Vol 15 No 6, June 2006
1009-1963/2006/15(06)/1196-05

Chinese Physics

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

Chaotic time series prediction using fuzzy sigmoid kernel-based support vector machines*

Liu Han(刘 涵)[a], Liu Ding(刘 丁)[a], and Deng Ling-Feng(邓凌峰)[b]

[a] *School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China*

[b] *Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada*

(Received 17 August 2005; revised manuscript received 13 March 2006)

Support vector machines (SVM) have been widely used in chaotic time series predictions in recent years. In order to enhance the prediction efficiency of this method and implement it in hardware, the sigmoid kernel in SVM is drawn in a more natural way by using the fuzzy logic method proposed in this paper. This method provides easy hardware implementation and straightforward interpretability. Experiments on two typical chaotic time series predictions have been carried out and the obtained results show that the average CPU time can be reduced significantly at the cost of a small decrease in prediction accuracy, which is favourable for the hardware implementation for chaotic time series prediction.

Keywords: support vector machines, chaotic time series prediction, fuzzy sigmoid kernel
PACC: 0545

1. Introduction

With the rapid development of chaos theory and its application, chaos system control and chaos signal prediction[1−3] have become research hotspots now. Chaos signal, derived from a certain nonlinear system as a type of broad band noise, is characteristic of being div-dimensional, sensitive to initial conditions, and difficult to make a long-term prediction. Therefore, chaos theory is now increasingly playing an important role in the areas of communication, radar, control, signal processing, medicine, etc. The chaotic time series prediction is based on embedding theorem and phase space reconstruction theory, which was proposed by Takens[4] to regain chaos attractors in a higher-dimensional space.

A statistical learning theory (SLT) introduced by Vapnik is specific for machine learning with small samples, based on which a support vector machines (SVM) was proposed as a new and promising classification and regression technique. The SVM has better generalization ability due to its principle of structural risk minimization. Furthermore, by transforming the problem of solving optimal classification plane into the problem of solving convex quadratic programming, it is capable of resolving practical problems about, for examples, small data samples, non-linearity, high dimension, local minimization, etc. There have existed some successful examples in chaotic time series prediction;[5,6] however, they are devoted only to the prediction accuracy but not the implementation efficiency. When the SVM is used, input data need to be mapped into high-dimension featured space by applying a symmetric and positive semi-definite (PSD) kernel function, in order to transform linear non-separable problems into linear separable ones. Mathematically, those kernel functions must meet Mercer conditions;[7] linear, polynomial, Gauss and sigmoid kernel functions are often used. Note that the sigmoid function has some restrictions and works only when some specific constraints (slope term α and offset term β) are satisfied; however, it has salient classification performance when applied to neural networks.

The sigmoid kernel function is a nonlinear function, therefore it is difficult to perform the hardware implementation, and time consuming as well. One solution is to use subsection linearization, but it may lead to being non-differentiable at subsection points. Another method is to extract the first term (linear

*Project supported by the Doctoral Program Foundation of Institutions of Higher Eduction of China (Grant No 20040700010) and the Nature Science Specialties Foundation of Education Bureau of Shaanxi Province, China (Grant No 05JK267).

term) of Taylor expansion at the cost of losing important nonlinear characteristics. In this study, the sigmoid function is combined with fuzzy logic method, which is favourable for predigesting calculation and hardware implementation. Experiments on two typical chaotic time series predictions have been carried out and the obtained results show that the average CPU time can be decreased significantly at the cost of a slight reduction in prediction accuracy.

The rest of the paper is organized as follows. In Section 2, the principle of SVM based on the fuzzy sigmoid kernel is presented. Then, two chaotic time series prediction examples are given in Section 3 to show the advantages of the proposed method. Finally, the conclusions drawn from our study are given in Section 4.

2. Principle of fuzzy sigmoid kernel-based SVM

Given a training set $\{x_k, y_k; k = 1, 2, ..., N\}$, where $x_k \in R^n$ and $y_k \in R$ are the input and output data, respectively. By appropriately choosing a nonlinear function $\Phi(\bullet)$, the above training set can be mapped into high-dimension featured space F, in which the optimized linear regression function can be constructed as follows:

$$f(x) = w^{\mathrm{T}} \cdot \Phi(x) + b, \tag{1}$$

which implies that the linear regression in high-dimensional space corresponds to the nonlinear regression in low-dimensional input space.

