2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Local Support Vector Regression For Financial Time Series Prediction

Kaizhu Huang, Haiqin Yang, Irwin King, and Michael R. Lyu

***Abstract*— We consider the regression problem for financial time series. Typically, financial time series are non-stationary and volatile in nature. Because of its good generalization power and the tractability of the problem, the Support Vector Regression (SVR) has been extensively applied in financial time series prediction. The standard SVR adopts the ℓ_p-norm ($p = 1$ or 2) to model the functional complexity of the whole data set and employs a fixed ϵ-tube to tolerate noise. Although this approach has proved successful both theoretically and empirically, it considers data in a global fashion only. Therefore it may lack the flexibility to capture the local trend of data; this is a critical aspect of volatile data, especially financial time series data. Aiming to address this issue, we propose the Local Support Vector Regression (LSVR) model. This novel model is demonstrated to provide a systematic and automatic scheme to adapt the margin locally and flexibly; the margin is fixed globally in the standard SVR. Therefore, the LSVR can tolerate noise adaptively. We provide both theoretical justifications and empirical evaluations for this novel model. The experimental results on synthetic data and real financial data demonstrate its advantages over the standard SVR.**

I. Introduction

We consider the regression or prediction problem for financial time series data in this paper. The objective is to learn a model from a given financial time series data set, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, and then use the learned model to make accurate predictions of y for future values of $\mathbf{x}$. The Support Vector Regression (SVR), a successful method in dealing with this problem, is well suited to generalization [8]. The standard SVR adopts the ℓ_p-norm ($p = 1$ or 2) to control the functional complexity and chooses an ϵ-insensitive loss function with a fixed tube (margin) to measure the empirical risk. By introducing the ℓ_p-norm, the optimization problem in SVR can be transformed to a tractable programming problem, in particular a quadratic programming problem when $p = 2$. Furthermore, the ϵ-tube has the ability to tolerate noise in data, and fixing the tube confers the advantage of simplicity. Although these settings are effective in common applications, they are designed in a global fashion and lack the flexibility to capture the local trend in some applications, in particular in stock markets data or financial time series. In the context of financial time series prediction, the data are usually highly volatile and the associated variance of noise varies over time. In such domains, fixing the tube cannot capture the local trend of data and cannot tolerate noise adaptively.

One typical illustration can be seen in Figure 1. In this figure, the data become more noisy as the x value of the data increases. As shown in Figure 1(a), with a fixed ϵ-margin (set to 0.04 in this example), SVR considers the data globally and equally: The derived approximating function in SVR deviates from the actual data trend. On the other hand, as illustrated in Figure 1(b), if we address the local volatility of the data by adaptively and automatically setting a small margin in low-volatile regions and a large margin in high-volatile regions, the resulting approximating function (the blue solid line in Figure 1(b)) is more appropriate and reasonable.

In order to address this issue, we propose the Local Support Vector Regression (LSVR) model. We will show that, by taking the local data trend into consideration, our model provides a systematic and automatic scheme to adapt the margin locally and flexibly. Moreover, we will demonstrate that this novel LSVR model has extensive connections with other models. Specifically, this model can be seen as an extension of a recently-proposed general large margin classifier, the Maxi-Min Margin Machine (M^4) [2], for regression tasks; it can also yield a special case, which will be proven to be equivalent with the standard SVR under certain mild assumptions. One critical feature of our model is that the associated optimization of LSVR can be relaxed as a Second Order Conic Programming (SOCP) problem, which can be efficiently solved in polynomial time [5]. Another appealing feature is that kernelization is also applicable to the LSVR model. Therefore, the proposed LSVR can generate non-linear approximating functions and hence can be applied to more general regression tasks.

The rest of this paper is organized as follows. In Section II, we review the standard Support Vector Regression. The linear LSVR model, including its model definition, appealing features, and optimization method, is described in Section III. In Section IV, we demonstrate how the LSVR model can be linked with other models including M^4 and SVR. The kernelized LSVR is tackled in Section V. In Section VI, we present the result of experiments using both synthetic and real financial data. Finally, we set out the conclusion and propose future work in Section VII.

