Vol 15 No 6, June 2006
1009-1963/2006/15(06)/1208-08

Chinese Physics

© 2006 Chin. Phys. Soc.
and IOP Publishing Ltd

Prediction of chaotic systems with multidimensional recurrent least squares support vector machines*

Sun Jian-Cheng(孙建成)[a)b)†], Zhou Ya-Tong(周亚同)[a)], and Luo Jian-Guo(罗建国)[a)]

[a)] *Department of Communication Engineering, University of Finance and Economics, Nanchang 330013, China*

[b)] *Department of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

(Received 25 December 2005; revised manuscript received 7 March 2006)

In this paper, we propose a multidimensional version of recurrent least squares support vector machines (MDRLS-SVM) to solve the problem about the prediction of chaotic system. To acquire better prediction performance, the high-dimensional space, which provides more information on the system than the scalar time series, is first reconstructed utilizing Takens's embedding theorem. Then the MDRLS-SVM instead of traditional RLS-SVM is used in the high-dimensional space, and the prediction performance can be improved from the point of view of reconstructed embedding phase space. In addition, the MDRLS-SVM algorithm is analysed in the context of noise, and we also find that the MDRLS-SVM has lower sensitivity to noise than the RLS-SVM.

Keywords: chaotic systems, support vector machines, least squares, noise
PACC: 0545

1. Introduction

Prediction, smoothing and filtering are traditional tasks applied to time series. Chaotic system is an important class of the nonlinear dynamical systems in economics, meteorology, chemical processes, biology and many other fields. Various techniques for predicting and modelling nonlinear time series have been investigated in past years. There are many traditional methods associated with time series analysis such as linear regression and auto regressive integrated moving average (ARIMA) models.[1] In addition, some new methods have been proposed to deal with the nonlinear time series analysis, such as chaotic attractor[2] and radial basis function.[3] However, the machine learning community has recently paid much attention to these problems and especially to nonlinear time series prediction in various areas such as biological signals, speech or financial markets. For instance, recurrent artificial neural networks[4] and support vector regressors[5–8] have been developed for prediction and modelling purposes for chaotic time series. The training set of algorithms in Refs.[5–8] is indeed involved with the theory of reconstructed embedding phase space in some sense. However, the reasonable explanation was not given to the training set, and the information of high-dimensional space was not utilized sufficiently, either.

In this paper, we focus on prediction tasks of chaotic time series by proposing multidimensional recurrent least squares support vector machines (MD RLS-SVM). Support vector machines (SVM) have become a subject of intensive study and they have been applied successfully to classification and regression.[9] The least squares (LS) version of SVM greatly simplifies the problem since the solution is characterized by a linear system.[10] In our work, a multidimensional regression version of RLS-SVM is developed for carrying out the prediction problem. We expect that the prediction performance can be improved since the multidimensional regression is a better solution than RLS-SVM based on the reconstructed embedding phase space theory.

In rest of this paper, we review the theory about reconstructed embedding phase space in Section 2. In Section 3, the MDRLS-SVM algorithm is derived in order to utilize the information of high-dimensional space. The sensitivity of MDRLS-SVM to noise environment is analysed in Section 4. Simulation results are offered in Section 5 to support our theoretic findings. Finally, concluding remarks are given in Section

*Project supported by the National Natural Science Foundation of China (Grant No 90207012).
†E-mail: sunjc.73@163.com

6.

2. Embedding phase space of dynamical system

The training data set is essentially utilizing SVM to fulfil the prediction task, and it is a phase space in chaotic time series. The phase space needs to be reconstructed since we can only acquire a scalar time series in various cases. Deterministic dynamical systems describe the time evolution of a system in some phase space $\Gamma \subset \boldsymbol{R}^m$. For simplicity, we assume that the phase space is a finite dimensional vector space. A state is specified by a vector $\boldsymbol{x} \in \boldsymbol{R}^m$. Then we can describe the dynamics by an explicit system of m first-order ordinary differential equations

$$\frac{\mathrm{d}}{\mathrm{d}\,t}\boldsymbol{x}(t) = f(t, \boldsymbol{x}(t)), \qquad t \in \boldsymbol{R}. \tag{1}$$

A time series can then be thought of as a sequence of observations $\boldsymbol{s} = h\{\boldsymbol{x}\}$ performed with some measurement function h. Since the sequence $\boldsymbol{s}$ (usually scalar) in itself does not properly represent the multidimensional phase space of the dynamical system, we have to employ some technique to unfold the multidimensional structure using the available data. Takens's embedding theorem guarantees the reconstruction of a state space representation from a scalar signal alone.[11] Takens states that if the sequence $\boldsymbol{s}$ does consist of scalar measurements of the state of a dynamical system, then on certain generic assumptions, the time delay embedding provides a one-to-one image of the original set $\{\boldsymbol{x}\}$, provided that the embedding dimension is large enough. Here $\boldsymbol{s}$ are vectors in a new space namely the embedding phase space that is formed from time delayed values of the scalar measurements.

