On System Identification Based on Online Least Squares Support Vector Machine

Bin Liu[1] **Zhiping Wang**[2]

[1]Department of Computer, Shaanxi University of Science and Technology, Xianyang 712081, P. R. China
[2]Henan Professional College of Economics and Business, Zhengzhou 450053, P. R. China

Abstract

System identification is a fundamental topic of control theory, and LS-SVM has been applied to system identification. An online training algorithm of LS-SVM for system identification is presented, which is suitable for the data set supplied in sequence rather than in batch. The online algorithm avoids computing large-scale matrix inverse when the number of support vectors changes, thus the computation time is reduced. In order to validate the performance of the online algorithm, the system identification experiments are considered. The simulation results show that the online training algorithm is suitable for the online system identification.

Keywords: System identification, Online learning, Least squares support vector machine

1. Introduction

Support vector machine (SVM) is a new universal learning machine in the framework of structural risk minimization (SRM) [1], which has been an active area topic in machine learning community. SVM achieves higher generalization performance than traditional neural networks in many practical applications. SVM uses the kernel functions (linear, Gaussian, polynomial and RBF kernels) to map the data in input space to a high-dimensional feature space where the problem becomes linearly separable [2].

The standard SVM is solved by quadratic programming methods, however these methods are often time consuming and are difficult to implement adaptively [10], and suffer from the problem of large memory requirement and CPU time when trained in batch mode. Least squares support vector machine (LS-SVM) is a modified version of SVM, which uses the equality constraints to replace the original convex quadratic programming problem [3]. Consequently, the global minimizer is much easier to obtain in LS-SVM by solving the set of linear equations. LS-SVM has been applied to classification [4], [5] and control theory [6]-[8].

Conventionally most learning algorithms for LS-SVM is batch learning, in which the input data are supplied and computed in batch [9]. For a specific application that involves a large data set and the data arrive sequentially, that is, online arrive, batch implementations of LS-SVM are inefficient because they must be retrained from scratch when the training set is modified. Therefore, it is not possible to apply LS-SVM for real-time applications, such as signal processing, system identification. In this case, an online training algorithm would be desirable.

Recently several online training algorithms for LS-SVM have been presented [10]-[13]. However, most of these algorithms are only described for classification. In this paper, an online training algorithm for LS-SVM is employed to system identification, which is an extension of the work in [13].

This paper is organized as follows. Section 2 briefly reviews LS-SVM. Section 3 sets up an online training algorithm. Section 4 applies online LS-SVM to system identification. Section 5 summarizes this paper.

2. Least squares support vector machine [6]

Given a training data set of N points $\{x_k, y_k\}_{k=1}^{N}$ with the input data $x_k \in R^n$ and the corresponding target $y_k \in R$. In feature space SVM models take the form

$$y(x) = \omega^T \varphi(x) + b \tag{1}$$

where the nonlinear mapping $\varphi(\cdot)$ maps the input vector into a higher dimensional feature space, b is the bias and ω is a weight vector of the same dimension as the feature space. In LS-SVM for function estimate, the following optimization problem is considered

$$\min_{\omega,e} J(\omega,e) = \frac{1}{2}\omega^T\omega + \frac{1}{2}\gamma\sum_{k=1}^{N} e_k^2 \tag{2}$$

subject to the equality constraints

$$y_k = \omega^T\varphi(x) + b + e_k \ \ , k = 1,\ldots,N \tag{3}$$

here γ is the regularization parameter.

This problem can be solved by using the optimization theory. The Lagrangian function for this problem can be define as follows

$$L(\omega,b,e;\alpha) = J(\omega,e) - \sum_{k=1}^{N}\alpha_k(\omega^T\varphi(x_k)+b+e_k-y_k) \ . \tag{4}$$

In this equation, the α_k's are called the Lagrangian multipliers. The saddle point can be found by setting the derivatives equal to zero

$$\begin{aligned}
\frac{\partial L}{\partial \omega} = 0 \quad &\rightarrow \quad \omega = \sum_{k=1}^{N}\alpha_k\varphi(x_k)\\
\frac{\partial L}{\partial b} = 0 \quad &\rightarrow \quad \sum_{k=1}^{N}\alpha_k = 0\\
\frac{\partial L}{\partial e_k} = 0 \quad &\rightarrow \quad \alpha_k = \gamma e_k\\
\frac{\partial L}{\partial \alpha_k} = 0 \quad &\rightarrow \quad \omega^T\varphi(x_k) + b + e_k - y_k = 0
\end{aligned} \tag{5}$$

for $k = 1,\ldots,N$. According to Karush-Kuhn-Tucker (KKT) conditions, we can eliminate e_k and ω and get the following set of linear equations

$$\begin{bmatrix} 0 & \overrightarrow{1}^T \\ \overrightarrow{1} & \Omega+\gamma^{-1}I \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

where $y = [y_1;\ldots;y_N]$, $\overrightarrow{1} = [1;\ldots;1]$, $\alpha = [\alpha_1;\ldots;\alpha_N]$ and $\Omega_{kl} = \varphi(x_k)^T\varphi(x_l)$ for $k,l = 1,\ldots,N$. According to Mercer's condition, there exists a mapping φ and an expansion

$$K(x,y) = \sum_i \varphi_i(x)\varphi_i(y), \ \ x,y \in R^n \ \ . \tag{7}$$

The resulting LS-SVM model for function estimation becomes

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

where α, b are the solutions of (6). In this paper, for the choice of the kernel function $K(\cdot,\cdot)$, Gaussian kernel can be chosen, $K(x,x^{'}) = exp(-\|x - x^{'}\|^2/2\sigma^2)$.