II. Support Vector Regression

We define a training data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in X, y_i \in \mathbb{R}$, N is the number of training data points, and X denotes the space of the input samples $\mathbb{R}^n$. The aim is to find a function

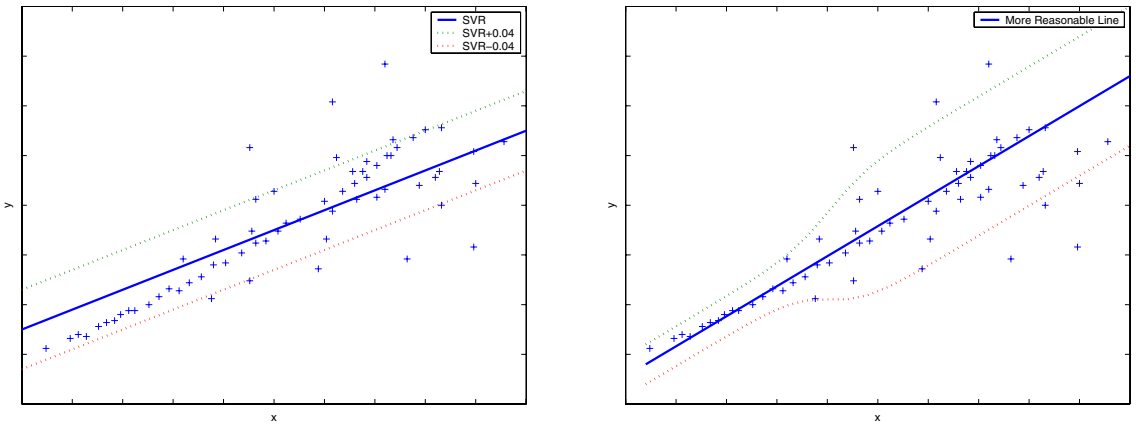
Kaizhu Huang is with Information Technology Laboratory, Fujitsu Research and Development Center Co.Ltd. E-Mail: kzhuang@frdc.fujitsu.com.

Haiqin Yang is with Titanium Technology Limited, Shenzhen, China. E-Mail:austin.yang@titanium-tech.com

Irwin King is with Department of Computer Science and Engineering, the Chinese University of Hong Kong. E-Mail: king@cse.cuhk.edu.hk.

Michael R. Lyu is with Department of Computer Science and Engineering, the Chinese University of Hong Kong. E-Mail: lyu@cse.cuhk.edu.hk.

0-7803-9490-9/06/$20.00/©2006 IEEE



(a) (b)

Fig. 1. Illustration of the ϵ-insensitive loss function with fixed and non-fixed margins in the feature space. In (b), a non-fixed margin setting is more reasonable. It can moderate the effect of the noise by enlarging (shrinking) the margin width in the local area with large (small) variance of noise.

which can not only approximate these data well, but also can predict the value of y for future data $\mathbf{x}$ accurately.

In general, the approximating function in SVR takes the following linear form, $f(x) = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w} \in \mathbb{R}^n$ and $b \in R$. Furthermore, the above linear regression model can be extended into the non-linear one by using Mercer's kernel. Now the question is to determine $\mathbf{w}$ and b from the training data by minimizing the regression risk, $R_{reg}(f) = \Omega[f] + C\sum_{i=1}^{N}\Gamma(f(\mathbf{x}_i) - y_i)$, where $\Omega[f]$ is the structure risk, used to control the smoothness or complexity of the function, $\Gamma(\cdot)$ is a cost function that measures the empirical risk, and C is a pre-specified trade-off value. Generally, in SVR, $\Omega[f]$ takes the form of $\|\mathbf{w}\|$ in l_1-SVR or $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ in l_2-SVR. The empirical cost function adopts the form of an ε-insensitive loss function [8], which is defined as follows: $\Gamma(f(\mathbf{x})-y) = \begin{cases} 0, & \text{if} \quad |y - f(\mathbf{x})| < \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}$. In this function, when the data points are in the range of $\pm\varepsilon$, they do not contribute to the empirical error.

The complete optimization of SVR (or more precisely, the optimization of l_1-SVR) can be written as follows:

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad \|\mathbf{w}\| + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{1}$$

$$\text{s.t.} \quad y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon + \xi_i, \tag{2}$$

$$(\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon + \xi_i^*, \tag{3}$$

$$\xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N, \tag{4}$$

where ξ_i and ξ_i^* are the corresponding positive and negative errors at the i-th point, respectively. This optimization problem can be solved by the Linear Programming method. When the structure risk term $\Omega[f]$ takes the form of $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ (as in l_2-SVR), the optimization becomes a Quadratic Programming problem.