Given the definition of Vapnik ε insensitive loss function,[8,9] we have

$$|y - f(x)|_\varepsilon = \begin{cases} 0, & |y - f(x)| \le \varepsilon, \\ |y - f(x)| - \varepsilon, & \text{else.} \end{cases} \tag{2}$$

Therefore, solving Eq.(1) can be transformed into resolving the following optimization problem:

$$\min_{\omega,b,\xi,\xi^*} J(w, \xi, \xi^*) = \frac{1}{2} w^{\mathrm{T}} w + C \sum_{k=1}^{N} (\xi_k + \xi_k^*), \tag{3}$$

$$\text{subject to} \quad \begin{cases} y_k - w^{\mathrm{T}} \Phi(x_k) - b \le \varepsilon + \xi_k, \\ w^{\mathrm{T}} \Phi(x_k) + b - y_k \le \varepsilon + \xi_k^*, \\ \xi_k, \xi_k^* \ge 0, \end{cases} \tag{4}$$

where ε denotes the approximation accuracy, ξ_k and ξ_k^* are slackness variables when the accuracy of f exceeds ε, and $C > 0$ is a punishment parameter. The above optimization problem can be transformed into the following Lagrange dual problem:

$$\max_{\alpha,\alpha^*} L(\alpha, \alpha^*) = -\frac{1}{2} \sum_{k,l=1}^{N} (\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*) K(x_k, x_l) - \varepsilon \sum_{k=1}^{N} (\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k (\alpha_k - \alpha_k^*), \tag{5}$$

$$\text{subject to} \quad \begin{cases} \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) = 0, \\ \alpha_k, \alpha_k^* \in [0, C], \end{cases} \tag{6}$$

where α_k and α_k^* are Lagrange multipliers and the kernel function $K(x_k, x_l)$ is defined as

$$K(x_k, x_l) = \Phi(x_k)^{\mathrm{T}} \cdot \Phi(x_l). \tag{7}$$

Therefore it is only necessary to choose an appropriate kernel function rather than manage to know the exact expression of Φ. The solution to above nonlinear function regression is

$$f(x) = \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) K(x, x_k) + b. \tag{8}$$

The linear, polynomial, Gauss and sigmoid kernel functions are defined as follows:

$$K(x_i, x_j) = x_i \cdot x_j, \tag{9}$$

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d, \tag{10}$$

$$K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right), \tag{11}$$

$$K(x_i, x_j) = \tanh(\alpha \cdot x_i \cdot x_j + \beta), \tag{12}$$

which are often used currently. It is well known that a kernel function meets Mercer condition if and only if it is a symmetric and PSD matrix. However, some non-PSD kernels, such as the sigmoid kernel, are also used in the expression (5). The sigmoid function can be a non-PSD kernel function of certain specific parameters α and β.[8] Therefore in this case, the dual relationship between the expressions (3) and (5) does not exist, and the kernel function (7) is not valid either. Nevertheless, the sigmoid kernel has been successfully used in many cases and some explanations can be found in Ref.[10]. Furthermore, the application of the non-PSD kernel function in SVM and corresponding learning algorithm are given in Ref.[11] and sigmoid kernel are proved to be PSD under certain constraints.

In order to favour the hardware implementation, a sigmoid function is applied to neural network learning in combination of fuzzy logic method in Ref.[12],

and promising performances are achieved. This motivated proposing the method of applying the fuzzy sigmoid kernel to chaotic time series prediction.

The sigmoid kernel based on fuzzy logic can be represented in the fuzzy linguistic form as follows:

$$K(x_i, x_j) = \begin{cases} -1, & x_i \cdot x_j \text{ is low}, \\ +1, & x_i \cdot x_j \text{ is high}, \\ a \cdot x_i \cdot x_j, & x_i \cdot x_j \text{ is medium}, \end{cases} \tag{13}$$

where a is the constant representing the degree of smoothness of the sigmoid function, as indicated in Fig.1.

