Neural Processing Letters **15:** 179–195, 2002.

ε-Descending Support Vector Machines for Financial Time Series Forecasting

FRANCIS E. H. TAY[1] and L. J. CAO[2]
[1]*Department of Mechanical & Production Engineering, National University of Singapore,10 Kent Ridge Crescent, 119260, Singapore. E-mail: mpetayeh@nus.edu.sg;* [2]*Institute of High Performance Computing, 89C Science Park Drive #02-11/12 118261 Singapore. E-mail: caolj@ihpc.nus.edu.sg*

Abstract. This paper proposes a modified version of support vector machines (SVMs), called ε-descending support vector machines (ε-DSVMs), to model non-stationary financial time series. The ε-DSVMs are obtained by incorporating the problem domain knowledge – non-stationarity of financial time series into SVMs. Unlike the standard SVMs which use a constant tube in all the training data points, the ε-DSVMs use an adaptive tube to deal with the structure changes in the data. The experiment shows that the ε-DSVMs generalize better than the standard SVMs in forecasting non-stationary financial time series. Another advantage of this modification is that the ε-DSVMs converge to fewer support vectors, resulting in a sparser representation of the solution.

Key words: non-stationary financial time series, support vector machines, tube size, structural risk minimization principle

1. Introduction

Financial time series are inherently non-stationary [1, 2]. That is, the distribution of financial time series changes over time. This leads to gradual changes in the actual relationship between the independent and dependent variables. In the modeling of financial time series, the learning algorithm used should take into account this characteristic. Usually, the information provided by the recent data points will be given more weights than that provided by the distant data points, as in non-stationary time series the recent data points could provide more important information than the distant data points [3, 4].

Recently, support vector machine (SVM) as a novel type of neural networks has received increasing attention in areas ranging from its original application of pattern recognition [5–7] to the extended application of regression estimation [8–11], due to its remarkable generalization performance. SVM was developed by Vapnik and his co-workers in 1995 [12]. Established based on the Structural Risk Minimization principle which seeks to minimize an upper bound of the generalization error rather than minimize the empirical error commonly implemented in other neural networks, SVMs achieve higher generalization performance than traditional neural networks in solving these machine learning problems. Another key property is that unlike

other networks' training which requires non-linear optimization with the danger of getting stuck into local minima, training SVMs is equivalent to solving a linearly constrained quadratic programming problem. Consequently, the solution of SVMs is always unique and globally optimal.

In SVMs, the solution to the problem is represented by sparse data points called support vectors. What are support vectors? In regression estimation, they are the training data points which associated approximation errors are equal to or larger than ε, the so-called tube size of SVMs. That is, they are the data points lying on or outside the ε-bound of the decision function. Therefore, the number of support vectors decreases as the tube size ε increases. In the case of a wide tube size where there are few support vectors, the decision function can be represented very sparsely. However, too wide a tube size will also depreciate the estimation accuracy as ε is equivalent to the approximation accuracy placed on the training data points. In the standard SVMs, ε is used as a constant value and selected empirically.

This paper proposes ε-descending SVMs (ε-DSVMs) to model financial time series by taking into account the non-stationarity of financial time series. Unlike the standard SVMs which use a constant tube in all the training data points, the ε-DSVMs use an adaptive tube which value will decrease from the distant training data points to the recent training data points. This modification is biased on the prior knowledge that in the non-stationary time series, the recent training data points could provide more important information than the distant training data points, and therefore it is desirable to place more weights on the recent training data points than the distant training data points. By using the proposed adaptive tube, the recent training data points will be approximated more accurately than the distant training data points. They also have larger probability of converging to support vectors. The proposed method is illustrated experimentally by using both simulated and real financial data sets. The experiment shows great improvement by the use of ε-DSVMs. The ε-DSVMs also have a sparser representation in the solution than the standard SVMs, resulting from the use of the adaptive tube.

This paper is organized as follows. Section 2 describes the basic theory of SVMs in regression estimation. Section 3 presents the ε-DSVMs. Section 4 discusses about the experimental results on both simulated and real data sets, followed by conclusions in the last section.

2. Theory of SVMs for Regression Approximation

Given a set of data points $(x_1, y_1), (x_2, y_2), \ldots, (x_1, y_1)$ ($x_i \in X \subseteq R^n$, $y_i \in Y \subseteq R$, l is the total number of training samples) randomly and independently generated from an unknown function, SVMs approximate the function using the following form:

$$f(x) = w \cdot \phi(x) + b$$

where $\phi(x)$ represents the high dimensional feature spaces which is nonlinearly

mapped from the input space x. The coefficients w and b are estimated by minimizing the regularized risk function (2).

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon(y_i, f(x_i)) \tag{2}$$

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

The first term $\|w\|^2$ is called the regularized term. Minimizing $\|w\|^2$ will make a function as flat as possible, thus playing the role of controlling the function capacity. The second term $\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon(y_i, f(x_i))$ is the empirical error 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). C is referred to as the regularized constant. ε is the tube size of SVMs. They are both user-prescribed parameters and determined empirically.

To get the estimations of w and b, equation (2) is transformed to the primal objective function (4) by introducing the positive slack variables $\xi_i^{(*)}$ (( ) denotes variables with and without *).