In our work, a discrete version of Eq.(1) is discussed with $t = n\Delta t$ by maps in the form of

$$\boldsymbol{s}_{n+1} = \boldsymbol{F}(\boldsymbol{s}_n). \tag{2}$$

The problem of prediction becomes equivalent to the problem of estimating the unknown function $\boldsymbol{F}(\cdot)$ in the embedding phase space.

3. Multidimensional recurrent least squares support vector machines and prediction algorithm

The foundations of SVM have been developed by Vapnik and possessed promising empirical performance.[9] And the prediction problem has been discussed in Ref. [6]. In fact, the prediction task utilizing the SVM was involved in the theory of reconstructed embedding phase space impliedly, namely for almost all the time delay and for some embedding dimension, Takens's embedding theorem ensures that there is a smooth map $f: \boldsymbol{R}^m \to \boldsymbol{R}$ such that

$$s_{(n+1)\tau_\mathrm{d}} = f(s_{n-(m-1)\tau_\mathrm{d}}, ..., s_{n-\tau_\mathrm{d}}, s_n) = f(\boldsymbol{s}_n), \tag{3}$$

where $\boldsymbol{s}_n$ are vectors in the embedding phase space. m and τ_d are referred to as embedding dimension and delay, respectively. So the prediction task can be achieved by estimating the function $f(\cdot)$, and Eq.(3) is also a regular form in SVM for prediction. It is a pity that this relationship was not presented straightforwardly in the above-mentioned references. However, the difference between Eq.(2) and Eq.(3) is obvious. The left-hand side of Eq.(2) is a vector while a scalar quantity appears in Eq.(3). What the difference means is the most important problem to be solved. The practical meaning of Eq.(2) is clear since it shows the relationship between two points in the embedding phase space. Nevertheless, we cannot confirm exactly the practical meaning of the Eq.(3), and it has not utilized the information of high-dimensional embedding space adequately. Suykens and Vandewalle proposed the recurrent least squares support vector machines (RLS-SVM) based on sum squared error (SSE) to deal with some problem in prediction such as iterative operation.[10] However, the same problem exists in their prediction methods. First of all, we review the classical RLS-SVM to explain our method.

Let $\{s(t)\}$, $t = 1, 2, ..., T$, be a scalar time series that is generated by a dynamical system and choose an optimal embedding dimension m, time delay τ_d and set $Q \subseteq \{1, \ldots, N\}$, then the training data will be constructed as follows:

$$\begin{aligned} D = &\{((s_{k-(m-1)\tau_\mathrm{d}}, ..., s_{k-\tau_\mathrm{d}}, s_k), s_{k+1}) \\ &\in \boldsymbol{R}^m \times \boldsymbol{R} | k \in Q\}. \end{aligned} \tag{4}$$

Given an initial condition $\hat{s}_i = s_i$ for $i = 1, 2, ..., m$, the prediction problem is given by

$$\begin{aligned} \hat{s}_{(k+1)\tau_\mathrm{d}} =& f(\hat{s}_{k-(m-1)\tau_\mathrm{d}}, ..., \hat{s}_{k-\tau_\mathrm{d}}, s_k) \\ =& w^\mathrm{T}\varphi(\boldsymbol{s}_k) + b, \end{aligned} \tag{5}$$

where $\varphi(\cdot) : \boldsymbol{R}^m \to \boldsymbol{R}^{n_\mathrm{h}}$ is a nonlinear mapping in a future space, $w \in \boldsymbol{R}^{n_\mathrm{h}}$ is the output

weight vector and $b \in \boldsymbol{R}$ is the bias term. $\boldsymbol{s}_k = [\hat{s}_{k-(m-1)\tau_d}; ...; \hat{s}_{k-\tau_d}; \hat{s}_k]$. So the function $f(\cdot)$ is estimated by using training data D, and the prediction can be realized based on the function $f(\cdot)$.

