Financial Time Series Prediction Using Non-fixed and Asymmetrical Margin Setting with Momentum in Support Vector Regression

Haiqin Yang, Irwin King, Laiwan Chan and Kaizhu Huang

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hqyang, king, lwchan, kzhuang}@cse.cuhk.edu.hk

Abstract. Recently, Support Vector Regression (SVR) has been applied to financial time series prediction. Typical characteristics of financial time series are non-stationary and noisy in nature. The volatility, usually time-varying, of the time series therefore contains some valuable information about the series. Previously, we had proposed to use the volatility in the data to adaptively changing the width of the margin in SVR. We have noticed that upside margin and downside margin would not necessary be the same, and we have observed that their choice would affect the upside risk, downside risk and as well as the overall prediction performance. In this work, we introduce a novel approach to adapt the asymmetrical margins using momentum. We applied and compared this method to predict the Hang Seng Index and Dow Jones Industrial Average.

Key words: Non-fixed and Asymmetrical Margin, Momentum, Support Vector Regression, Financial Time Series Prediction

1 Introduction

A time series is a collection of observations that measures the status of some activities over time [8, 9]. It is the historical record of some activities, with a consistency in the activity and the method of measurement, where the measurement is taken at equally spaced intervals, e.g., day, week, month, etc. In practice, there are various time series and they are collected in a wide range of disciplines, from engineering to economics. For example, the air temperatures of a certain city measured in successive days or weeks consists of a series; a certain share prices occurred in successive days, months is another series.

Of all the different possible time series, the financial time series is unusual since it contains several specific characteristics: small sample sizes, high noise, non-stationarity, non-linearity, and varying associated risk.

Support Vector Machines (SVMs) are recent generalization models, which find a generalization function through training samples, especially by small samples. It also extends to solve the regression problem by Support Vector Regression (SVR) [37, 30]. Nowadays, SVR has been successfully applied to time series prediction [18, 16] and financial forecasting [33, 31].

Usually, SVR uses the ε-insensitive loss function to measure the empirical risk (training error). This loss function not only measures the training error, but also controls the sparsity of the solution. When the ε-margin value is increased, it tends to reduce the number of support vectors [35]. Extremely, a constant objective function may occur when the width of margin is too wide. Therefore, the ε-margin value setting affects the complexity and the generalization of the objective function indirectly.

Since the ε-margin value setting is very important, researchers proposed various methods to determine it. Usually, there are four kinds of methods to deal with it. First, most practitioners set the ε-margin value to a non-negative constant just for convenience. For example, in [33], they simply set the margin width to 0. This amounts to the least modulus loss function. In other instances the margin width has been set to a very small value [37, 16, 7]. The second method is the cross-validation technique [18, 6]. It is usually too expensive in terms of computation. A more efficient approach is to use another variant called ν-SVR [26, 28, 27, 21], which determines ε by using another parameter ν. It states that ν may be easier to specify than ε. This introduces another parameter setting problem. Another approach by Smola et al [29] is to find the "optimal" choice of ε based on maximizing the statistical efficiency of a location parameter estimator. They showed that the asymptotically optimal ε should scale linearly with the input noise of the training data, and this was verified experimentally, but their predicted value of the optimal ε does not have a close match with their experimental results. In sum, the previous methods tries to use a suitable or an optimal ε-margin value for that particular data set; the ε-margin value is always fixed and symmetrical for that data set. However, in stock market, it is volatile and the associated risk changes with time. A fixed and symmetrical ε-margin setting may lack the ability to capture stock market information promptly and may not be suitable for stock market prediction. Furthermore, our experience showed that ε-margin value is not necessary the same all the time [40].

In [40], we have extended the standard SVR with adaptive margin and classified it into four categories: Fixed and Symmetrical Margin (FASM), Fixed and Asymmetrical Margin (FAAM), Non-fixed and Symmetrical Margin (NASM), and Non-fixed and Asymmetrical Margin (NAAM). Comparing FASM with FAAM, we know that the downside risk can be reduced by employing asymmetrical margins. While comparing FASM, FAAM with NASM, a good predictive result is obtained by exploiting the standard deviation to calculate the margin. However, NAAM requires the adaptation of the margin width and the degree of asymmetry, and no exact algorithm for such margin setting has been introduced.

In this work, we propose to use NAAM which combines two characteristics of the margin; non-fixed and asymmetry, to reduce the predictive downside risk while controlling the accuracy of the financial prediction. More specifically, we add a momentum term to achieve this. The width of the margin is determined by the standard deviation [40]. The asymmetry of the margin is controlled by the momentum. This momentum term can trace the up trend and down tendency of the stock prices. Since the financial time series often follows a long term trend but with small short term fluctuations, we exploit a larger up margin and a smaller down margin to under-predict the stock price when the momentum is positive and we use a smaller up margin and a larger down margin to over-predict the stock price while the momentum is negative. A simple illustration is shown in Fig. 1. In this work, we will use this downside risk avoiding strategy in the prediction.