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{4}$$

subject to

$$\begin{aligned} &y_i - w \cdot \phi(x_i) - b \leqslant \varepsilon + \xi_i \\ &w \cdot \phi(x_i) + b - y_i \leqslant \varepsilon + \xi_i^*, \quad i = 1, \ldots, l. \\ &\xi_i^{(*)} \geqslant 0 \end{aligned}$$

Finally, by introducing Lagrange multipliers and exploiting the optimality constraints, the decision function (1) has the following explicit form [12]:

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

In function (5), $a_i^{(*)}$ are the so-called Lagrange multipliers. They satisfy the equalities $a^i * a_i^* = 0$, $a_i \geqslant 0$ and $a_i^* \geqslant 0$ where $i = 1, \ldots, l$, and they are obtained by maximizing the dual function of (4), which has the following form:

$$W(a_i^{(*)}) = \sum_{i=1}^{l} y_i(a_i - a_i^*) - \varepsilon\sum_{i=1}^{l}(a_i + a_i^*) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(a_i - a_i^*)(a_j - a_j^*)K(x_i, x_j) \tag{6}$$

with the following constraints:

$$\sum_{i=1}^{i}(a_i - a_i^*) = 0$$

$$0 \leqslant a_i^{(*)} \leqslant C, \quad i = 1, \ldots, l.$$

In function (6), $K(x_i, x_j)$ is defined as the kernel function. Its value is equal to $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. The elegance of using the kernel function is that one can deal with feature spaces of arbitrary dimensionality without having to compute the map $\phi(x)$ explicitly [13]. Any function that satisfies Mercer's condition [12] can be used as the kernel function.

Based on the Karush–Kuhn–Tucker (KKT) conditions [14], only a number of coefficients $(a_i - a_i^*)$ will assume nonzero, and the corresponding training data points have approximation errors equal to or larger than ε, and are referred to as support vectors. According to function (5), it is evident that only the support vectors are used to determine the decision function as the values of $(a_i - a_i^*)$ for the other training data points are equal to zero. As support vectors are usually only a small subset of the training data points, this characteristic is referred to as the sparsity of the solution.

3. ε-Descending Support Vector Machines (ε-DSVMs)

In ε-DSVMs, instead of a constant value, the tube size adopts the following exponential function:

$$\varepsilon_i = \varepsilon \frac{1 + \exp(p - 2p * i/l)}{2} \tag{7}$$

Where i represents the data sequence, with $i = 1$ being the most recent training data point and $i = 1$ being the most distant training data point. p is the parameter to control the descending rate. ε_i is called the descending tube as its value will decrease from the distant training data points to the recent training data points.

In ε-DSVMs, the recent training data points are penalized more heavily than the distant training data points can be explained from both the approximation accuracy aspect and the characteristic of the solution of SVMs aspect. As aforementioned, ε is equivalent to the approximation accuracy placed on the training data points. A small ε corresponds to a large slack variable $\xi_i^{(*)}$ and high approximation accuracy. On the contrary, a large ε corresponds to a small slack variable $\xi_i^{(*)}$ and low approximation accuracy. According to (4), a large slack variable will make the empirical error have a large impact relatively to the regularized term. Therefore, the data point by using a smaller value of ε will be penalized more heavily than the data point by using a larger value of ε. The characteristic of the solution of SVMs can also be used to explain that there are more weights in the recent training data points than the distant training

data points. As described in Section 2, the solution of SVMs is represented by support vectors. Also the support vectors are a decreasing function of ε. This means that the recent training data points by using a smaller ε will have a larger probability of converging to the determinant support vectors than the distant training data points by using a larger ε. Thus, the recent training data points will be obtained more attention in the representation of the solution than the distant training data points.

The behaviours of the weight function (7) can be summarized as follows. Some examples are illustrated in Figure 1.

(i) When $p \to 0$, then

$$\lim_{p \to 0} \varepsilon_i = \varepsilon_0.$$

In this case, the weights in all the training data points are equal to 1.0.

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

$$\lim_{p \to \infty} \varepsilon_i = \begin{cases} \infty & i < \dfrac{1}{2} \\ 0.5\varepsilon & i \geqslant \dfrac{l}{2} \end{cases}.$$