In the above optimization problem, the standard SVR fixes the margin ϵ globally for all data points. Although this simple setting achieves great success in many tasks, it lacks the flexibility to capture the data's volatility, which is a typical feature of financial time series data. In order to address this problem, we therefore develop the novel Local Support Vector Regression model.

III. Local Support Vector Regression Model

In this section, we first present the definition of the LSVR model. We then detail its interpretation and its appealing characteristics. After that, we state its corresponding optimization method.

A. Model Definition

The objective of the LSVR model is to learn the linear approximating function in $\mathcal{D}$ by making the function locally as involatile as possible while keeping the error as small as possible. We formulate this objective as follows:

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad \frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{5}$$

$$\text{s.t.} \quad y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i,$$

$$(\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, \tag{6}$$

$$\xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N,$$

where ξ_i, ξ_i^*, and ϵ are defined as in the previous section. Σ_i is the covariance matrix formed by the i-th data point and those data points close to it.

B. Interpretations and Appealing Properties

In this section, we interpret our novel LSVR model. First, we discuss the physical meaning of the term $\mathbf{w}^T\Sigma_i\mathbf{w}$. Suppose $y_i = \mathbf{w}^T\mathbf{x}_i + b$ and $\bar{y_i} = \mathbf{w}^T\bar{\mathbf{x}_i} + b$ ($\bar{\mathbf{x}_i}$ denotes the mean of $\mathbf{x}_i$ and a certain number of points closest to it). We have the variance around the i-th data point as $\Delta_i = \frac{1}{2k+1}\sum_{j=-k}^{k}(y_{i+j} - \bar{y_i})^2 = \frac{1}{2k+1}\sum_{j=-k}^{k}(\mathbf{w}^T(\mathbf{x}_{i+j} - \bar{\mathbf{x}_i}))^2 = \mathbf{w}^T\Sigma_i\mathbf{w}$, where $2k$ is the number of data points closest to the i-th data point. Therefore, $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$

actually captures the volatility in the local region around the i-th data point. In addition, Δ_i can also measure the complexity of the function around the i-th data point, since it reflects the smoothness in the corresponding local region.

By using the first interpretation of $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$ (representing the local volatility), LSVR can systematically and automatically vary the margin: If the i-th data point lies in an area with a larger variance of noise, it will contribute to a larger $\epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ or a larger local margin, resulting in a reduction of the impact of the noise around the point. On the other hand, if the i-th data point is in the region with a smaller variance of noise, the local margin, $\epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$, will be smaller; in this case, the corresponding point will contribute more in the fitting process. By contrast, the standard SVR adopts a fixed margin, which treats each point equally and therefore lacks the ability to tolerate variations of noise.

By applying the second interpretation of $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$, namely, a measure describing the local functional complexity, LSVR controls the overall smoothness of the approximating function by minimizing the average of Δ_i, as seen in (5). In contrast, the standard SVR globally minimizes a complexity term, i.e., $\|\mathbf{w}\|$ or $\frac{1}{2}\mathbf{w}^T\mathbf{w}$, which is insensitive to local changes in the complexity of the function.

C. Optimization Method

In order to solve the optimization problem of (5), we introduce auxiliary variables, $t_1,\ldots,t_N$, and transform the problem as follows:

$$\begin{aligned}\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i+\xi_i^*),\\ \text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i+b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i, \quad (7)\\ & (\mathbf{w}^T\mathbf{x}_i+b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, \quad (8)\\ & \sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \le t_i,\\ & t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1,\ldots,N.\end{aligned}$$

It is clear that (7) and (8) are non-convex constraints. This may present difficulties in optimizing the LSVR problem. In the following, we relax the optimization to a Second Order Conic Programming problem (SOCP) problem [5] by replacing $\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ with its upper bound t_i.

$$\begin{aligned}\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i+\xi_i^*),\\ \text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i+b) \le \epsilon t_i + \xi_i,\\ & (\mathbf{w}^T\mathbf{x}_i+b) - y_i \le \epsilon t_i + \xi_i^*,\\ & \sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \le t_i,\\ & t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1,\ldots,N.\end{aligned}$$

Since t_i is closely related to $\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$, weighting the margin width with t_i will achieve the original objective, i.e., adapting the margin flexibly. Furthermore, the relaxed form is a linear programming problem under quadratic cone constraints, or more specifically it is a Second Order Conic Programming problem. Therefore, this problem can be solved in polynomial time by using many general optimization packages, e.g., Sedumi [9]. Another advantage is that the relaxation also enables the application of kernelization, which can yield more general non-linear approximating functions. This will be demonstrated in Section V.

