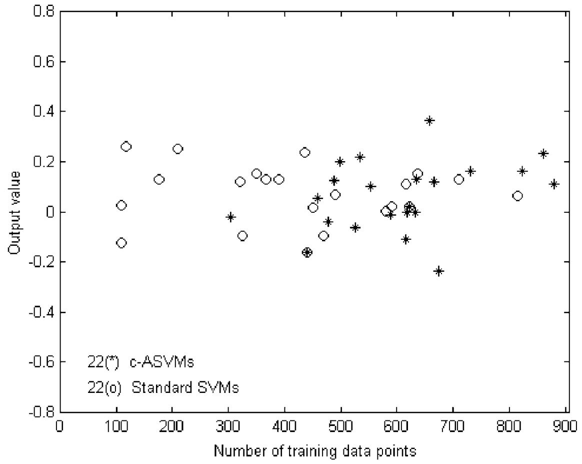


Fig. 5. Non-error support vectors in the C-ASVMs and the standard SVMs in CME-SP.

5. Conclusions

In this paper, a modified version of SVMs is proposed to model financial time series by taking into account the non-stationary characteristic of financial time series. This is obtained by modifying the empirical risk while keeping the regularized term in its original form. The performance of the modified SVMs is evaluated using three real futures contracts, and the simulation results demonstrated that the C-ASVMs is effective in dealing with the structural change of financial time series. The effectiveness of this method indicates that the empirical risk is an important

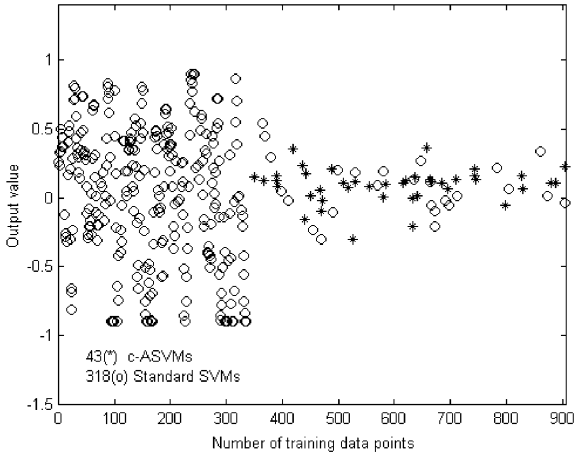


Fig. 6. Different error support vectors in the C-ASVMs and the standard SVMs in CME-SP.

component in SVMs, since SVMs is established on the SRM principle which consists of minimizing the sum of the empirical risk and the regularized term, and the regularized term was retained in its original form for the simulation. Furthermore, the C-ASVMs converge to fewer support vectors than those of the standard SVMs, resulting in a sparser representation of solution.

Future work will generalize the C-ASVMs into other futures contracts. It is also interesting to explore more sophisticated weights function which can closely follow the dynamics of financial time series. Other types of kernel functions will be explored for further improving the performance of SVMs in financial time series forecasting.

Acknowledgements

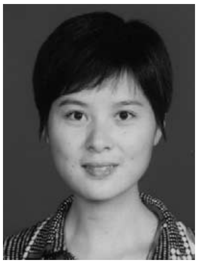
We would like to express our special thanks to Man-Drapeau Research Company for providing the data. In addition, we would like to thank the reviewers for making this a better paper.

References

[1] W.C. Chiang, T.L. Urban, G. Baildridge, A neural network approach to mutual fund net asset value forecasting, Omega 24 (2) (1996) 205–215.
[2] N.D. Freitas, M. Milo, P. Clarkson, Sequential support vector machines, Proceedings of the 1999 IEEE Signal Processing Society Workshop, Madison, WI, USA, 1999, pp. 31–40.
[3] J.W. Hall, Adaptive selection of US stocks with neural nets, in: G.J. Deboeck (Ed.), Trading On the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, Wiley, New York, 1994, pp. 45–65.
[4] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall International Inc., Englewood Cliffs, NJ, 1999.