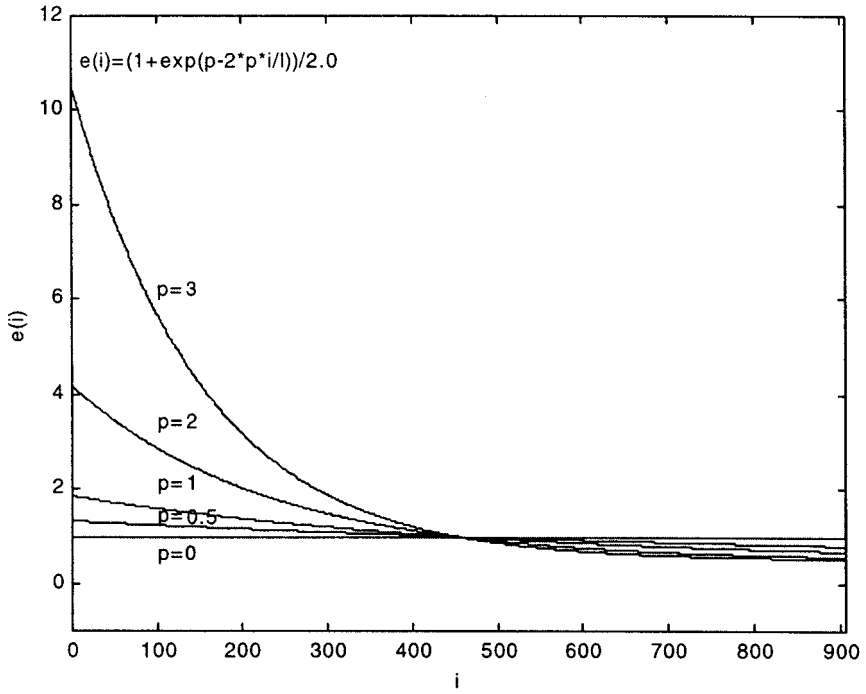


Figure 1. The weight function of ε-DSVMs. In the x-axis, i represents the data sequence.

In this case, the weights for the first half of the training data points are increased to an infinite value while the weights for the second half of the training data points are equal to 0.5.

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

In ε-DSVMs, the regularized risk function has the original form but the constraints are changed to (8) whereby every training data point uses different tube size ε_i.

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

subject to

$$\begin{aligned} & y_i - w \cdot \phi(x_i) - b \leqslant \varepsilon_i + \xi_i \\ & w \cdot \phi(x_i) + b - y_i \leqslant \varepsilon_i + \xi_i^*, \quad i = 1, \ldots, l. \\ & \xi_i^{(*)} \geqslant 0 \end{aligned} \tag{8}$$

Thus, the dual function has the function form of (9) with the original constraints.

$$W(a_i^{(*)}) = \sum_{i=1}^{l} y_i(a_i - a_i^*) - \sum_{i=1}^{l} \varepsilon_i(a_i + a_i^*) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(a_i - a_i^*)(a_j - a_j^*)K(x_i, x_j) \tag{9}$$

subject to

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

The Sequential Minimal Optimization algorithm extended by Scholkopf and Smola [15, 16] can be used to optimize the ε-DSVMs whereby the Lagrange multipliers are obtained according to function (9).

4. Experiment Results

4.1. TWO SIMULAED DATA SETS

Two simulated data sets studied in [4] are examined in the first series of experiment. They are referred to as data-1 and data-2. Each data set contains seven periods

of the sine wave which are defined by the following functions:

$$\text{data} - 1:\ y = m_1(n + x) * \sin(2\pi x) + m_2 \tag{10}$$

$$\text{data} - 2:\ y = m_1(n + x) + m_2(n + x) * \sin(2\pi x) \tag{11}$$

Where $m_1 = 0.1$, $m_2 = 0.5$, $x \in [0, 1]$, $n \in \{0, \ldots, 6\}$. Both time series are non-stationary in the sense that the variance of data-1 and the variance and mean of data-2 change over time. The same experimental setup used by Refenes et al. is used here. Briefly, the setup is as follows: in each of the seven periods, there are 100 consecutive and equally spaced data points. The first six periods (i.e. 600 data points) are used for training and the seventh period (i.e. 100 data points) for testing.

The purpose of the experiment is to compare the ε-DSVMs with the standard SVMs. To do this, the Gaussian function is chosen as the kernel function of SVMs. The values of δ^2, C, and ε are respectively chosen as 0.01, 10 and 0.05 as these values produce the smallest NMSE on the test set in the standard SVMs. The same values of the parameters are used in ε-DSVMs for compare. The NMSE of the test set is calculated as follows:

$$NMSE = \frac{1}{\delta^2 n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{12}$$

$$\delta^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{13}$$

$$\bar{y} = \sum_{i=1}^{n} y_i \tag{14}$$

Where n represents the total number of data points in the test set. $\hat{y}$ represents the predicted value. $\bar{y}$ denotes the mean of the actual output values.

In ε-DSVMs, the converged NMSE with various control rates is illustrated in Figure 2. This figure shows that in both data-1 and data-2, the NMSE firstly dramatically decreases as p increase, and then it violates when p keeps on increasing. This indicates that the ε-DSVMs could produce a smaller NMSE than the standard SVMs corresponding to $p = 0$. The difference in performance increases with the incremental of p (0–10), but when p is larger than 10, there is a little overweight to the recent training data points.

Figure 3 and Figure 4 respectively illustrate the predicted and actual values in data-1 and data-2. In the ε-DSVMs, the value of p that produces the smallest NMSE on the test set is used. On the training set as shown in Figures 3 (a) and Figure 4 (a), the standard SVMs forecast more closely to the actual values than the ε-DSVMs in the distant data points (about the first 400 data points), but in the recent data