We now analyze the time complexity of LSVR. As indicated in [5], if the SOCP is solved based on interior-point methods, it contains a worst-case complexity of $O(n^3)$. Adding the cost of forming the system matrix (constraint matrix), which is $O(Nn^3)$, the total complexity would be $O((N+1)n^3)$, which is in the same order as the Maxi-Min Margin Machine and can be solved in polynomial time. Note that for time series prediction, we do not need to use sorting methods to find the closest points for each data sample, since the series itself provides the order information. For example, $2k$ points closest to the i-th point are simply those data with time values $i-k, i-k+1,\ldots,i-1,i+1,\ldots,i+k$. Therefore no further computation is involved.

IV. Connections with Other Models

In this section, we establish various connections from our novel model to other models. We first show that the LSVR can be considered as the extension of the Maxi-Min Margin Machine to regression tasks. We then demonstrate how the standard SVR can be incorporated as a special case of LSVR.

A. Connection with Maxi-Min Margin Machine

The LSVR model can also be considered as an extension of the general large margin classifier, the Maxi-Min Margin Machine (M^4) [2]. Within the framework of binary classification for class X and Y, the M^4 model is formulated as follows:

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad \text{s.t.} \qquad (9)$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \ge \rho, \quad i = 1,2,\ldots,N_{\mathbf{x}}\,, \qquad (10)$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \ge \rho, \quad j = 1,2,\ldots,N_{\mathbf{y}}\,, \qquad (11)$$

where $\mathbf{\Sigma_x}$ and $\mathbf{\Sigma_y}$ refer to the covariance matrices of the X and the Y data, respectively. The M^4 model seeks to maximize the margin defined as the minimum Mahalanobis distance for all training samples,[1] while simultaneously classifying all the data correctly. This model has been shown to contain the Support Vector Machine, the Minimax Probability Machine [4], and the Fisher Discriminant Analysis as special cases. Furthermore, it can be linked with the Minimum Error Minimax Probability Machine (MEMPM) known as a worst-case distribution-free classifier [3].

Within the framework of classifications, M^4 considers different data trends for different classes, i.e., it adopts the covariance information of two classes of data, $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$. Analogously, in the novel LSVR model, we allow different

[1] This also inspired the name of this model.

data trends for different regions, which is more suitable for a regression application.

B. Connection with Support Vector Regression

We now analyze the connection of the LSVR model with the standard Support Vector Regression model. By considering the data trend globally and equally, i.e., setting $\Sigma_i = \Sigma$, for $i = 1, \ldots, N$, we can transform the optimization of (5) as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad & \sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \qquad (12) \\ \text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + \xi_i, \\ & (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + \xi_i^*, \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N, \end{aligned}$$

Further, if $\Sigma = \mathbf{I}$, we obtain:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad & \|\mathbf{w}\| + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \qquad (13) \\ \text{s.t.} \quad & y_i - (\mathbf{w}\mathbf{x}_i + b) \le \|\mathbf{w}\|\epsilon + \xi_i, \\ & (\mathbf{w}\mathbf{x}_i + b) - y_i \le \|\mathbf{w}\|\epsilon + \xi_i^*, \qquad (14) \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N, \end{aligned}$$

The above optimization problem is very similar to the ℓ_1-norm SVR, except that it has a margin related to the complexity term. In the following, we will prove that the above optimization is actually equivalent to the ℓ_1-norm SVR in the sense that, if one of the models for a given value of the parameter ϵ produces a solution $\{\mathbf{w}, b\}$, then the other method can derive the same solution by adapting its corresponding parameter ϵ.