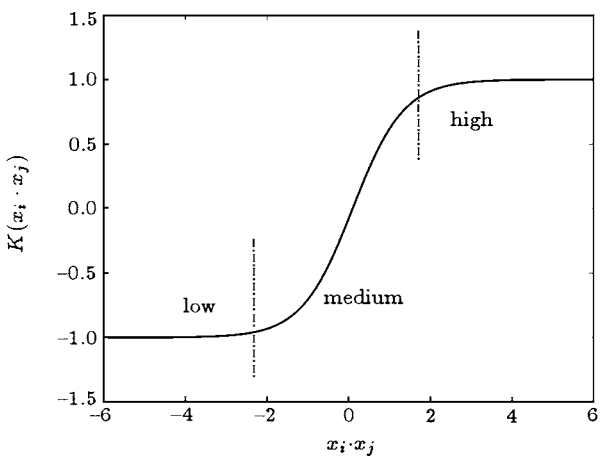


Fig.1. Linguistic variables used in fuzzy kernels.

According to the expression (13), the sigmoid kernel can be defined as a series of membership functions. A triangle membership function is chosen for simplicity, as indicated in Fig.2. Fuzzy memberships are composed of low, medium, and high partitions. Although more detailed partitions can also be obtained such as very low, low, medium, high, and very high, it will increase the computational load. Since the sigmoid kernel must be continuous, the upper and lower boundaries of the triangle membership can be calculated from $-(\beta \pm 1)/\alpha$. Finally, the expression (13) can be written as a function of α and β as follows:

$$K(x_i \cdot x_j) = \begin{cases} -1, & x_i \cdot x_j \le -(\beta+1)/\alpha, \\ +1, & x_i \cdot x_j \ge -(\beta-1)/\alpha, \\ M(x_i, x_j, \alpha, \beta), & \text{else}, \end{cases} \tag{14}$$

where

$$M(x_i, x_j, \alpha, \beta) = 2(x_i \cdot x_j + \beta/\alpha) - \alpha^2 (x_i \cdot x_j + \beta/\alpha)\,|x_i \cdot x_j + \beta/\alpha|\,. \tag{15}$$

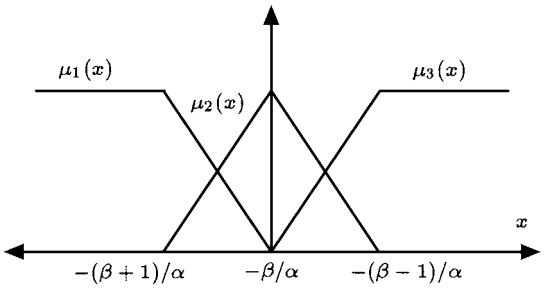


Fig.2. 3 membership functions.

Therefore, the sigmoid kernel can be expressed as follows:

$$K(x_i, x_j) = (-1)\mu_1(x_i \cdot x_j) + a x_i \cdot x_j \mu_2(x_i \cdot x_j) + (+1)\mu_3(x_i \cdot x_j). \tag{16}$$

The fuzzy sigmoid kernel has several advantages.

1) The kernel function is differentiable at any point in the entire region;

2) Membership functions defined in Fig.2 can be easily implemented by a microprocessor, such as DSP;

3) A nonlinear kernel function can be approximated by a number of fuzzy membership functions with different complexities.

3. Examples

Two chaotic time series predictions based on SVM have been carried out to validate the fuzzy sigmoid kernel function proposed in this work. In the experiments, parameter C in SVM and α and β in kernel function are determined using a 10-fold cross validation method. The SVM has been trained using the iterated re-weighted least squares (IRWLS) algorithm,[13] which is more effective than the standard quadratic programming in terms of the execution time and memory consumption. Experiments are performed using a Pentium 2.4-GHz 512-MB computer.