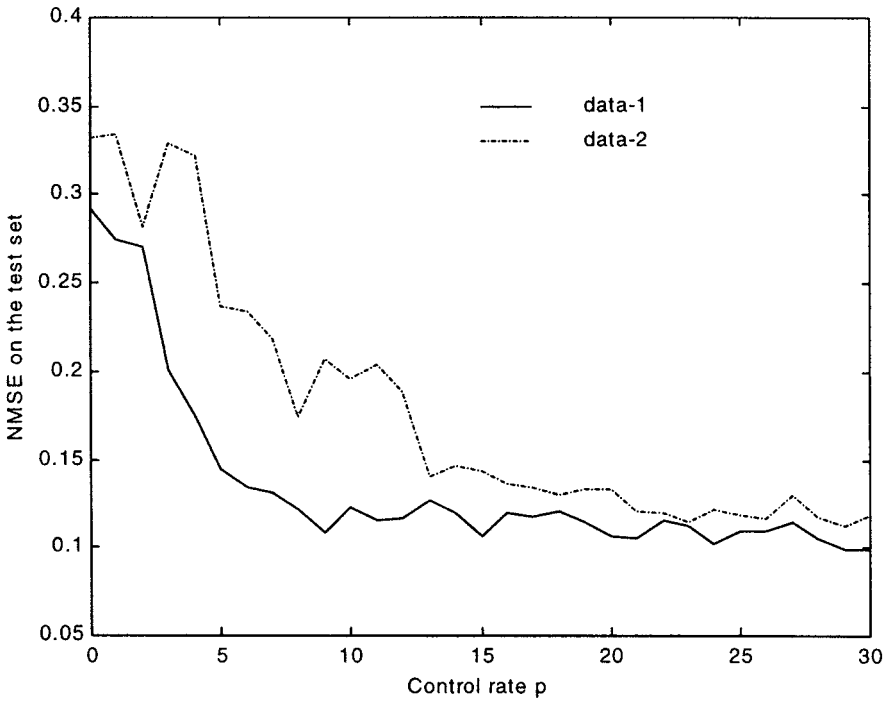
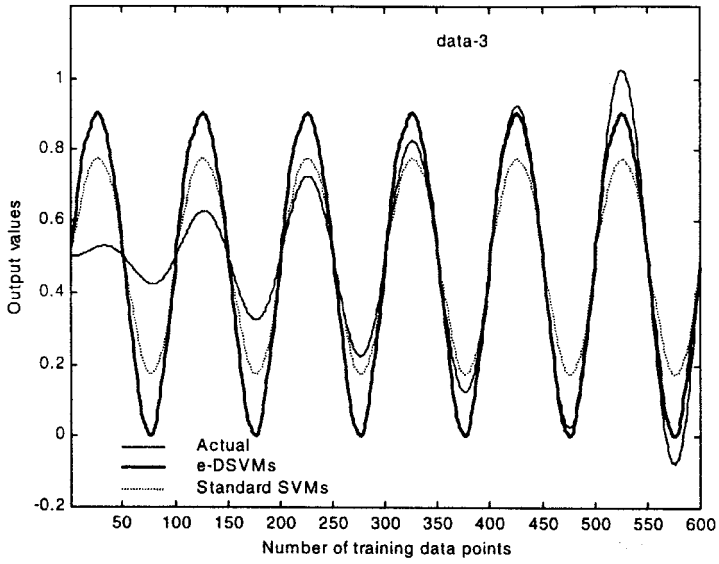


Figure 2. Converged NMSE with various control rates p in ε-DSVMs.

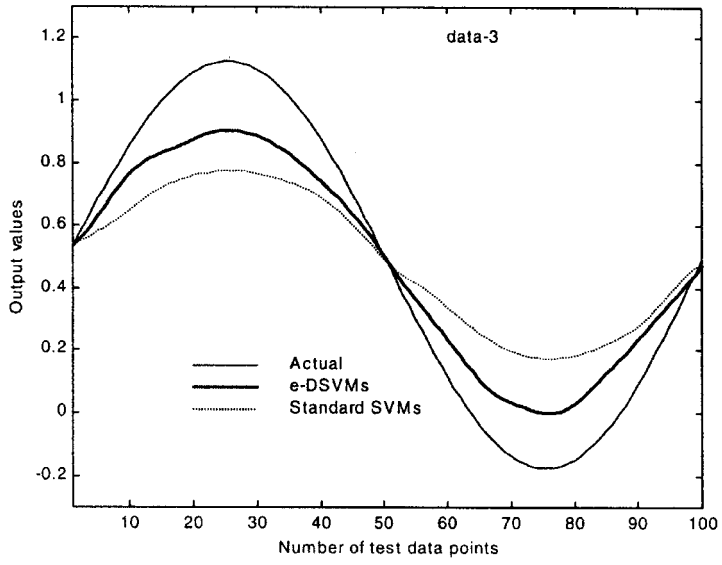
points (about the last 200 data points) the ε-DSVMs forecast more closely to the actual values than the standard SVMs. The result is consistent with the idea of ε-DSVMs. Figure 3 (b) and Figure 4 (b) show that on the test set the ε-DSVMs perform much better than the standard SVMs.

For a better understanding the ε-DSVMs, the converged support vectors in the two methods are also studied. Figure 5 gives a comparison of the support vectors with $0 < |a_i - a_i^*| < C$ (referred to as non-error support vectors) in the ε-DSVMs and 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 ε-DSVMs, 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.