In Ref.[10], Suykens and Vandewalle proposed that Eq.(5) could be converted into optimal problems that can be described as follows:

$$\min_{w,e} J(w,e) = \frac{1}{2} w^{\mathrm{T}} w + \gamma \frac{1}{2} \sum_{k=m+1}^{N+m} e_k^2 \qquad (6)$$

subject to

$$\begin{aligned} s_{k+1} - e_{k+1} =& w^{\mathrm{T}} \varphi(\boldsymbol{s}_k - \boldsymbol{e}_k) + bk \\ =& m+1, ..., N+m, \end{aligned} \qquad (7)$$

where $e_k = s_k - \hat{s}_k$, $\boldsymbol{s}_k = [s_{k-(m-1)\tau_d}; ...; s_{k-\tau_d}, s_k]$, $\boldsymbol{e}_k = [e_{k-(m-1)\tau_d}; ...; e_{k-\tau_d}; e_k]$, and γ is an adjustable constant. The basic idea of mapping function $\varphi(\cdot)$ is to map the data into a high-dimension featured space, and to do linear regression in this space. The mapping function $\varphi(\cdot)$ can be paraphrased by a kernel function $K(\cdot,\cdot)$ because of the application of Mercer's theorem, which means that

$$K(x_i, x_j) = \varphi(x_i)^{\mathrm{T}} \varphi(x_j) \qquad (8)$$

with radial basis functions (RBF) kernels, one employs

$$K(x_i, x_j) = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\Big). \qquad (9)$$

We can deduce that the practical meaning of RLS-SVM for prediction is unclear based on the above discussion. The information of the high-dimensional embedding phase space is not utilized completely though the training data come from the reconstructed embedding phase space. So we propose the multidimensional regression estimation problem when the output variable is a vector $\boldsymbol{s} \in \boldsymbol{R}^k$ instead of a scalar quantity in Eq.(5). In order to illustrate this problem clearly, we compare the training data pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ of RLS-SVM in Eq.(9) with those of our multidimensional version (MDV) in Eq.(10).

$$\begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x}_{N-m} \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{m-2} \\ s_1 & s_2 & \cdots & s_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-m} & s_{N-m+1} & \cdots & s_{N-2} \end{bmatrix}, \quad \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-m-2} \end{bmatrix} = \begin{bmatrix} s_{m-1} \\ s_m \\ \vdots \\ s_{N-1} \end{bmatrix}, \qquad (10)$$

$$\begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x}_{N-m-1} \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{m-1} \\ s_1 & s_2 & \cdots & s_m \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-m-1} & s_{N-m} & \cdots & s_{N-2} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y_1} \\ \vdots \\ \boldsymbol{y}_{N-m-1} \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & \cdots & s_m \\ s_2 & s_3 & \cdots & s_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-m} & s_{N-m+1} & \cdots & s_{N-1} \end{bmatrix} \qquad (11)$$

where m is the embedding dimension and N is the length of scalar time series. We can see clearly that the training data pairs of MDV express the relationship between two points in high-dimensional space, however, the meaning of Eq.(10) is not clear from the point of view of reconstructed embedding phase theory.

Fortunately, the multidimensional regression problem can be divided into k 1D problems for the RLS-SVM since minimum variance estimate is equivalent to the multidimensional estimate, but, not for SVM due to the insensitive zone ranging around the estimate will not equally treat each training sample. So we investigate the multidimensional versions of RLS-SVM, and the algorithm is formulated as follows.

The prediction problems are reconsidered in the embedding phase space which is reconstructed based on the observed scalar time series $\boldsymbol{s} = \{s_0, s_2, ..., s_{N-1}\}$. The prediction method is presented as follows:

$$\boldsymbol{s}_k = \boldsymbol{f}(\boldsymbol{s}_{k-1}) \qquad (12)$$

$k = 0, 1, ..., N-m$, $\boldsymbol{s}_k = [s_{k-(m-1)\tau_d}, s_{k-(m-2)\tau_d}, ..., s_k]$, where the m and τ_d are referred to as embedding dimension and time delay respectively. The τ_d is set to be one for simplicity. So the prediction function is given by

$$\hat{\boldsymbol{s}}_k = \boldsymbol{W}^{\mathrm{T}} \varphi(\boldsymbol{s}_{k-1}) + \boldsymbol{b}, \qquad (13)$$

where $\boldsymbol{W} = [\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^m]$, $\boldsymbol{b} = [b^1, b^2, ..., b^m]$.