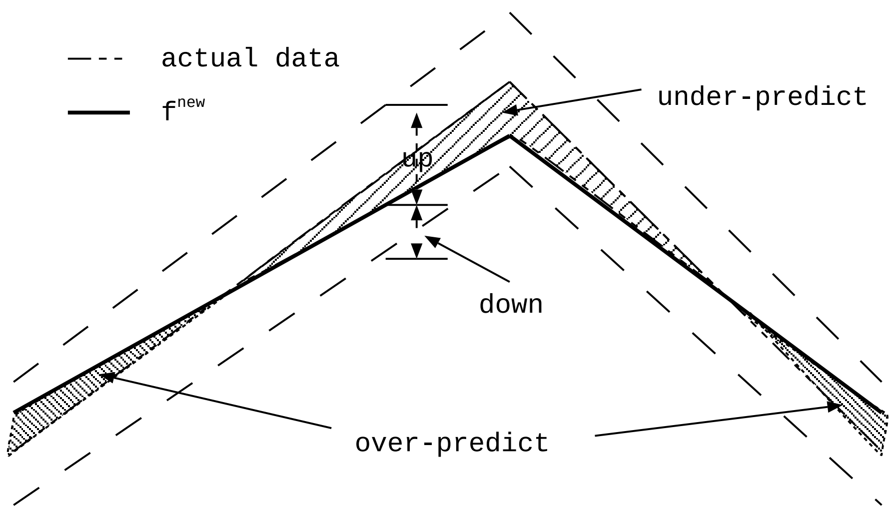

Fig. 1. Margin setting

We organize the paper as follows. First, we give an overview of the time series analysis models in Sect. 2. Next, we introduce the SVR with a general ε-insensitive loss function and the concept of momentum in Sect. 3. The accuracy metrics and experimental results are elucidated in Sect. 4. Finally, we conclude the paper with a brief discussion and final remarks in Sect. 5.

2 Time Series Analysis Models

There are many models for time series analysis. Generally, they are classified into *linear* and *non-linear* models, see Fig. 2.

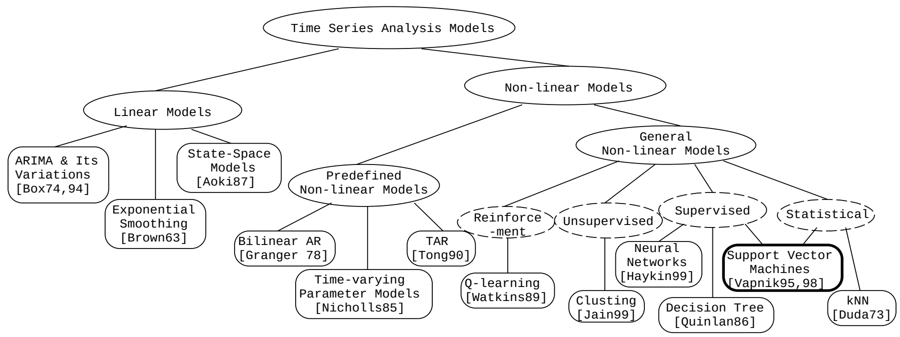

Fig. 2. Time series analysis models

Linear models have the characteristics of simplicity, usefulness and easy application. They work well for linear time series, but may fail otherwise. ARIMA models are typical linear models and used as the benchmark models for time series analysis [4].

Although linear models have both mathematical and practical convenience, there is no reason why real life time series should all be linear, and so the use of non-linear models seems potential promising [9]. In the 1980's, non-linear models were investigated and were proposed by the existing linear models [13, 22]. For example, Bilinear autoregressive or Bilinear AR models [12], time-varying parameter models [24, 20] and threshold autoregressive (TAR) model [32]. These models are agreeable due to the scrutiny given in their development for the standard statistical considerations of model specification, estimation, and diagnosis, but their general parametric nature tends to require significant a prior knowledge of the form of relationship being modeled. Therefore, they are not effective for modeling financial time series because the non-linear functions are hard to choose. Another class of non-linear models are general non-linear models, also called machine learning. These models can learn a model from a given time series without non-linear assumptions. They include reinforcement learning, e.g., Q-learning [38], unsupervised learning, e.g., clustering methods [15], supervised learning, e.g., decision tree [23] and neural network (NN) models [25, 10, 1, 14], and statistical learning, e.g., k-nearest-neighbors(kNN) [11].

SVMs are recently proposed to model the non-linear relationship of the data. They have attracted the interests of researchers due to the following reasons. First, SVMs are grounded on the VC theory, which claims to guarantee the generalization [35, 36]. Second, SVMs were proposed to solve the classification problem in the beginning. The margin maximization has visual geometric interpretation [35, 2]. Third, training SVM leads to solve the Quadratic Programming (QP) problem. For any convex programming problem, every local solution will also be global. Therefore, SVM training always finds a global solution, which is usually a unique solution [5]. Fourth, SVMs can tackle the non-linear cases by introducing the kernel function [17]. Here, our work just concentrate on the regression model, Support Vector Regression.

3 SVR with Momentum

In this section, we will give a brief introduction of Support Vector Regression with a general ε-insensitive loss function. Then we will describe the concept of momentum for the margin setting in Sect. 3.2.

3.1 SVR with a General ε-insensitive Loss Function

Usually, a regression problem is to estimate (learn) a function

$$f(\mathbf{x}, \boldsymbol{\lambda}): \quad X(\mathbb{R}^d) \to \mathbb{R},$$

where $\boldsymbol{\lambda} \in \Lambda$, Λ is a set of abstract parameters, from a set of independent identically distributed (i.i.d.) samples with size N,

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N), \quad \mathbf{x}_i \in X(\mathbb{R}^d), \quad y_i \in \mathbb{R}, \tag{1}$$

where the above samples are drawn from an unknown distribution $P(\mathbf{x}, y)$.