Figure 6 shows the support vectors with $|a_i - a_i^*| = C$ (referred to as error support vectors) which are different in the ε-DSVMs and standard SVMs. Error support vectors which are the same in the two methods are not shown in this figure. Compared to the standard SVMs, ε-DSVMs have fewer error support vectors in

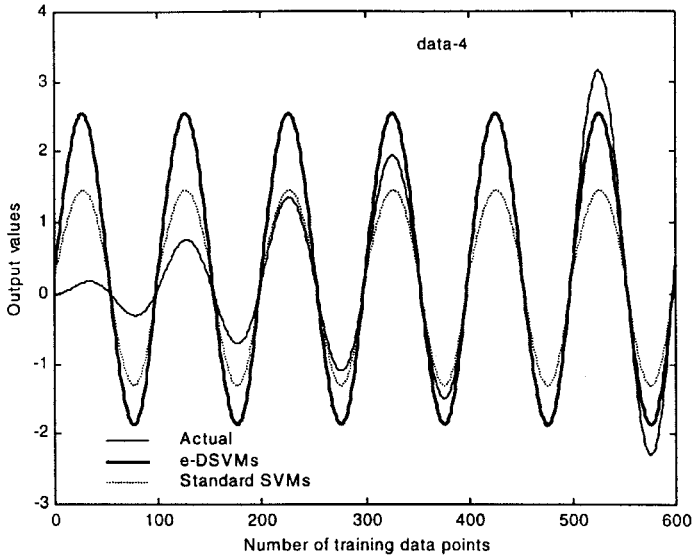


(a)

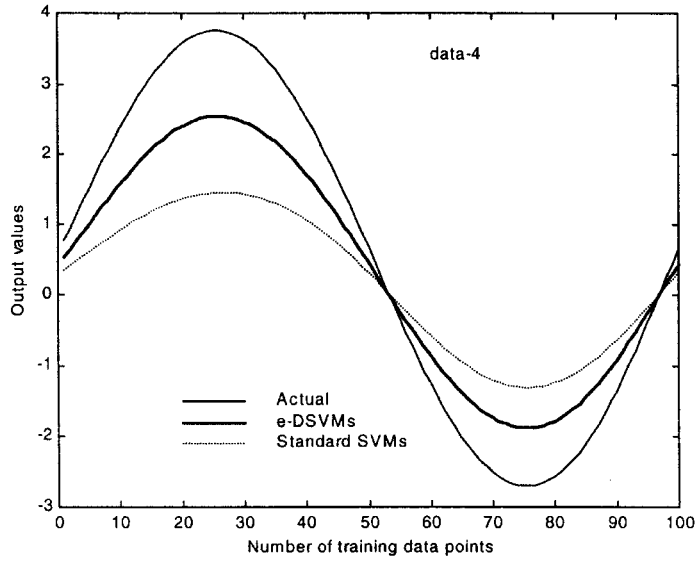


(b)

Figure 3. Predicted and actual values in data-3. (a) On the training set. (b) On the test set.



(a)



(b)

Figure 4. Predicted and actual values in data-4. (a) On the training set. (b) On the test set.

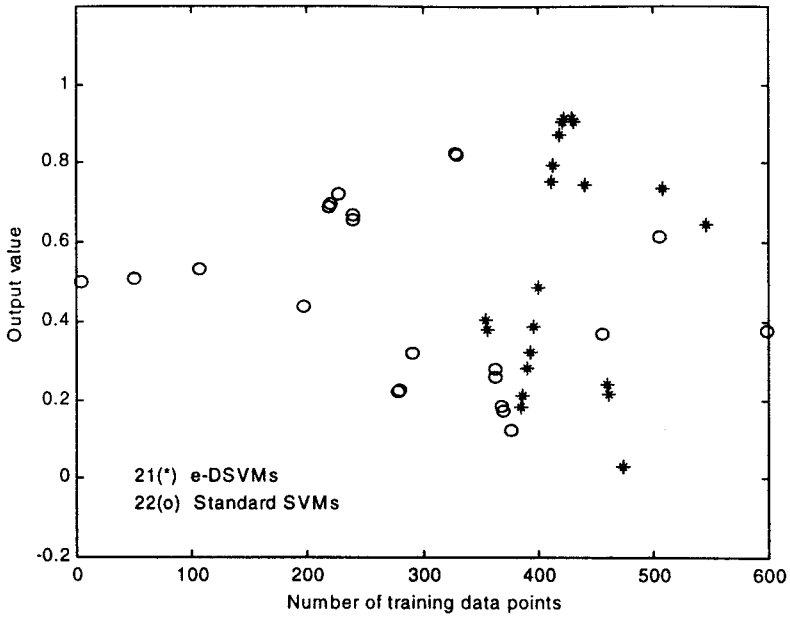


Figure 5. Non-error support vectors in the ε-DSVMs and standard SVMs.

the distant data points than the standard SVMs, resulting from the use of larger ε. Thus, the solution of ε-DSVMs is much sparser than that of the standard SVMs.

4.2. FINANCIAL DATA SETS

Five real futures contracts collated from the Chicago Mercantile Market are examined in the experiment. They are the Standard&Poor 500 stock index futures (CME-SP), United Sates 30-year government bond (CBOT-US), Unite States 10-year government bond (CBOT-BO), German 10-year government bond (EUREX-BUND) and French government stock index futures (MATIF-CAC40). A subset of the available data is used to reduce the requirement of the network design. The corresponding time periods used are listed in Table I. The daily closing prices are used as the data sets.