The first example is Mackey–Glass time series prediction. The dynamic equation of Mackey–Glass time series is defined as

$$\dot{x} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t). \tag{17}$$

The above system dynamics shows a chaos state when the initial conditions meet $x(0) = 1.2$, and $\tau = 17$. The goal of chaotic time series prediction is to use the known values of the time series up to the point $x = t$ to predict the value at a certain future time step $x = t + p$. The main idea

of the prediction method is to establish a mapping from a total number of D points in the time series $[x(t-(D-1)\Delta),...,x(t-\Delta),x(t)]$with an interval Δ to a predicted future value $x(t+p)$. Choosing $D=4$, $\Delta=6$ and $p=4$, Eq.(17) can be solved by using the fourth Runge-Kutta method in order to obtain the value of each integral time point $x(t)$. 1000 pairs of input-output points are selected from $x(t)$:

$$[x(t-18), x(t-12), x(t-6), x(t), x(t+4)] \quad (18)$$

where $t=118-1117$, and the first 500 pairs are used as the training data while the last 500 pairs as the testing data.

The prediction results and errors for Mackey–Glass time series with fuzzy sigmoid kernel are shown in Fig.3 and Fig.4, respectively. In order to compare the performance of fuzzy sigmoid kernel with that of standard sigmoid kernel, root mean squared error (RMSE) and average CPU execution time are chosen as test criteria. The RMSE, which is defined as

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}, \quad (19)$$

determines the accuracy of SVM regression model, and the average CPU execution time reflects the implementation efficiency of SVM. It is shown in Table 1 that the average CPU execution time based on the fuzzy sigmoid kernel SVM is reduced significantly (by about 40.1%), although the accuracy has a slight decrease (about 3.6%). A comparison between the proposed method and the method based on Least Square Support Vector Machines (LS-SVM)[6] is also shown in Table 1. It can be seen that the prediction based on the fuzzy sigmoid kernel SVM is more efficient.

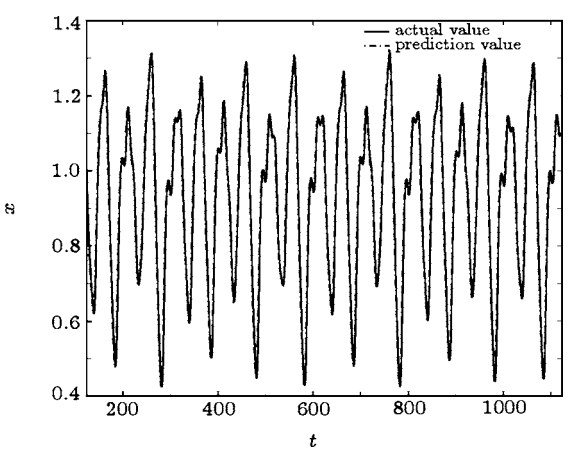


Fig.3. Mackey–Glass time series prediction in t=124–1123.

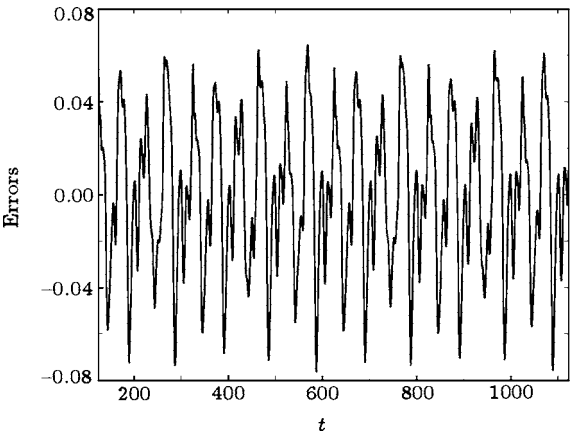


Fig.4. Mackey–Glass time series prediction error.

The second example is Henon time series prediction. The dynamic equation of the Henon time series is defined as

$$x_{n+1}=1-1.4x_n^2+0.3x_{n-1}. \quad (20)$$

Initial values are chosen as $x_0=0.2311$, $x_1=0.9501$, and 1000 pairs of input–output points are picked from x_n based on the expression (18). Similarly, the first and final 500 pairs are used as training data and testing data, respectively. The prediction results and errors are shown in Fig.5 and Fig.6, respectively. Again, the RMSE and average CPU execution time are compared in Table 2. The same conclusion as the above can be obtained, namely the average CPU execution time is reduced significantly (by about 48.4%), although the accuracy has a small decrease (about 5.8%).