Lemma 1: The LSVR model with setting $\Sigma_i = \mathbf{I}$ is equivalent to the ℓ_1-norm SVR in the sense that: (1) Assuming a unique ϵ_1^* exists for making ℓ_1-norm SVR optimal,[2] if for ϵ_1^* the ℓ_1-norm SVR achieves a solution $\{\mathbf{w}_1^*, b_1^*\} = \mathrm{SVR}(\epsilon_1^*)$, then the LSVR can produce the same solution by setting the parameter $\epsilon = \frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$, i.e., $\mathrm{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) = \mathrm{SVR}(\epsilon_1^*)$. (2) Assuming a unique ϵ_2^* exists for making the special case of LSVR optimal,[3] if for ϵ_2^* the special case of LSVR achieves a solution $\{\mathbf{w}_2^*, b_2^*\} = \mathrm{LSVR}(\epsilon_2^*)$, then the ℓ_1-norm SVR can produce the same solution by setting the parameter $\epsilon = \epsilon_2^*\|\mathbf{w}_2^*\|$, i.e., $\mathrm{SVR}(\epsilon_2^*\|\mathbf{w}_2^*\|) = \mathrm{LSVR}(\epsilon_2^*)$.

The proof can be seen in the appendix. In addition, if in LSVR we use the item of $\mathbf{w}^T\Sigma\mathbf{w}$ instead of its square root form as the structure risk or complexity risk, a similar proof can also be developed showing that the ℓ_2-norm SVR is equivalent to the special case of LSVR with $\Sigma_i = \Sigma$. In summary, we can see that the LSVR model actually contains the standard SVR model as its special case.

[2]This means that setting ϵ to ϵ_1^* will minimize the objective function of SVR.

[3]This means that setting ϵ to ϵ_2^* will minimize the objective function of LSVR

V. Kernelization

Only linear approximating functions are discussed in the above. We next kernelize the LSVR in order to generate nonlinear approximating functions. Assume a kernel mapping from the original space to a feature space is formulated as: $\mathbf{x}_i \to \varphi(\mathbf{x}_i)$, where $i = 1, \ldots, N$, and $\varphi : \mathbb{R}^n \to \mathbb{R}^f$ is a mapping function. The optimization of the relaxed LSVR in the feature space can be written as:

$$\begin{aligned} \min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \qquad (15) \\ \text{s.t.} \quad & y_i - (\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \le \epsilon t_i + \xi_i, \qquad (16) \\ & (\mathbf{w}^T\varphi(\mathbf{x}_i) + b) - y_i \le \epsilon t_i + \xi_i^*, \qquad (17) \\ & \sqrt{\mathbf{w}^T\Sigma_i^{\varphi}\mathbf{w}} \le t_i, \qquad (18) \\ & t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1, \ldots, N. \end{aligned}$$

To apply the kernelization, we need to represent the optimization and the final approximating function in a kernel form, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$. In the follownig, we present Theorem 1 showing that the representer theory is validate in LSVR.

Theorem 1: If the corresponding local covariance Σ_i^{φ} can be estimated by the mapped training data, i.e., $\hat{\varphi}_i$, Σ_i^{φ} can be written as

$$\Sigma_i^{\varphi} = \frac{1}{2k+1}\sum_{j=-k}^{k}(\varphi(\mathbf{x}_{i+j}) - \hat{\varphi}_i)(\varphi(\mathbf{x}_{i+j}) - \hat{\varphi}_i)^T,$$

$$\hat{\varphi}_i = \frac{1}{2k+1}\sum_{j=-k}^{k}\varphi(\mathbf{x}_{i+j}),$$

where we just consider $2k$ data points which are the closest to the i-th data point, then the optimal $\mathbf{w}$ lies in the span of the mapped training data.