Table I. Five futures contracts and their used time periods

Futures	Time period
CME-SP	24/05/1989 08/10/1993
CBOT-US	23/05/1991 20/10/1995
CBOT-BO	23/05/1991 17/10/1995
EUREX-BUND	31/05/1991 25/10/1995
MATIF-CAC40	18/10/1993 09/04/1998

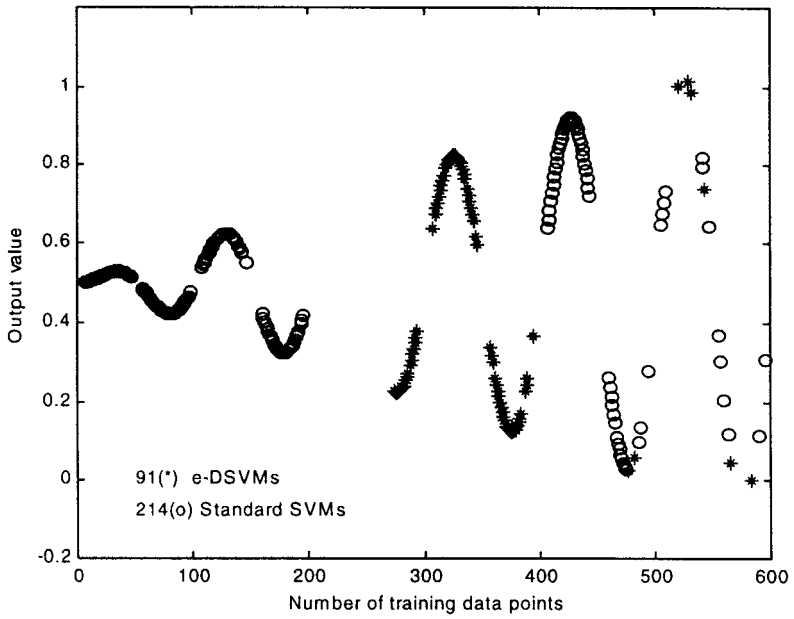
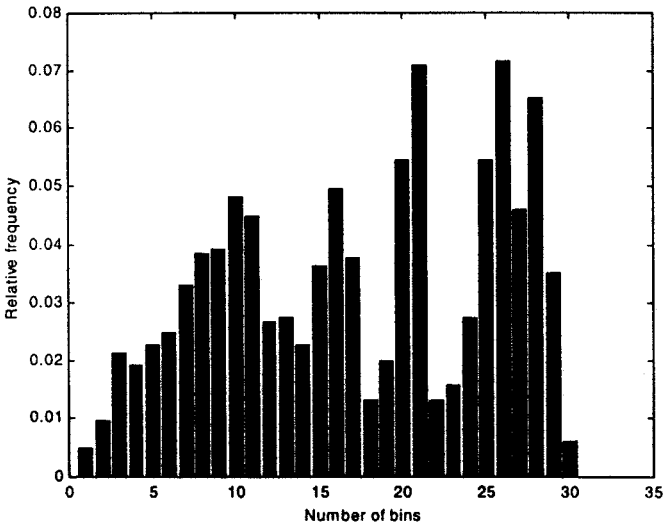


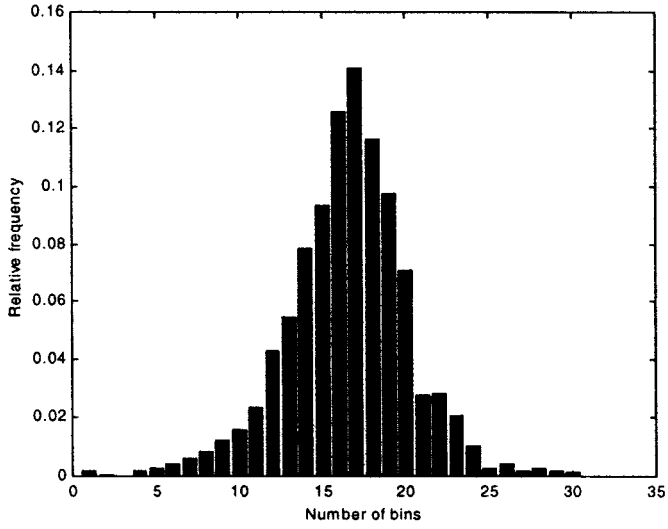
Figure 6. Different error support vectors in the ε-DSVMs and standad SVMs.

Choosing a suitable forecasting horizon is the first step in data preprocessing. From the trading aspect, the forecasting horizon should be sufficiently long so that the over-trading resulting in excessive transaction costs could be avoided. From the prediction aspect, the forecasting horizon should be short enough as the persistence of financial markets is of limited duration. As suggested by Thomason [17], a forecasting horizon of 5 days is a suitable choice for daily data. As the precise values of the daily prices is often not as meaningful to trading as its relative magnitude, and also the high-frequency components in financial data is often more difficult to successfully model, the original closing price is transformed into a five-day relative difference in percentage of price (RDP). As mentioned by Thomason, 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 will follow more closely to a normal distribution as illustrated in Figure 7. This modification to the data distribution will improve the predictive power of the neural network.