$\varphi(\cdot) : \boldsymbol{R}^m \to \boldsymbol{R}^{n_h}$ is the nonlinear mapping in the future space. So the estimation of above function can then be similarly stated as the quadratic optimization problem.

$$\min_{w,e} J(\boldsymbol{w}, \boldsymbol{e}) = \frac{1}{2}\sum_{j=1}^{m} \|\boldsymbol{w}^j\|^2 + \gamma \frac{1}{2}\sum_{k=1}^{N-m} \|\boldsymbol{e}_k\|^2 \quad (14)$$

subject to

$$\begin{aligned} \boldsymbol{s}_k - \boldsymbol{e}_k =& \boldsymbol{W}^{\mathrm{T}}\varphi(\boldsymbol{s}_{k-1} - \boldsymbol{e}_{k-1}) + \boldsymbol{b}\,, \\ & k = 1, 2, ..., N-m\,, \end{aligned} \quad (15)$$

where $\boldsymbol{e}_k = \boldsymbol{s}_k - \hat{\boldsymbol{s}}_k$. Making use of the Lagrange multipliers leads to the minimization of

$$\begin{aligned} L(\boldsymbol{w}, \boldsymbol{e}, \boldsymbol{b}, \alpha) =& \frac{1}{2}\sum_{j=1}^{m} \|\boldsymbol{w}^j\|^2 + \gamma \frac{1}{2}\sum_{k=1}^{N-m} \|\boldsymbol{e}_k\|^2 \\ & + \sum_{k=1}^{N-m} \alpha_{k-m}[\boldsymbol{s}_k - \boldsymbol{e}_k \\ & - \boldsymbol{W}^{\mathrm{T}}\varphi(\boldsymbol{s}_{k-1} - \boldsymbol{e}_{k-1}) - \boldsymbol{b}] \end{aligned} \quad (16)$$

with respect to $\boldsymbol{w}^j$, b^j and $\boldsymbol{e}^j$ and its maximization with respect to the Lagrange multipliers, α. The solution to this problem is given by the Karush–Kuhn–Tucker theorem. So the multi-dimensional versions of RLS-SVM can then be presented as

$$\hat{\boldsymbol{s}}_k = \sum_{p=1}^{m}\sum_{l=1}^{N-m} \alpha_l K(\boldsymbol{z}_l, \boldsymbol{s}_{k-1}) + b^p\,, \quad (17)$$

where $\boldsymbol{z}_l = \boldsymbol{s}_l - \boldsymbol{e}_l$, and $K(\cdot)$ is the RBF kernel function that is given in Eq.(9).

From the above discussion it can be seen that our algorithm carried out the prediction task in the high dimensional space that possesses more physical and geometrical information than the scalar time series. So the better prediction performance should be acquired theoretically.

4. Analysis of sensitivity to noise for MDRLS-SVM

In fact, the prediction algorithm mentioned above can also be regard as the function approximation algorithm. In this paper, MDRLS-SVM is proposed to carry out the function approximation in a reconstructed phase space. Since the sequence (usually scalar) in itself does not properly represent the multidimensional phase space of the original dynamical system, the multidimensional structure using the available data should be reconstructed using Takens's embedding theorem.[11]

Once the scalar time series has been embedded we can model the dynamics by fitting a function to map points forward in time. There exist a variety of methods for approximating the mapping function either locally or globally, however the basic idea is the same in each case. Having defined a space of functions, we wish to choose the function that approximates the dynamics best. Generally this means minimizing the errors in a least squared sense, where the error terms are defined as

$$\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{f}(\boldsymbol{x}_{i-1})\,, \quad (18)$$

where $\boldsymbol{e}_i$ is the approximation error associated with $\boldsymbol{x}_i$, the ith point in the series, and $f(\cdot)$ is the approximating function.

For analysing the relationship between function approximation and noise reduction, we review the approach that is so called shadowing method.[12] Suppose that we observe a noisy time series $\{\boldsymbol{y}_t\}$, where $t = 1,\ 2, ..., N$, and it can be decomposed as

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{n}_t\,, \quad (19)$$

where $\boldsymbol{x}_t$ is the 'desired' part of the signal and $\boldsymbol{n}_t$ is the 'noise'. In order to get rid of the noise, the distance between the trajectory segments $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ is introduced as

$$D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \|\boldsymbol{y}_t - \boldsymbol{x}_t\|^2}\,, \quad (20)$$

where $\|\boldsymbol{y}_t - \boldsymbol{x}_t\|$ is the Euclidean distance between the vectors $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, thus the Euclidean norm is used here to measure the distances both between the two points $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and between two trajectories $\boldsymbol{x}$ and $\boldsymbol{y}$. In Ref.[12], the noise reduction was implemented by minimizing the $D^2(\cdot)$ subject to the constraint that $\boldsymbol{x}$ is deterministic with respect to $\boldsymbol{f}$, i.e.

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t)\,. \quad (21)$$

We can reach the conclusion, by comparing Eq.(18) with Eq.(20), that the noise reduction procedure can be viewed as a function approximation procedure in a general way, since both procedures essentially aim to minimize the approximation errors, only with respect to different parameters.