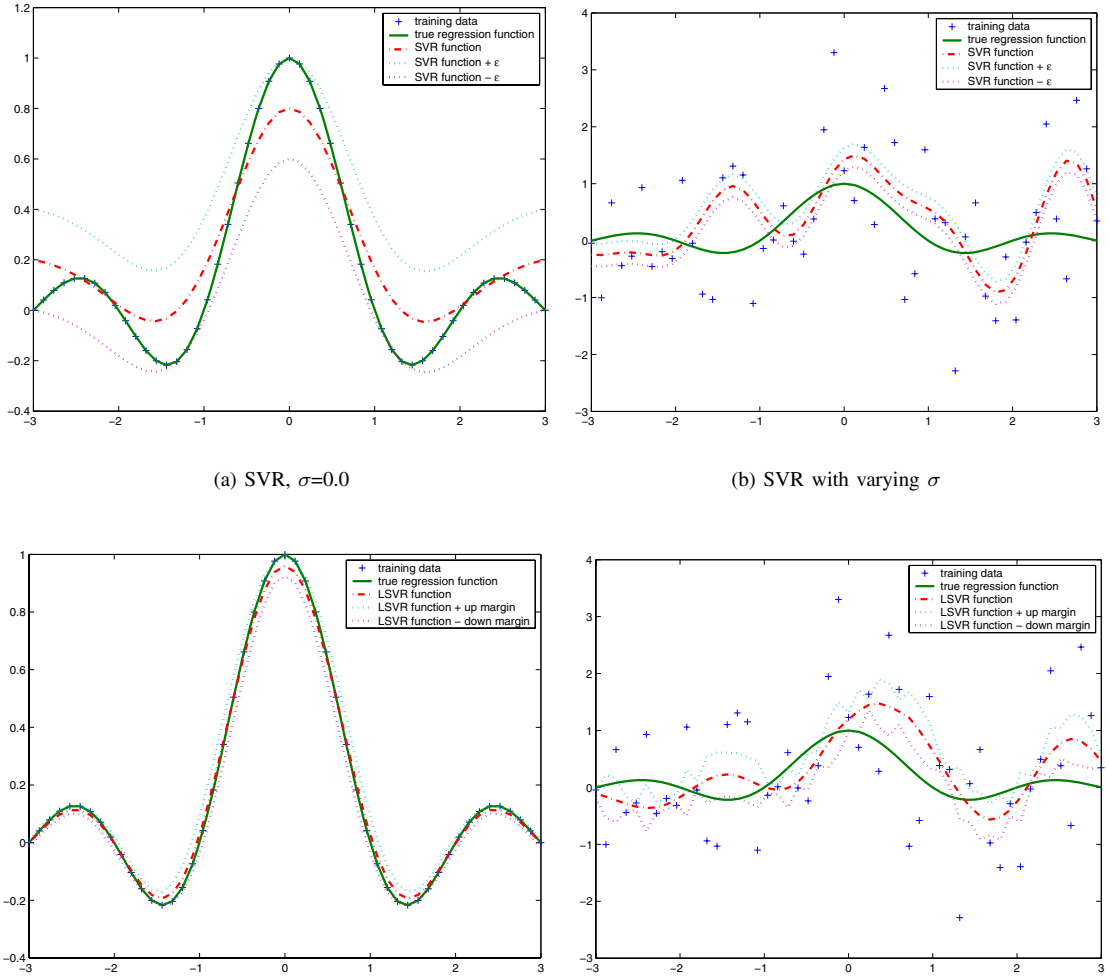
The proof is very similar to the proof for representer theory of the MEMPM and M^4 [2][3]. Due to the space limit, we omit the detailed procedure here.

VI. Experiments

In this section, we report the experiments on both synthetic *sinc* data and real world financial series data. The SOCP problem associated with our LSVR model is solved using a general software package, Sedumi [9]. The SVR algorithm is performed by LIBSVM [1].

A. Evaluations on Synthetic Sinc Data

50 examples (x_i, y_i) are generated from a *sinc* function [8], where x_i are drawn uniformly from $[-3.0,\ 3.0]$, and $y_i = \sin(\pi x_i)/(\pi x_i) + \tau_i$, with τ_i drawn from a Gaussian with zero mean and variance σ^2. Two cases are evaluated. In the one case, the standard deviation is set to zero, i.e., $\sigma = 0.0$; in the other case, the standard deviation of the data increases linearly from 0.5 at $x = -3.0$ to 1.5 at $x = 3.0$. Hence, in this case, the variance of noise is different in different regions. We use the default parameters $C = 100.0$ and the RBF kernel $\mathbf{K}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u}-\mathbf{v}\|^2)$. Table I reports the average results over 100 random trails with



(a) SVR, σ=0.0

(b) SVR with varying σ

(c) LSVR, σ=0.0

(d) LSVR with varying σ

Fig. 2. Experimental results on synthetic $sinc$ data with ϵ=0.2.