Now the aim is to find a function $f(\mathbf{x}, \boldsymbol{\lambda}^*)$ with the smallest possible value for the *expected risk* (or test error) as

$$R[\boldsymbol{\lambda}] = \int l(y, f(\mathbf{x}, \boldsymbol{\lambda}))P(\mathbf{x}, y)d\mathbf{x}dy, \tag{2}$$

where l is a loss function which can be defined as one needs.

However, the probability of distribution $P(\mathbf{x}, y)$ is usually unknown. We are unable to compute, and therefore to minimize the expected risk $R[\boldsymbol{\lambda}]$ in (2), but we may know some information of $P(\mathbf{x}, y)$ from the samples in (1), we can compute a stochastic approximation of $R[\boldsymbol{\lambda}]$ by the so called *empirical risk*:

$$R_{emp}[\boldsymbol{\lambda}] = \frac{1}{N}\sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i, \boldsymbol{\lambda})). \tag{3}$$

This is because of that the law of large numbers guarantees that the empirical risk converges in probability to the expected risk. However, for practical problem, the size of samples is small. Only minimizing the empirical risk may cause problems, such as bad estimation or overfitting, and we cannot obtain good result when new data come in.

To solve the small sample problem, the statistical theory, or VC theory, has provided bounds on the deviation of the empirical risk from the expected risk [34, 36]. A typical uniform Vapnik and Chervonenkis bound, which holds with probability $1 - \eta$, has the following form:

$$R[\boldsymbol{\lambda}] \leq R_{emp}[\boldsymbol{\lambda}] + \sqrt{\frac{h(\ln\frac{2N}{h}+1) - \ln\frac{\eta}{4}}{N}}, \quad \forall \boldsymbol{\lambda} \in \Lambda, \tag{4}$$

where h is the VC-dimension of $f(\mathbf{x}, \boldsymbol{\lambda})$.

From this bound, it is clear that in order to achieve small expected risk, i.e., good generalization performance, both the empirical risk and the ratio between the VC-dimension and the number of samples has to be small. Since the empirical risk is usually a decreasing function of h, it turns out that for a given number of samples, there is an optimal value of the VC-dimension. The choice of an appropriate value of h (which in most techniques is controlled by the number of free parameters of the model) is very important in order to get good performance, especially when the number of samples is small.

Therefore, a different induction principle, *Structural Risk Minimization Principle*, was proposed and developed by Vapnik [34, 35, 36] in the attempt to overcome the problem of choosing an appropriate VC-dimension.

SVMs were developed to implement the SRM principle [35]. They were used in the classification at first [3]; they were also extended to solve the regression problem [35]. When SVMs were used to solve the regression problem, they were usually called Support Vector Regression (SVR). The aim of SVR is to find a function f with parameters $\mathbf{w}$ and b by minimizing the following regression risk:

$$R_{reg}(f) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C\sum_{i=1}^{N} l(f(\mathbf{x}_i), y_i), \tag{5}$$

where $\langle , \rangle$ denotes the inner product, the first term can be seen as the margin in SVMs and therefore can measure the VC-dimension [35]. A common interpretation is that the Euclidean norm, $\langle \mathbf{w}, \mathbf{w} \rangle$, measures the flatness of the function f, minimizing $\langle \mathbf{w}, \mathbf{w} \rangle$ will make the objective function as flat as possible [30].

The function f is defined as

$$f(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle + b, \tag{6}$$

where $\boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \to \Omega$, maps $\mathbf{x} \in X(\mathbb{R}^d)$ into a high (possible infinite) dimensional space Ω, and $b \in \mathbb{R}$.

In general, ε-insensitive loss function is used as the loss function [35, 30]. For this function, when the samples are in the range of $\pm\varepsilon$, they do not contribute to the output error. Thus, it leads to the sparseness of the solution. The function is defined as

$$\Gamma(f(\mathbf{x}) - y) = \begin{cases} 0, & \text{if} \quad |y - f(\mathbf{x})| < \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}. \tag{7}$$

In [40], we have introduced a general ε-insensitive loss function, $\Gamma'(f(\mathbf{x}_i) - y_i), i = 1, \ldots, N$, which is given as

$$\begin{cases} 0, & \text{if } -d(\mathbf{x}_i) < y_i - f(\mathbf{x}_i) < u(\mathbf{x}_i) \\ y_i - f(\mathbf{x}_i) - u(\mathbf{x}_i), & \text{if } y_i - f(\mathbf{x}_i) \geq u(\mathbf{x}_i) \\ f(\mathbf{x}_i) - y_i - d(\mathbf{x}_i), & \text{if } f(\mathbf{x}_i) - y_i \geq d(\mathbf{x}_i) \end{cases}, \tag{8}$$

where $d(\mathbf{x}), u(\mathbf{x}) \geq 0$, are two functions to determine the down margin and up margin respectively.