5. Simulation and discussion

In the following procedure, we used the Lorenz system[13] to test the prediction performance of our method. The equations are described as follows:

$$\begin{aligned} \frac{\mathrm{d}\,x}{\mathrm{d}\,t} &= \sigma(y-x)\,, \\ \frac{\mathrm{d}\,y}{\mathrm{d}\,t} &= rx-y-xz\,, \\ \frac{\mathrm{d}\,z}{\mathrm{d}\,t} &= xy-bz\,. \end{aligned} \tag{22}$$

A Lorenz attractor can be generated by taking $\sigma = 10$, $r = 28$ and $b = -8/3$. A trajectory has been generated for initial condition [0.1; 0.1; 0.1] by using a chaos function of TSTOOL toolbox (this toolbox can be free to download from www.physik3.gwdg.de/tstool/indexde.html).

The reconstructed embedding phase space has been discussed in Section 2, Cao proposed a method to determine the minimal sufficient embedding dimension that used an improved false nearest neighbour method.[14] The time delayed mutual information was suggested by Fraser and Swinney[15] as an effective tool to determine a reasonable τ_{d}. After the transients had died down a time series $s(k)$ of 700 points was obtained from the x coordinate. The time series was divided into two subsets referred to as the training subset and the test subset. The training subset consists of 500 time entries, and the test subset consists of 200 entries.

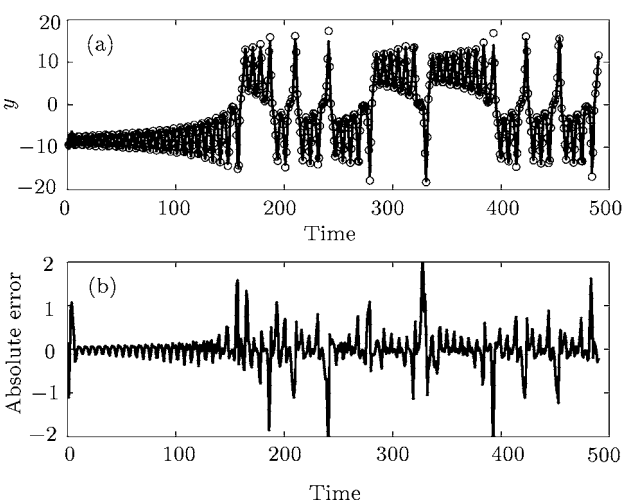


Fig.1. Function training using RLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel. The circle and continuous lines are for the true and training points respectively in the above plot.

The algorithms of MDRLS-SVM and RLS-SVM each can be divided into two parts that are training and prediction. The training procedure is to learn and understand the characters of the system, Thereby the prediction can be implemented based on the results of training procedure. So we first evaluate the performance of MDRLS-SVM in training procedure. In order to test the performance, the absolute errors between evaluated and true time series are defined as

$$E_{\mathrm{abserr}}(k) = \hat{s}(k) - s(k)\,, \tag{23}$$

where $\hat{s}(k)$ and $s(k)$ are the evaluated point and true point, respectively. The training performances of MDRLS-SVM and RLS-SVM without noise are illustrated in Fig.1 and Fig.2, respectively.

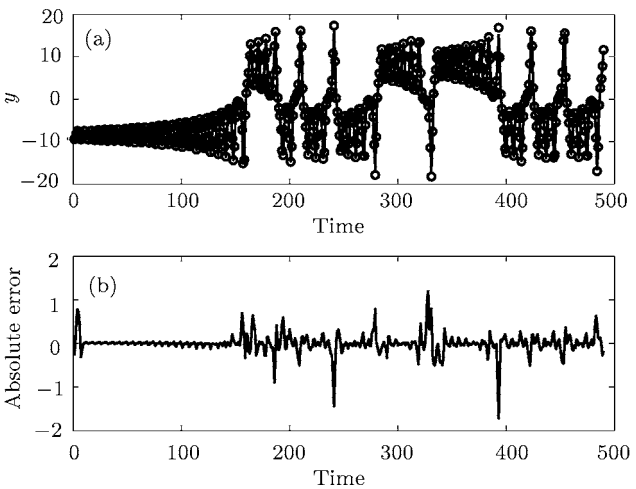


Fig.2. Function training using MDPLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel. The circle and continuous lines are for the true and training points respectively in the above plot.