ϵ	Case I: $\sigma = 0.0$		Case II: Varying σ	
	LSVR	SVR	LSVR	SVR
0.0	0	0	0.1825±0.1011	0.3101±0.1165
0.2	0.0004	0.0160	0.2338±0.0888	0.2761±0.1111
0.4	0.0016	0.0722	0.1917±0.0726	0.2217±0.0840
0.6	0.0044	0.1695	0.1540±0.0687	0.2384±0.0867
0.8	0.0082	0.1748	0.1333±0.0674	0.2333±0.1096
1.0	0.0125	0.1748	0.1115±0.0597	0.2552±0.1218
2.0	0.0452	0.1748	0.0959±0.0421	0.2616±0.1517

TABLE I

EXPERIMENTAL RESULTS (MSE±STD) OF THE LSVR MODEL AND THE SVR ALGORITHM ON THE $sinc$ DATA WITH DIFFERENT ϵ VALUES

different ϵ values. Figure 2 illustrates the difference between the LSVR model and the SVR algorithm when $\epsilon = 0.2$. For the first case, $\sigma = 0.0$, the LSVR model can adjust the margin automatically to fit the data with a smaller Mean Square Error (MSE), as shown in Figure 2(c). However, since it uses a fixed margin, the SVR algorithm models the data poorly(see Figure 2(a)); as a result, the MSE increases as ϵ increases. We also note that, when $\epsilon \geq 0.8$, there are no support vectors in SVR and the MSE reaches a maximum. In the second case (with varying σ), the LSVR model has smaller MSEs and smaller STDs for all ϵ's. Figure 2(b) and 2(d) also show that the resulting approximating function in LSVR is smoother than that in SVR.

B. *Evaluations on Real Financial Data*

We evaluate our model on financial time series data; these are highly volatile in nature. The experimental data used are drawn from three major indices for the period January 2, 2004 to April 30, 2004: (1) the Dow Jones Industrial Average (DJIA), (2) the NASDAQ, and (3) the Standard & Poor 500 index (S&P500).

Following the procedure in [7], we convert the daily closing prices (d_t) of these indices to continuously compounded returns ($r_t = \log \frac{d_{t+1}}{d_t}$) and set the ratio of the number of

the training return series to the number of test return series to $5:1$. We perform normalization on these return series by $R_t = \frac{r_t - Mean(r_t)}{SD(r_t)}$, where the means and standard deviations are computed for each individual index in the training period. We compare the performance of the LSVR model against the SVR. The predicted system is modelled as $\hat{R}_t = f(\mathbf{x}_t)$, where $\mathbf{x}_t$ takes the previous four days' normalized returns as indicators, i.e., $\mathbf{x}_t = (R_{t-4}, R_{t-3}, R_{t-2}, R_{t-1})$. We choose to use the preceding four data points based on the suggestions in [7]. We then apply the modelled function f to test the performance by one-step ahead prediction. The trade-off parameter C and the parameter β of the RBF kernel ($\mathbf{K}(\mathbf{u}, \mathbf{v}) = \exp(-\beta\|\mathbf{u}-\mathbf{v}\|^2)$), are obtained by five-fold cross validation, conducting the SVR on the following paired points: $[2^{-5}, 2^{-4}, \ldots, 2^{10}] \times [2^{-5}, 2^{-4}, \ldots, 2^{10}]$. We obtain the corresponding parameters $\{2^4, 2^{-3}\}$, $\{2^{-3}, 2^1\}$, and $\{2^0, 2^2\}$ respectively for DJIA, NASDAQ and S&P500.

Pompe [7] has suggested that there is a relationship in the sequential five days' values. We therefore select $k = 2$, i.e., five days' values, to model the local volatility. Since when $\epsilon \geq 2.0$, there are no support vectors in the SVR, we just restrict the ϵ values in the range of $0.0, 0.2, \ldots, 1.0$ to 2.0. The corresponding MSE's are reported in Table II. As observed, the LSVR model demonstrates a consistent superiority to the SVR algorithm, even though the paired parameters (C, β) are not tuned for our LSVR model. Furthermore, a paired t-test [6], performed on the best results of both models in Table II, shows that the LSVR model outperforms SVR with $\alpha = 10\%$ significance level for a one-tailed test.