When $d(\mathbf{x})$ and $u(\mathbf{x})$ are both constant functions and $d(\mathbf{x}) = u(\mathbf{x})$, equation (8) amounts to the ε-insensitive loss function in (7) and we labeled it as FASM (Fixed and Symmetrical Margin). When $d(\mathbf{x})$ and $u(\mathbf{x})$ are both constant functions but $d(\mathbf{x}) \neq u(\mathbf{x})$, this case is labeled as FAAM (Fixed and Asymmetrical Margin). In the case of NASM (Non-fixed and Symmetrical Margin), $d(\mathbf{x}) = u(\mathbf{x})$ and they are varied with the data. The last case is with a non-fixed and asymmetrical margin(NAAM), where $d(\mathbf{x})$ and $u(\mathbf{x})$ are varied with the data and $d(\mathbf{x}) \neq u(\mathbf{x})$.

After using the standard method to find the solution of (5) with the loss function of (8) as [35], we obtain $\mathbf{w} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \boldsymbol{\phi}(\mathbf{x}_i)$, by solving the following Quadratic Programming (QP) problem:

$$\begin{aligned} \min Q(\boldsymbol{\alpha}, \boldsymbol{\alpha^*}) = {} & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle \\ & + \sum_{i=1}^{N} (u(\mathbf{x}_i) - y_i)\alpha_i + \sum_{i=1}^{N} (d(\mathbf{x}_i) + y_i)\alpha_i^*, \end{aligned}$$

subject to

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0, C], \tag{9}$$

where α_i and α_i^* are corresponding Lagrange multipliers used to push and pull $f(\mathbf{x}_i)$ towards the outcome of y_i respectively.

The above QP problem is very similar to the original QP problem in [35], therefore, it is easy to modify the previous algorithm to implement this QP problem. Practically, we implement our QP problem by modifying the libSVM from [7] with adding a new data structure to store both margins: up margin, $u(\mathbf{x})$, and down margin, $d(\mathbf{x})$. Obviously, this will not impact the time complexity of the SVR algorithm; we just need more space, linear to the size of data points, to store the corresponding margins.

Furthermore, using a kernel function, the estimation function in (6) becomes

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \kappa(\mathbf{x}, \mathbf{x}_i) + b, \tag{10}$$

where the kernel function, $\kappa(\mathbf{x}, \mathbf{x}_i) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i) \rangle$, is a symmetric function and satisfies the Mercer's condition. In this work, we select a common kernel function, RBF function, as the kernel function,

$$\kappa(\mathbf{x}, \mathbf{x}_i) = \exp(-\beta \|\mathbf{x} - \mathbf{x}_i\|^2), \tag{11}$$

where β is the kernel parameter.

In the following, we exploit the Karush-Kuhn-Tucker (KKT) conditions to calculate b. Here, they are

$$\alpha_i(u(\mathbf{x}_i) + \xi_i - y_i + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle + b) = 0,$$
$$\alpha_i^*(d(\mathbf{x}_i) + \xi_i^* + y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle - b) = 0,$$

and

$$(C - \alpha_i)\xi_i = 0,$$
$$(C - \alpha_i^*)\xi_i^* = 0.$$

Therefore, when there exists i, such that $\alpha_i \in (0, C)$ or $\alpha_i^* \in (0, C)$, b can be computed as follows:

$$b = \begin{cases} y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle - u(\mathbf{x}_i), \text{ for } \alpha_i \in (0, C) \\ y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle + d(\mathbf{x}_i), \text{ for } \alpha_i^* \in (0, C) \end{cases}.$$

When no $\alpha_i^{(*)} \in (0, C)$, the average method [7] is used.

3.2 Momentum

Momentum is a well known term in physics. We borrow this term in the work. The differences are: in physics, momentum is used to measure the change of state of a body by external forces; in our work, the momentum is used to measure the up and down trend of stock market, which is impelled by the investors. The term in both areas reflects the difference of change, but with different kinds of external forces: in our work, the external forces are the investment of investors.

More specifically, we construct a margin setting, which is a linear combination of the standard deviation and the momentum. The up margin and down margin are set in the following forms:

$$u(\mathbf{x}_i) = \lambda_1 \times \sigma(\mathbf{x}_i) + \mu \times \Delta(\mathbf{x}_i), \quad i = 1, \ldots, N,$$
$$d(\mathbf{x}_i) = \lambda_2 \times \sigma(\mathbf{x}_i) - \mu \times \Delta(\mathbf{x}_i), \quad i = 1, \ldots, N, \tag{12}$$

where $\sigma(\mathbf{x}_i)$ is the standard deviation of input $\mathbf{x}_i$, $\Delta(\mathbf{x}_i)$ is the momentum at point $\mathbf{x}_i$, λ_1, λ_2 are both positive constants, called coefficients of the margin width and μ is a non-negative constant, called coefficient of momentum. Using this margin setting (12), the width of margin at point $\mathbf{x}_i$ is determined by $\sigma(\mathbf{x}_i)$ and the sum of λ_1 and λ_2, i.e.,

$$W(\mathbf{x}_i) = (\lambda_1 + \lambda_2) \times \sigma(\mathbf{x}_i).$$

The standard deviation here is used to reflect the change of volatility; therefore, when in a high volatility mode, we use a broad width of margin; when in a low volatility situation, we use a narrow width of margin.