From the simulation results, we can see that both MDRLS-SVM and RLS-SVM can learn the system perfectly. Our method can acquire better performance than RLS-SVM in the absence of noise. However, the improvement is not obvious. In fact, numerous leaning methods, such as neural networks, can also achieve satisfactory performance in training procedure sometimes. Nevertheless, some methods are not robust in some cases, for instance, time series with noise and outliers. So we show the training performances of MDRLS-SVM and RLS-SVM in the context of noise in Fig.3, Fig.4 and Fig.5, respectively. Before further discussion, the sensitivity to noise is defined as follows to measure the training performance:

$$\xi(k) = E_{\mathrm{abserr}-N} - E_{\mathrm{abserr}}\,, \tag{24}$$

where, the $E_{\mathrm{abserr}-N}$ and E_{abserr} are the absolute errors defined in Eq.(23) in the noise and clean environments, respectively.

From Fig.3 and Fig.4, we find that the training performances of MDRLS-SVM and RLS-SVM decrease due to the noise being introduced. These results show that the sensitivity to noise exists in the two methods; however, their difference is obvious.

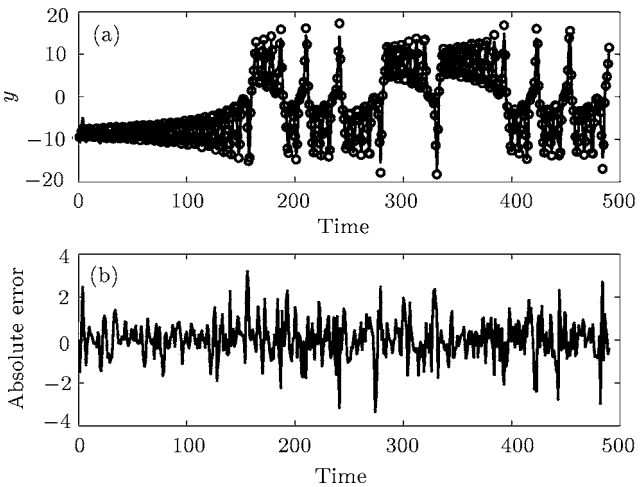


Fig.3. Function training using RLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in noise environment (SNR=20dB). The circle and continuous lines are for the true and training points respectively in the above plot.

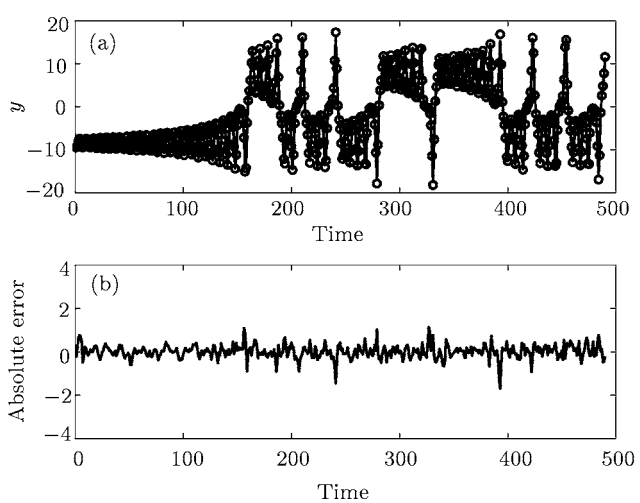


Fig.4. Function training using MDRLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in noise environment (SNR=20dB). The circle and continuous lines are for the true and training points respectively in the above plot.

To evaluate the sensitivity, the results are given in Fig.5 according to the definition in Eq.(24). In other words, the sensitivity to noise reflects the robust feature of the algorithm in the noise case. The result in Fig.5 shows an identity of views to the analysis in Section 4, that is, the MDRLS-SVM has better robust feature than the RLS-SVM in the context of noise.

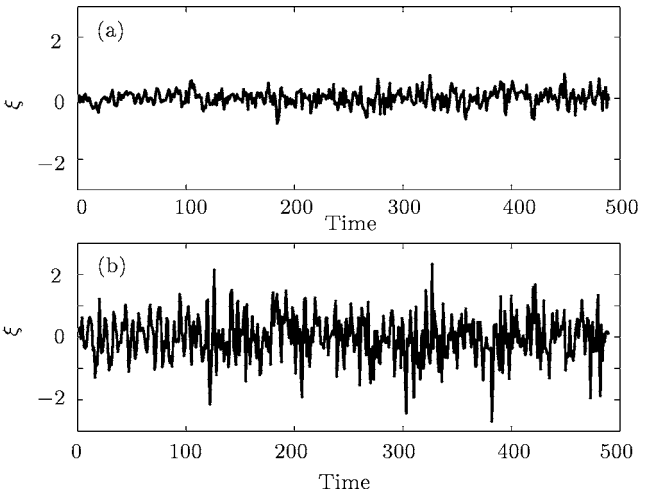


Fig.5. Sensitivity to noise. (a) MDRLS-SVM. (b) RLS-SVM.