[5] J. Hutchinson, A. Lo, T. Poggio, A non-parametric approach to pricing and hedging derivative securities via learning networks, J. Finance 49 (3) (1994) 851–889.
[6] T. Joachimes, Text categorization with support vector machines, Technical Report, 1997, ftp://ftp-ai.informatik.uni-dortmund.de/pub/Reports/report23.ps.z.
[7] I. Kaastra, S.B. Milton, Forecasting futures trading volume using neural networks, J. Futures Markets 15 (8) (1995) 853–970.
[8] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka, Stock market prediction system with modular neural networks, In: R.R. Trippi, E. Turban (Eds.), Neural Networks Finance Investing: Using Artificial Intelligence to Improve Real-World Performance, Irwin Publ., Chicago, 1996, pp. 497–510.
[9] J. Moody, J. Utans, Architecture selection strategies for neural networks: application to corporate bond rating prediction, in: A.P. Refenes (Ed.), Neural Networks in the Capital Markets, Wiley, New York, 1994, pp. 277–300.
[10] K.R. Muller, J.A. Smola, B. Scholkopf, Prediction time series with support vector machines, Proceedings of International Conference on Artificial Neural Networks, 1997, Lausanne, Switzerland, pp. 999–1004.
[11] K.R. Muller, J.A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, Prediction time series with support vector machines, in: B. Scholkopf, A.J. Smola (Eds.), Advances in Kernel Methods, The MIT Press, London, England, 1999, 243–254.
[12] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, Proceedings of the IEEE NNSP'97, Amelia Island, FL, 1997.
[13] A.N. Refenes, Y. Bentz, D.W. Bunn, A.N. Burgess, A.D. Zapranis, Financial time series modeling with discounted least squares back-propagation, Neurocomputing 14 (1997) 123–138.
[14] A.J. Smola, Learning with Kernels, Ph.D. Thesis, GMD, Birlinghoven, Germany, 1998.
[15] A.J. Smola, B. Scholkopf, A tutorial on support vector regression, NeuroCOLT Technical Report TR, Royal Holloway College, London, UK, 1998.
[16] B. Scholkopf, C. Burges, V. Vapnik, Extracting support data for a given task, in: U.M. Fayyad, R. Uthurusamy (Eds.), Proceedings of the First International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, CA, 1995.
[17] M. Schmidt, Identifying speaker with support vector networks, Interface '96 Proceedings, Sydney, 1996.
[18] M. Thomason, The practitioner methods and tool, J. Comput. Intell. Finance 7 (3) (1999) 36–45.
[19] M. Thomason, The practitioner methods and tool, J. Comput. Intell. Finance 7 (4) (1999) 35–45.
[20] M. Thomason, The practitioner methods and tool, J. Comput. Intell. Finance 7 (6) (1999) 35–45.
[21] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[22] V.N. Vapnik, S.E. Golowich, A.J. Smola, Support vector method for function approximation, regression estimation, and signal processing, Adv. Neural Inform. Process. Systems 9 (1996) 281–287.
[23] S.A.M. Yaser, A.F. Atiya, Introduction to financial forecasting, Appl. Intell. 6 (1996) 205–213.
[24] G.Q. Zhang, Y.H. Michael, Neural network forecasting of the British Pound/US Dollar Exchange Rate, Omega 26 (4) (1998) 495–506.

Francis E.H. Tay received his Ph.D. from MIT in 1996. He is currently an Assistant Professor in the National University of Singapore. His current research interests are in Design Methods and Design Automation including Data-Mining using soft computing.



Cao Lijuan received M.S. in Mechanical Engineering from the Southeast University of China, she is currently studying Ph.D. degree in National University of Singapore. Her research area centers on artificial intelligence methods such as neural networks, genetic algorithms, and support vector machines.