For the setting of momentum, in fact, there are many ways to calculate it. For example, it may be set as a constant. In this work, we exploit the Exponential Moving Average (EMA), which is time-varying and can reflect the up trend and down tendency of the financial data. An n-day's EMA is calculated by

$$EMA_i = EMA_{i-1} \times (1 - r) + y_i \times r,$$

where $r = 2/(1 + n)$ and it begins from the first day, $EMA_1 = y_1$. Here, n is called the length of EMA. The current day's momentum is set as the difference between the current day's EMA and the EMA in the previous k day, i.e.,

$$\Delta(x_i) = EMA_i - EMA_{i-k} \tag{13}$$

where k is called the lag of momentum. Equation (13) actually detects the degree of the change in the stock market.

From above configurations, we know that the margin setting of (12) includes the case of NASM (when $\mu = 0$). When $\mu \neq 0$, it is the case of NAAM. If $\Delta(\mathbf{x}) > 0$, we know that an up trend occurs. Based on our downside risk avoiding predictive strategy, we would use a larger up margin and a smaller down margin to under-predict the stock price. While if $\Delta(\mathbf{x}) < 0$, i.e., in the situation of down trend, we would use a smaller up margin and larger down margin to over-predict the stock price.

In addition, in the margin setting of (12) and momentum setting of (13), we have to specify the concrete setting of parameters. For the coefficients of margin width, λ_1 and λ_2, they are set to $\frac{1}{2}$; therefore, we can make the margin width at day i equal to the standard deviation of input $\mathbf{x}_i$. For the coefficient of momentum, μ, it is equal to 1; the lag of momentum, k, is equal to 1. The setting of these two parameters is coming from our experience in [39]. Actually, the only undetermined parameter is the length of EMA, n. In the following experiments, we use different length of EMA to test their effects and we find that it is related to the volatility of financial data.

4 Experiments

In this section, we first define the performance measurement of our experiments. Then we detail the setup of experiments with their results compared.

4.1 Accuracy Metrics

We use the following statistical metrics to evaluate the prediction performance, including Mean Absolute Error (MAE), Up side Mean Absolute Error (UMAE), and Down side Mean Absolute Error (DMAE). The definitions of these criteria are listed in the Table 1. MAE is the measure of the discrepancy between the actual and predicted values. The smaller the MAE, the closer are the predicted values to the actual values. UMAE is the measure of up side risk. DMAE is the measure of down side risk. The smaller the UMAE and DMAE, the smaller are the corresponding predictive risks.

Table 1. Accuracy metrics

Metrics	Calculation
MAE	$\text{MAE} = \frac{1}{m} \times \sum_{i=1}^{m} \lvert a_i - p_i \rvert$
UMAE	$\text{UMAE} = \frac{1}{m} \times \sum_{i=1, a_i \geq p_i}^{m} (a_i - p_i)$
DMAE	$\text{DMAE} = \frac{1}{m} \times \sum_{i=1, a_i < p_i}^{m} (p_i - a_i)$

a_i and p_i are the actual values and predicted values at day i respectively.
m is the number of testing data.

4.2 Experimental Procedure and Results

In this section, we conduct the SVR algorithm with four categories of margin settings, Autoregressive (AR) model with order four and RBF network on two indices respectively and compare their results.

Two indices are used in the experiments:

1) HSI: daily closing prices of Hong Kong's Hang Seng Index from January 2nd, 1998 to December 29$^{\text{th}}$, 2000.
2) DJIA: daily closing prices of Dow Jones Industrial Average from January 2nd, 1998 to December 29$^{\text{th}}$, 2000.

The ratio of the number of training data and the number of testing data is set to 5:1. Therefore, the corresponding training time periods and testing periods are obtained and listed in Table 2.

Furthermore, we model the system as $p_t = f(\mathbf{x}_t)$, where f is learned by the stated three models: SVR, AR and RBF network, from the training data; $\mathbf{x}_t = (a_{t-4}, a_{t-3}, a_{t-2}, a_{t-1})$, a_t is the daily closing index in day t, an intrinsic assumption here is that there is (non)linear relationship between sequential five days' index values. After finding the function f, we use the testing data to test the predictive performance of the models.

The experiments are conducted on Sun Blade 1000, RAM 2GB and Solaris 8.

Table 2. Indices, time periods and parameters

Indices	Training time periods	Testing time periods	C	β
HSI	02/01/1998 - 04/07/2000	05/07/2000 - 29/12/2000	2^6	2^{-3}
DJIA	02/01/1998 - 29/06/2000	30/06/2000 - 29/12/2000	2^{-1}	2^1

4.2.1 SVR Algorithm

Before generating the model, we perform a cross-validation on the training data to determine the parameters that are needed in SVR. They are C, the cost of error and β, the parameter of kernel function. The parameters we used are listed in Table 2.