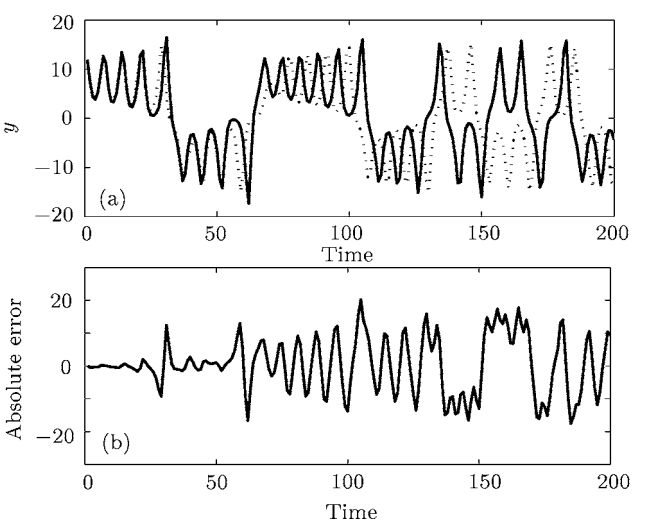


Fig.6. Prediction using RLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in clean environment. The dotted and continuous lines are for the prediction and true points respectively in the above plot.

In addition to the training performance mentioned above, another important issue that we focus on is the prediction ability (generalization ability is used in some literature) of the machine learning algorithm. So the prediction performances of the RLS-SVM and our MDRLS-SVM are shown in Figs.6–9, respectively. We compare the prediction performance of RLS-SVM with that of MDRLS-SVM in the clean and noise environments, respectively. The criterion for evaluating is similar in form to the definition in

Eq.(21) other than the $\hat{s}(k)$ that is referred to as the prediction point.

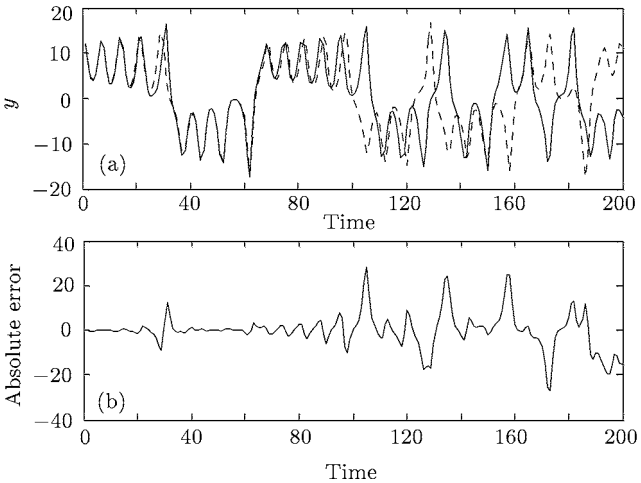


Fig.7. Prediction using MDRLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in clean environment. The dotted and continuous lines are for the prediction and true points respectively in the above plot.

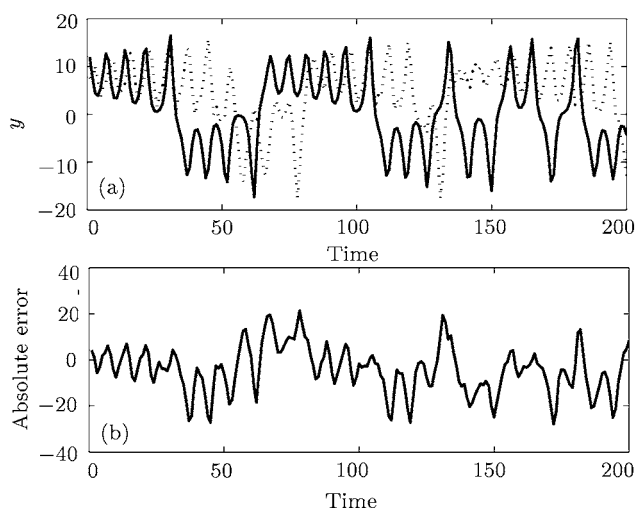


Fig.8. Prediction using RLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in noise environment (SNR=20dB). The dotted and continuous lines are for the prediction and true points respectively in the above plot.

The performances of prediction of the MDRLS-SVM and the RLS-SVM in the clean environment are illustrated in Fig.6 and Fig.7, respectively. We can see that about 50 steps can be predicted accurately for the RLS-SVM, and the prediction errors increase gradually in the rest of points. The errors mainly result from two sources. One is the essential characters of the chaotic system, that is, the chaotic time series that only can be predicted in a short range due to the 'butterfly effect'. The other, obviously, comes from the algorithm in which the system only can be learnt accurately in a certain principle, and the prediction errors will cumulate with prediction step increasing. However, Fig. 7 shows that about 100 steps can be predicted perfectly for our MDRLS-SVM algorithm. Comparing with the RLS-SVM, the prediction performance is improved since the MDRLS-SVM can utilize enough information of high-dimensional space. In the same way, the prediction error increases in the rest of the points due to the same source as that for the RLS-SVM, which is mentioned above.