In each futures contract, a total of 20 candidate indicators are constructed. They are the 3 lagged transformed closing prices (x_1, x_2, x_3), 14 lagged RDP values ($x_4, \ldots x_{17}$) and 3 technical indicators: moving average convergence divergence x_{18} (MACD), on balance volume x_{19} (OBV), and volatility x_{20}. The lagged RDP values and transformed closing price are all recommended by [18]. The MACD is defined as the difference of two exponential moving averages, and it is commonly used to predict market trends in financial markets. The OBV moves in the same



(a)



(b)

Figure 7. Histograms. (a) Of CME-SP daily closing price. (b) Of RDP + 5.

direction as price. That is, as price increases, the OBV will gain magnitude. As OBV can relate the volume into price, it is also used as input here. The volatility denotes the range of the highest price and lowest price in one day, and it is often used as a measure of the market risk. The calculations for all the indicators are given in Table II. Then genetic algorithm (GA) [19] is applied to SVMs for feature selection. The selected feature set of GA is used as the inputs of SVMs. A detailed description about feature selection in SVMs can be referred to [20, 21]. As listed in Table II, the output variable RDP + 5 is obtained by first smoothening the closing price with a 3-day exponential moving average, because the application of a smoothing transform to the dependent variable generally enhances the prediction performance of neural networks [17].

The long left tail in Figure 7 (b) 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 the neural networks, RDP values beyond the limits of ±2 standard deviations are selected as outliers. They are replaced with the closest marginal values. Another pre-processing technique used in this study is data scaling. All the data points are scaled into the range of [−0.9, 0.9] as the data points include both positive

Table II. Input and output variable

	Indicator	Calculation
Input variables	x_1	$P(i) - \overline{EMA_{100}(i)}$
	x_2	$P(i-1) - \overline{EMA_{100}(i-1)}$
	x_3	$P(i-2) - \overline{EMA_{100}(i-2)}$
	x_4	$(p(i) - p(i-5))/p(i-5) * 100$
	x_5	$(p(i-1) - p(i-6))/p(i-6) * 100$
	x_6	$(p(i-2) - p(i-7))/p(i-7) * 100$
	x_7	$(p(i-3) - p(i-8))/p(i-8) * 100$
	x_8	$(p(i-4) - p(i-9))/p(i-9) * 100$
	x_9	$(p(i) - p(i-10))/p(i-10) * 100$
	x_{10}	$(p(i-1) - p(i-11))/p(i-11) * 100$
	x_{11}	$(p(i-2) - p(i-12))/p(i-12) * 100$
	x_{12}	$(p(i) - p(i-15))/p(i-15) * 100$
	x_{13}	$(p(i-1) - p(i-16))/p(i-16) * 100$
	x_{14}	$(p(i-2) - p(i-17))/p(i-17) * 100$
	x_{15}	$(p(i) - p(i-20))/p(i-20) * 100$
	x_{16}	$(p(i-1) - p(i-21))/p(i-21) * 100$
	x_{17}	$(p(i-2) - p(i-22))/p(i-22) * 100$
	x_{18}	$\overline{EMA_{10}(i)} - \overline{EMA_{20}(i)}$
	x_{19}	$\begin{cases} p(i) \geqslant p(i-1) & obv+ = volume(i) \\ p(i) \geqslant p(i-1) & obv- = volume(i) \end{cases}$
	x_{20}	$k * sqrt(1/n * \sum_{i=1}^{n} \log^2(h(i)/l(i)))(k = 80, n = 5)$
Output variable	RDP + 5	$\overline{(p(i+5)} - \overline{p(i))/p(i)} * 100$
		$\overline{p(i)} = \overline{EMA_3(i)}$

$EMA_n(i)$ is the n-day exponential moving average of the ith day.
$p(i)$, $h(i)$, $l(i)$ are the closing, highest and lowest price of the ith day.

and negative values. All of the five data sets are partitioned into three parts according to the time sequence. The first part is used as the training set, the second part is used as the validation set which is to select the optimal parameters of SVMs. The last part is used as the test set. 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 all the data sets.

In this investigation, the Gaussian function is still used as the kernel function of the SVMs. The optimal values of δ^2, C and ε in the standard SVMs are chosen based on the validation set. The same values of the parameters are used in the ε-DSVMs. The validation set is also used in the ε-DSVMs to choose the optimal p.

The best results obtained in the ε-DSVMs and standard SVMs are listed in Table III. It can be observed that in four of the studied futures (CME-SP, CBOT-US, EUREX-BUND and MATIF-CAC40), the ε-DSVMs converge to a smaller NMSE than the standard SVMs. In CBOT-BO, there are comparable results between the ε-DSVMs and standard SVMs. This may be explained by the fact that CBOT-BO is more stationary than the other futures so that the ε-DSVMs have no dominance over the standard SVMs.

A paired t-test [22] is performed to determine if there is significant difference between the two methods based on the NMSE of the test set. The calculated t-value (Table III) shows that the ε-DSVMs outperform the standard SVMs with $\alpha = 5\%$ significance level for a one-tailed test. It can be concluded that the ε-DSVMs are more effective in modelling non-stationary financial time series.