4.2.1.1 NASM and NAAM

The margins setting is based on (12). More specifically, in the case of NASM, we set $\lambda_1 = \lambda_2 = \frac{1}{2}$ and $\mu = 0$, thus the overall margin widths are equal to the standard deviation of input $\mathbf{x}$. In the case of NAAM, we also fix $\lambda_1 = \lambda_2 = \frac{1}{2}$; therefore, we have a fair comparison of NASM case. From our experience [39], $k = 1$ and $\mu = 1$ are suitable for different data sets. The uncertain term for the margin setting is n, the length of EMA. Hence, we use different n, equal to 10, 30, 50, 100 respectively to test the effect of the length of EMA. From the result of Table 3 and Table 4, we can see that the DMAE values in all cases of NAAM are smaller than that in NASM case, thus we have a smaller predictive downside risk in NAAM case. This also meets our assumption, i.e., it is a downside risk avoiding strategy for the prediction. We also see that the MAE gradually decreases with the length of EMA increases and when the length equals 100, the MAE is the smallest in all case of NAAM and is smaller than that of NASM. In Table 4, when the length equals 30, the MAE is the smallest in all cases of NAAM and it is also smaller than that of NASM.

Table 3. Effect of the length of EMA on HSI

type	n	MAE	UMAE	DMAE
NASM		217.18	108.95	108.23
	10	221.01	119.70	101.31
NAAM	30	218.32	123.56	94.76
	50	217.12	120.31	96.81
	100	**216.60**	120.60	96.00

Here, we plot the daily closing prices of HSI with 100-days' EMA and the prices of DJIA with 30-days' EMA in Fig. 3 and Fig. 4 respectively and list the Average Standard Deviations (ASD) of input $\mathbf{x}$ of the training data sets

Table 4. Effect of the length of EMA on DJIA

type	n	MAE	UMAE	DMAE
NASM		87.17	44.17	43.00
	10	86.61	43.79	42.81
NAAM	30	**86.58**	45.10	41.48
	50	87.36	47.02	40.34
	100	87.02	45.67	41.35

of both data sets, the Average of Absolute Momentums (AAM) of input $\mathbf{x}$ for the best length of both training data sets respectively in Table 5. We can observe that the ASD of HSI is higher than that of DJIA and the ratio of AAM to ASD is smaller for HSI than that for DJIA. This indicates that the data is more volatile in the HSI data; hence we may use a longer length of EMA to represent this volatility for the prediction.