In order to evaluate the prediction performance of our method in the noise case, simulation is carried out by adding white noise with Guess distribution to clean time series. Figures 8 and 9 show the simulation results of the RLS-SVM and the MDRLS-SVM, respectively. Based on the results, we can deduce that the prediction performances of two methods degrade due to introducing the noise. Specially, the performance of RLS-SVM cannot be accepted in this case. Nevertheless, 50 steps can still be predicted exactly for the MDRLS-SVM. These results are consistent with those of training procedure, namely, the MDRLS-SVM is more robust than the RLS-SVM.

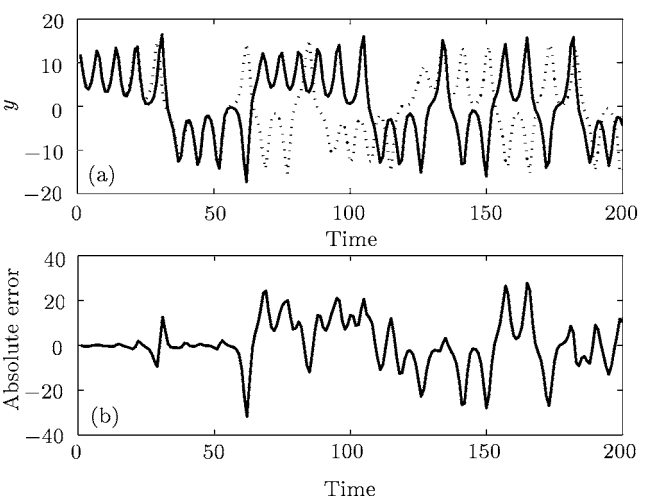


Fig.9. Prediction using RLS-SVM ($\gamma = 10$, $\sigma = 10$) with RBF kernel in noise environment (SNR=20dB). The dotted and continuous lines are for the prediction and true points respectively in the above plot.

6. Conclusions and future work

In order to improve the prediction performance, we introduce the multidimensional version of RLS-SVM in this paper. The better performance can be acquired based on the reconstructed embedding phase space theory since the high-dimensional space contains more information of system than the scalar time series. In addition, we analyse the sensitivity to noise

of MDRLS-SVM. Finally, the Lorenz system is used to test the performance. The simulation result shows that the better performance can be achieved for the MDRLS-SVM, especially in context of noise.

We think that the MDRLS-SVM may be effective in application when the 'true' multidimensional case is taken into account, such as multidimensional economic time series and multiple-input multiple-output system (MIMO) that is formed without using techniques of reconstructed embedding phase space. In addition, the embedding theory of chaotic system may guide us in determining the length of training data, which is a complicated problem in machine leaning method. These are our future research subjects about multidimensional regression.

References

[1] Brillinger D R 1981 *Time series, Data Analysis and Theory* (New York: McGraw-Hill)

[2] Li K P and Chen T L 2002 *Chin. Phys. Lett.* **19** 904

[3] Casdagli M 1989 *Physica* D **35** 335

[4] Dorffner G 1996 *Neural Network World* **6** 447

[5] Mukherjee S, Osuna E and Girosi F 1997 *In Proc. IEEE NNSP*'97 511

[6] Smola M K, Rätsch A J, Schölkopf G, Kohlmorgen J B and Vapnik V 1997 *Artificial Neural Networks—ICANN*'97 999

[7] Cui W Z, Zhu C C, Bao W X and Liu J H 2005 *Chin.Phys.* **14** 922

[8] Sun J C, Zhang T Y and Liu F 2004 *Chin. Phys.* **13** 2045

[9] Vapnik V 1995 *The Nature of Statistical Learning Theory* (New York: Springer)

[10] Suykens J A K and Vandewalle J 2000 *IEEE Tran. on Circuits and System—I: Fundamental Theory and Applications* **47** 1109

[11] Takens F 1981 *Dynamical systems and turbulence* (Berlin: Springer) p 366

[12] Farmer J D and Sidorowich J J 1991 *Physica* D **47** 373

[13] Kaplan D and Glass L 1995 *Understanding Nonlinear Dynamics* (New York: Springer)

[14] Cao L 1997 *Physica* D **110** 43

[15] Fraser A M and Swinney H L 1986 *Phys. Rev.* A **33** 1134