TABLE II

EXPERIMENTAL RESULTS OF THE LSVR MODEL AND THE SVR ALGORITHM ON THE FINANCIAL DATA WITH DIFFERENT ϵ VALUES

ϵ	DJIA		NASDAQ		S&P500	
	LSVR	SVR	LSVR	SVR	LSVR	SVR
0.0	0.9204	1.3241	1.2897	1.3050	**1.2372**	1.2833
0.2	0.9835	1.1274	**1.2896**	1.3246	1.2399	1.2831
0.4	0.9341	0.9156	1.2898	1.3314	1.2442	1.2952
0.6	**0.9096**	0.9387	1.2901	1.3404	1.2540	1.2887
0.8	0.9273	0.9450	1.2904	1.3891	1.2788	1.2798
1.0	0.9434	0.9713	1.2908	1.4105	1.3044	1.2664
2.0	0.9666	1.0337	1.2928	1.3619	1.2643	1.3220

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed the Local Support Vector Regression model in order to improve the performance of the standard Support Vector Regression model for time series prediction. In contrast to the standard Support Vector Regression model, our novel model offers a systematic and automatic scheme to adapt the margin locally and flexibly. Therefore, it can tolerate noise adaptively. We have demonstrated that this promising model not only captures the local information of data in approximating functions, but also incorporates the standard SVR as a special case. Moreover, kernelization can also be applied to this novel model. Therefore it can generate non-linear approximating functions and can be applied to general regression tasks. The experiments conducted on synthetic $sinc$ data and three series from real financial time series indices show that our model outperforms the standard SVR in modelling the data.

APPENDIX

Proof of Lemma 1. *Proof:* Since (1) and (2) are very similar statements, we only prove (1). When ϵ is set to $\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$ in the special case of LSVR, the value of the objective function of LSVR will always be smaller than the one obtained by setting $\{\mathbf{w}, b\} = \{\mathbf{w}_1^*, b_1^*\}$, since $\{\mathbf{w}_1^*, b_1^*\}$ is easily verified to satisfy the constraints of LSVR and SVR contains the objective function same as LSVR with $\Sigma_i = \mathbf{I}$. Namely,

$$f_{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) \leq f_{SVR}(\epsilon_1^*), \tag{19}$$

where we use $f_{SVR}(\epsilon_s)$ ($f_{LSVR}(\epsilon_s)$) to denote the value of the SVR (LSVR) objective function when ϵ is set to a specific value ϵ_s.

We assume the solution to be $\{\mathbf{w}_2, b_2\}$ when ϵ is set to $\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$ in the special case of LSVR. Similarly, by setting $\epsilon = \epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}$ in SVR, we have:

$$f_{SVR}(\epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}) \leq f_{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}). \tag{20}$$

Combining (19) and (20), we have:

$$f_{SVR}(\epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}) \leq f_{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) \leq f_{SVR}(\epsilon_1^*)\ . \tag{21}$$

Since ϵ_1^* is the unique ϵ that achieves the objective of minimizing SVR, (21) implies that $\|\mathbf{w}_2\| = \|\mathbf{w}_1^*\|$. This further implies that $\mathbf{w}_2$ is equal to $\mathbf{w}_1^*$, since, with $\|\mathbf{w}_2\| = \|\mathbf{w}_1^*\|$, the optimization of LSVR is exactly the same as that of SVR. This will naturally lead to the same solution. ■

REFERENCES

[1] C.-C. Chang and C.-J Lin. LIBSVM: a library for support vector machines, 2001. URL:http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[2] K. Huang, H. Yang, I. King, and M. R. Lyu. Learning large margin classifiers locally and globally. In Russ Greiner and Dale Schuurmans, editors, *The Twenty-first International Conference on Machine Learning (ICML-2004)*, pages 401–408, 2004.

[3] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. The minimum error minimax probability machine. *Journal of Machine Learning Research*, 5:1253–1286, 2004.

[4] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[5] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[6] D. C. Montgomery and G. C. Runger. *Applied statistics and probability for engineers*. Wiley, New York, 2nd edition, 1999.

[7] B. Pompe. Mutual information and relevant variables for predictions. In Abdol S. Soofi and Liangyue Cao, editors, *Modelling and forecasting financial data: techniques of nonlinear dynamics*, pages 61–92. Kluwer Academic Publishers, Boston, Mass., 2002.

[8] 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.

[9] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.