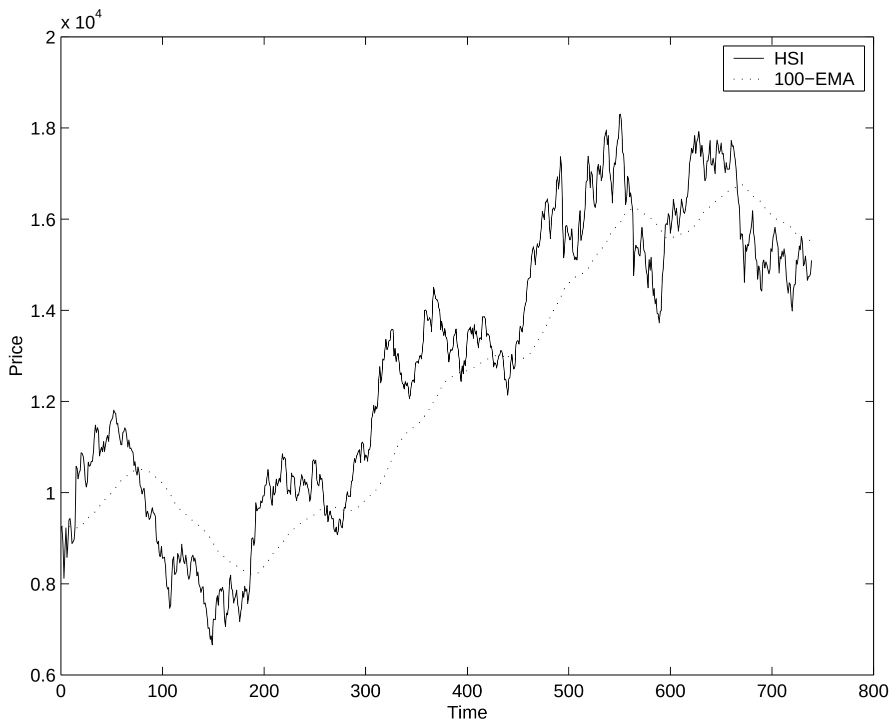

Fig. 3. HSI and 100 days' EMA

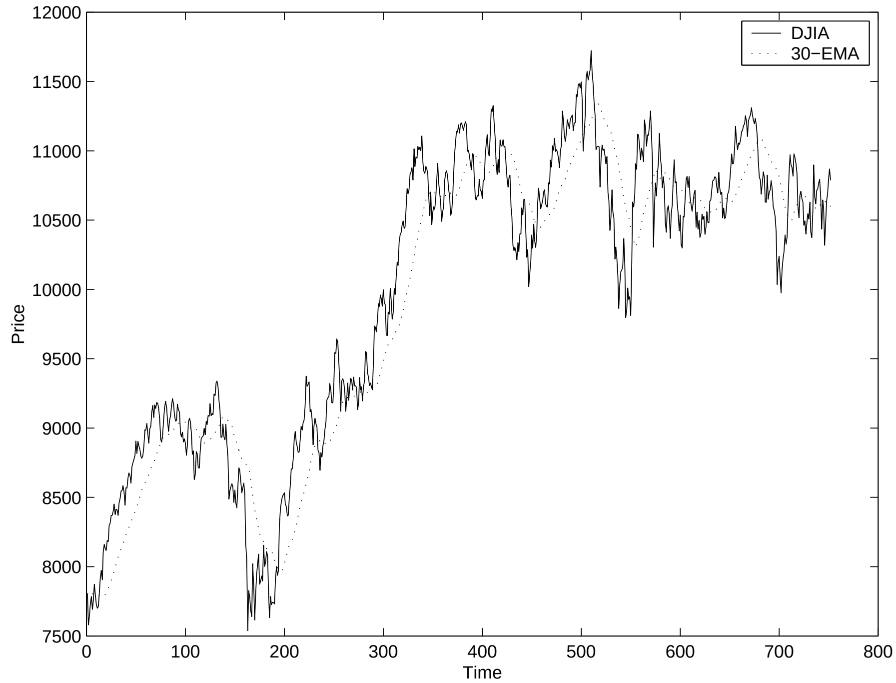

Fig. 4. DJIA and 30 days' EMA

Table 5. ASD and AAM

data set	ASD	AAM		ratio
		n	Δ	
HSI	182.28	100	20.80	0.114
DJIA	79.95	30	15.64	0.196

4.2.1.2 FASM and FAAM

For the fixed margin setting, we set the margin width to 0.03, i.e. $u(\mathbf{x})+d(\mathbf{x}) = 0.03$, for both data sets. The up margin, $u(\mathbf{x})$, ranges from 0 to 0.03, each increments is 0.0075. For these setting, we obtain the results in Table 6 for data set HSI and in Table 7 for data set DJIA. Comparing the corresponding results of non-fixed margin settings (Table 3 and Table 4) with the results of fixed margin settings (Table 6 and Table 7), we observe that the predictive performance of non-fixed margin settings is better than that of the fixed margin cases generally. From Table 6 and Table 7, we can see that the MAE is in a wide range. This means that using a fixed margin setting may have bad predictive result which gives more risk.

Table 6. Fixed margin results on HSI

$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
0	0.03	259.32	43.37	215.95
0.0075	0.0225	233.28	66.21	167.07
0.0150	0.0150	220.50	94.07	126.43
0.0225	0.0075	216.87	126.96	89.91
0.03	0	227.17	167.34	59.83

Table 7. Fixed margin results on DJIA

$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
0	0.03	99.97	17.00	82.97
0.0075	0.0225	90.42	25.24	65.18
0.0150	0.0150	86.70	35.46	51.24
0.0225	0.0075	87.61	48.47	39.14
0.03	0	93.24	64.30	29.94

4.2.2 AR Model

Here, we use the AR model with order four to predict the prices of HSI and DJIA; hence, we can compare the AR model with NASM, NAAM in SVR with the same order. The results are listed in the Table 8. We can see that NASM and NAAM are superior to AR model with same order.

Table 8. Results on AR(4)

data set	MAE	UMAE	DMAE
HSI	217.75	105.96	111.79
DJIA	88.74	46.36	42.38

4.2.3 RBF network

The RBF network we used is implemented in NETLAB [19]. We perform the one-step ahead prediction to predict the prices of HSI and DJIA. More specifically, we set the effect of hidden units to 3, 5, 7, 9 and set other parameters as default. The corresponding results are listed in Table 9 for HSI, in Table 10 for DJIA respectively. Comparing these two tables with Table 3 and Table 4, we can see that NASM and NAAM are also better than the RBF network.

Table 9. Effect of number of hidden units on HSI

# hidden	MAE	UMAE	DMAE
3	386.65	165.08	221.57
5	277.83	128.92	148.91
7	219.32	104.15	115.17
9	221.81	109.46	112.35

Table 10. Effect of number of hidden units on DJIA

# hidden	MAE	UMAE	DMAE
3	88.31	44.60	43.71
5	98.44	48.46	49.98
7	90.53	46.22	44.31
9	87.23	44.09	43.14

5 Discussion and Conclusion

In this work, we propose to use non-fixed and asymmetrical margin (NAAM) setting in the prediction of HSI and DJIA. From the experiments, we make the following observations:

1. Comparing NAAM with the case of NASM which just uses the standard deviation, we find that by adding the momentum to set the margin we can reduce the predictive downside risk. We may also improve the accuracy of our prediction by selecting a suitable length of EMA.
2. The selection of the length of EMA may depend on the volatility of the financial data. A long term EMA may be suitable for a higher volatility financial time series. A short term EMA may be suitable for the opposite case.
3. In SVR, non-fixed margin settings (NAAM and NASM) are better than the fixed margin settings (FAAM and FASM). Using a fixed margin setting may have more risk, which results in poor performance.
4. The SVR algorithm with NASM and NAAM outperforms the AR model with the same order.
5. The SVR algorithm with NASM and NAAM is also better than the RBF network.

In our work, how to find more suitable parameters easily, i.e., C and β, for a specific data set is still a problem. In addition, we just consider the momentum term to trace the changing trend of the stock market here. Other more general or robust methods are still needed to explore and to apply in the margin settings to capture the information of stock market promptly.

Acknowledgement

The work described in this work was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region, China.