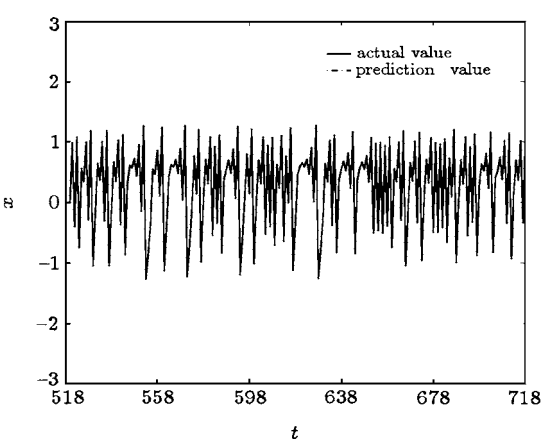


Fig.5. Henon time series prediction in t=518–717.

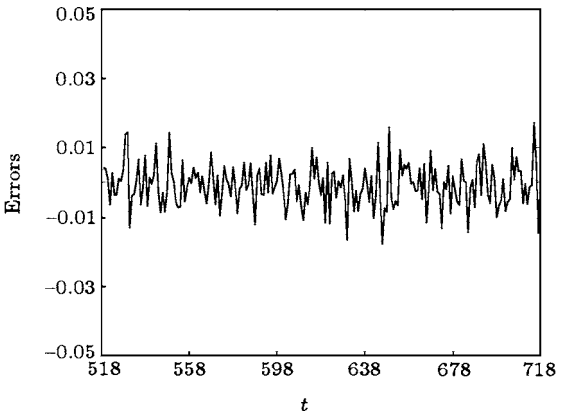


Fig.6. Henon time series prediction error.

Table 1. Comparison of results from different Mackey–Glass time series predictions.

	standard sigmoid	LS-SVM[6]	fuzzy sigmoid
RMSE	0.0107	0.0113	0.0111
CPU time/s	1.87	1.57	1.12

Table 2. Comparison between results from different of Henon time series predictions.

	standard sigmoid	fuzzy sigmoid
RMSE	0.0087	0.0092
CPU time/s	2.13	1.10

4. Conclusions

In this paper the problem of chaotic time series prediction is addressed using the fuzzy sigmoid kernel SVM. The results of two typical chaotic time series predictions show that the average CPU execution time can be reduced significantly, which is favourable for the hardware implementation at the cost of a small decrease in prediction accuracy. The future study is to consider the chaotic time series prediction problem based on the SVM with another kernel, for example, the fuzzy RBF kernel.

References

[1] Liu D, Ren H P and Kong Z Q 2003 *Acta Phys. Sin.* **52** 531 (in Chinese)

[2] Liu H, Liu D and Ren H P 2005 *Acta Phys. Sin.* **54** 4019 (in Chinese)

[3] Dou C X and Zhang S Q 2005 *Chin. Phys.* **14** 902

[4] Takens F 1981 *Lecture Notes in Mathematics* **898** 366

[5] Muller K R, Smola A J, Ratsch G, Scholkopf B and Kohlmorgen J 1997 *Lecture Notes in Computer Science* **1327** 999

[6] Ye M Y and Wang X D 2004 *Chin. Phys.* **13** 454

[7] Courant R and Hilbert D 1953 *Methods of Mathematical Physics* (New York: Interscience/Wiley)

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

[9] Vapnik V 1999 *IEEE Trans. Neural Network* **10** 988

[10] Scholkopf B, Burge C and Smola 1998 *Advances in Kernel Methods: Support Vector Machines* (Cambridge: MIT Press)

[11] Lin H T and Lin C J 2004 *A Study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods* (Technical Report, National TaiWan University)

[12] Soira E, Martin J, Camps G, Serrano A J, Calpe J and Chova L 2003 *IEEE Trans. Neural Networks* **14** 1576

[13] Pérez-Cruz F and Artés-Rodríguez 2001 *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* **2** 1265