5. Conclusions

This paper proposes the ε-DSVMs to model financial time series by incorporating the non-stationarity of financial time series into SVMs. The ε-DSVMs use an adaptive tube to place more weights on the distant training data points and less weights on the recent training data points. The superior performance of the ε-DSVMs over the standard SVMs is demonstrated by using both simulated data sets and five real futures contracts. Another advantage of the ε-DSVMs is that their solution is sparser than the standard SVMs.

One question may arise from the modification. When the predicted time series are very non-stationary, a better result will be obtained when the ε-DSVMs use a large value of p. This will cause a large number of distant training data points to converge

Table III. Averaged NMSE on the test set in real financial time series

Methods	Standard SVMs	ε-DSVMs
CME-SP	0.8407	0.8368
CBOT-US	0.9465	0.9040
CBOT-BO	0.9664	0.9608
EUREX-BUND	0.8988	0.8765
MATIF-CAC40	1.0130	0.9818
t-values	$2.4334 > t_{0.05,4} = 2.132$	

to non-support vectors. Does it mean that the ε-DSVMs are equivalent to deleting the distant training data points? An experimental investigation shows that there are two differences between the ε-DSVMs and the method of purposely deleting the distant training data points. Firstly, there are still equal tube sizes in all the training data points in the latter method. So the recent training data points do not get more attention than the distant training data points. Therefore, the prediction accuracy in the deleting method is inferior to the ε-DSVMs. Secondly, as there is no prior knowledge of how many distant training points could be eliminated, the deleting method is not very practical. If too many distant data points are deleted, the long-term relationship between the inputs and output variable will be distorted and thus it will lead to a poor prediction.

Future work will involve a theoretic analysis of the ε-DSVMs. More sophisticated weights function which can closely follow the dynamics of financial time series will be explored for further improving the performance of SVMs in financial time series forecasting.

References

1. Hall, J. W.: Adaptive selection of U.S. stocks with neural nets, Trading On the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, ed by G. J. Deboeck, pp. 45–65. New York, Wiley, 1994.
2. Yaser, S. A. M. and Atiya, A. F.: Introduction to financial forecasting, *Applied Intelligence*, **6** (1996), 205–213.
3. Freitas, N. D., Milo, M. and Clarkson, P.: Sequential support vector machines, Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp. 31–40.
4. Refenes, A. N., Bentz, Y., Bunn, D. W., Burgess, A. N. and Zapranis, A. D.: Financial time series modeling with discounted least squares back-propagation, *Neurocomputing*, **14** (1997), 123–138.
5. Scholkopf, B., Burges, C. and Vapnik, V.: Extracting support data for a given task, Proceedings of First International Conference on Knowledge Discovery & Data Mining, AAAI press, Menlo Park, CA, 1995.
6. Schmidt, M.: Identifying speaker with support vector networks, In Interface '96 Proceedings, Sydney, 1996.
7. Joachimes, T.: Text categorization with support vector machines, Technical Report, ftp://ftp-ai.informatik.uni-dortmund.de/pub/Reports/report23.ps.z.
8 Muller, R., Smola, J. A. and Scholkopf, B.: Prediction time series with support vector machines, In Proceedings of International Conference on Artificial Neural Networks, pp. 999, 1997.
9. Mukherjee, S., Osuna E. and Girosi, F.: Nonlinear prediction of chaotic time series using support vector machines, Proc. Of IEEE NNSP'97, Amelia Island, FL, 1997.
10. Vapnik, V. N., Golowich, S. E. and Smola, A. J.: Support vector method for function approximation, regression estimation, and signal processing, *Advances in Neural Information Processing Systems*, **9** (1996), 281–287.
11. Muller, K. R., Smola, J. A., Ratsch, G., Scholkopf, B. and Kohlmorgen, J.: Prediction time series with support vector machines, *Advances in Kernel Methods*, The MIT Press, London, England, 1999.

12. Vapnik, V. N.: *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
13. Cristianini, N. and Taylor, J. S.: *An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods*, New York: Cambridge University Press, 2000.
14. Kuhn, H. W. and Tucker, A. W.: Nonlinear programming, In Proceedings 2th Berkeley Symposium on Mathematical Statistics and Probabilistics, Berkeley, University of California Press, pp. 481–492, 1951.
15. Smola, A. J. and Scholkopf, B.: A tutorial on support vector regression, NeuroCOLT Technical Report TR, Royal Holloway College, London, UK, 1998.
16. Smola, A. J.: *Learning with Kernels*, PhD Thesis, GMD, Birlinghoven, Germany,1998.
17. Thomason, M.: The practitioner methods and tool, *Journal of Computational Intelligence in Finance*, **7**(3) (1999), 36–45.
18. Thomason, M.: The practitioner methods and tool, *Journal of Computational Intelligence in Finance*, **7**(4) (1999), 35–45.
19. Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.
20. Tay, F. E. H. and Cao, L. J.: Saliency analysis of support vector machines for feature selection, accepted by the *Journal of Neural Network World*, 2001.
21. Tay, F. E. H. and Cao, L. J.: A comparative study of saliency analysis and genetic algorithm for feature selection in support vector machines, accepted by the *Journal of Intelligent Data Analysis*, 2000.
22. Montgomery, D. C. and Runger, G. C.: *Applied Statistics and Probability for Engineers*, Wiley & Sons, New York, 1999.