References

1. D. E. Baestaens. *Neural Network Solutions for Trading in Financial Markets.* London: Financial Times: Pitman Pub., 1994.
2. K. Bennett and E. Bredensteiner. Duality and Geometry in SVM Classifiers. In P. Langley, editor, *Proc. of Seventeenth Intl. Conf. on Machine Learning*, pages 57–64, San Francisco, 2000. Morgan Kaufmann.
3. B. E. Boser, I. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Computational Learing Theory*, pages 144–152, 1992.
4. G. E. P. Box and G. M. Jenkins. *Time-Series Analysis, Forecasting and Control.* San Francisco: Holden-Day, third edition, 1994.
5. C. Burges and D. Crisp. Uniqueness of the SVM Solution. In S. A. Solla, T. K. Leen, and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 223–229, Cambridge, MA, 2000. MIT Press.
6. Li Juan Cao, Kok Seng Chua, and Lim Kian Guan. c-Ascending Support Vector Machines for Financial Time Series Forecasting. In *International Conference on Computational Intelligence for Financial Engineering (CIFEr2003)*, pages 329–335, 2003.
7. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a Library for Support Vector Machines (version 2.31), 2001.
8. C. Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, fifth edition, 1996.
9. C. Chatfield. *Time-Series Forecasting.* Chapman and Hall/CRC, 2001.
10. B. Cheng and D. M. Titterington. Neural Networks: A Review from a Statistical Perspective. *Statistical Science*, 9:2–54, 1994.
11. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* New York: Wiley, London; New York, 1973.
12. C. W. J. Granger and A. P. Andersen. *Introduction to Bilinear Time Series.* Göttingen: Vandenhoeck and Ruprecht, 1978.
13. C. W. J. Granger and R. Joyeux. An Introduction to Long-Memory Time Series Models and Fractional Differencing. *Journal of Time Series Analysis*, 1, 1980.
14. S. Haykin. *Neural Networks : A Comprehensive Foundation.* Upper Saddle River, N. J.: Prentice Hall, 2nd edition, 1999.
15. A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.
16. S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines. In J. Principe, L. Giles, N. Morgan, and E. Wilson, editors, *IEEE Workshop on Neural Networks for Signal Processing VII*, pages 511–519. IEEE Press, 1997.
17. K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schö1kopf. An introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, 2001.

18. K. R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting Time Series with Support Vector Machines. In W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, editors, *ICANN*, pages 999–1004. Springer, 1997.
19. Ian T. Nabney. *Netlab: Algorithms for Pattern Recognition.* Springer, London; New York, 2002.
20. D. F. Nicholls and A. Pagan. Varying Coefficient Regression. In E.J. Hannan, P.R. Krishnaiah, , and M.M. Rao, editors, *Handbook of Statistics*, volume 5, pages 413–449, North Holland, Amsterdam, 1985.
21. B. Schölkopf Pai-Hsuen Chen, Chih-Jen Lin. A Tutorial on ν-Support Vector Machines. Technical report, National Taiwan University, 2003.
22. M. B. Priestley. *Spectral Analysis and Time Series.* New York: Academic Press, London, 1981.
23. J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.
24. Baldev Raj and Aman Ullah. *Econometrics: A Varying Coefficients Approach.* New York: St. Martin's Press, 2nd edition, 1981.
25. B. D. Ripley. Statistical Aspects of Neural Networks. In O. E. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall, editors, *Network and Chaos – Statistical and Probablistic Aspects*, pages 40–123, London, 1993. Chapman and Hall.
26. B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Support Vector Regression with Automatic Accuracy Control. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of ICANN'98 Perspectives in Neural Computing*, pages 111–116, Berlin, 1998. Spring.
27. B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the Tube: A New Support Vector Regression Algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 330 – 336, Cambridge, MA, 1999. MIT Press.
28. B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New Support Vector Algorithms. Technical Report NC2-TR-1998-031, GMD and Australian National University, 1998.
29. A. Smola, N. Murata, B. Schölkopf, and K.-R. Müller. Asymptotically Optimal Choice of ε-Loss for Support Vector Machines. In *Proc. of Seventeenth Intl. Conf. on Artificial Neural Networks*, 1998.
30. A. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.
31. E. H. Tay and L. J. Cao. Application of Support Vector Machines to Financial Time Series Forecasting. *Omega*, 29:309–317, 2001.
32. H. Tong. *Non-Linear Time Series.* Clarendon Press, Oxford, 1990.
33. T. B. Trafalis and H. Ince. Support Vector Machine for Regression and Applications to Financial Forecasting. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000)*, volume 6, pages 348–353. IEEE, 2000.
34. V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data.* (in Russian), Nauka, Moscow, 1979.
35. V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.
36. V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.
37. V. N. Vapnik, S. Golowich, and A. Smola. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In M. Mozer,

M. Jordan, and T. Petshe, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 281–287, Cambridge, MA, 1997. MIT Press.

38. C. Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, England, 1989.
39. Haiqin Yang. Margin Variations in Support Vector Regression for the Stock Market Prediction. Master's thesis, Chinese University of Hong Kong, 2003.
40. Haiqin Yang, Laiwan Chan, and Irwin King. Support Vector Machine Regression for Volatile Stock Market Prediction. In Hujun Yin, Nigel Allinson, Richard Freeman, John Keane, and Simon Hubbard, editors, *Intelligent Data Engineering and Automated Learning — IDEAL 2002*, volume 2412 of *LNCS*, pages 391–396. Springer, 2002.