According to above description, LS-SVM uses equality constraints and can be solved by solving a set of linear equations, while the standard SVM is solved by quadratic programming methods. Therefore, the solution of the standard SVM is more complex and time consuming than that of LS-SVM.

3. Online least squares support vector machine for system identification [13]

According to [13], the online training algorithm for LS-SVM is stated as follows.

Considering LS-SVM model based on the first N pairs of data has been built, and the new data (x_{N+1}, y_{N+1}) is coming in.

In (6), Let

$$\begin{bmatrix} 0 & \overrightarrow{1}^T \\ \overrightarrow{1} & \Omega+\gamma^{-1}I \end{bmatrix} = A_N, \begin{bmatrix} b \\ \alpha \end{bmatrix} = \alpha_N, \begin{bmatrix} 0 \\ y \end{bmatrix} = Y_N. \tag{9}$$

Then (6) is written as

$$A_N\alpha_N = Y_N \Rightarrow \alpha_N = A_N^{-1}YN \tag{10}$$

The subscript N means that the current model is based on the first N pairs of data. For $N+1$ pairs of data, have

$$\alpha_{N+1} = A_{N+1}^{-1}Y_{N+1} \tag{11}$$

where

$$\begin{aligned}
A_{N+1} &= \begin{bmatrix} A_N & b_1 \\ b_2 & c \end{bmatrix}\\
b_1 &= [1 \ K_{1,N+1} \ K_{2,N+1} \ \ldots \ K_{N,N+1}]^T\\
b_2 &= b_1^T\\
c &= K_{N+1,N+1}\\
Y_{N+1} &= \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}
\end{aligned}$$

According to [14], the following lemmas hold.

Lemma 1 . *For matrix* $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, *where* A_{11}^{-1}, A_{22}^{-1} *exist and let* $A_{11}' = [A_{11} - A_{12}A_{22}^{-1}A_{21}]^{-1}, A_{12}' = A_{11}^{-1}A_{12}[A_{21}A_{22}^{-1}A_{12} - A_{22}]^{-1}, A_{21}' = (A_{21}A_{11}^{-1}A_{12} - A_{22})^{-1}A_{21}A_{11}^{-1}, A_{22}' = [A_{22} - A_{21}A_{11}^{-1}A_{12}]^{-1}$, *then the following equation holds*

$$A^{-1} = \begin{bmatrix} A_{11}' & A_{12}' \\ A_{21}' & A_{22}' \end{bmatrix} \tag{12}$$

Lemma 2 . *For matrix A, B, C, D, where A^{-1}, C^{-1} exist, the following equation is true*

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} \quad (13)$$

According to lemma 1 and lemma 2, we can get the following theorem.

Theorem 1 . *The matrix A_{N+1}^{-1} in (11) can be computed from A_N^{-1} without the need of computing the matrix inverse.*

Proof. According to (12),

$$A_{N+1}^{-1} = \begin{bmatrix} A_N & b_1 \\ b_2 & c \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} (A_N - \frac{1}{c}b_1b_2)^{-1} & A_N^{-1}b_1(b_2A_N^{-1}b_1 - c)^{-1} \\ (b_2A_N^{-1} - c)^{-1}b_2A_N^{-1} & (c - b_2A_N^{-1}b_1)^{-1} \end{bmatrix}^{-1}$$

Applying (13) to (14), we have

$$[A_N - \frac{1}{c}b_1b_2]^{-1} = A_N^{-1} - A_N^{-1}b_1[-c+b_2A_N^{-1}b_1]^{-1}b_2A_N^{-1} \quad (14)$$

Let $\Delta = [c-b_2A_N^{-1}b_1]^{-1}$, (14) can changes into:

$$A_{N+1}^{-1} = \begin{bmatrix} A_N^{-1} & 0 \\ o^T & 0 \end{bmatrix} + \Delta \begin{bmatrix} A_N^{-1}b_1 \\ -1 \end{bmatrix} [b_2A_N^{-1} \ \ -1] \quad (15)$$

It is clear that A_{N+1}^{-1} can be computed from A_N^{-1} without the need of computing the matrix inverse.

As A_{N+1}^{-1} in (15) is computed in an incremental way, the expensive inversion operation is avoided. Therefore the corresponding coefficients and bias $\alpha_{N+1} = [b \ \alpha]^T$ can be computed according to (11). Then function estimation can work. Given a new input x, the corresponding function value $y(x)$ can be estimated by (8). So the online algorithm of LS-SVM is set up.

According to [13], the online algorithm is described as following.

Step 1 $Initial()$;
Step 2 $Add\_New\_Training\_Samples(x_{N+1}, Y_{N+1})$;
$[b_1, b_2, c] = Compute\_Parameter(x, y)$;
$A_{N+1}^{-1} = Online\_Add(A_N^{-1}, b_1, b_2, c)$;
$\alpha_{N+1} = Update\_Coefficients(A_{N+1}^{-1}, Y_{N+1})$;
$N = N + 1$;
Step 3 *Return Step* 2.

4. Simulation experiments

To verify the online lS-SVM for system identification, in this section, the experiments are simulated. In these experiments, the Gaussian kernel function is adopted.

Example 1. Consider a nonlinear system.

$$\begin{aligned} f(i) \;=\; & x(i-2)x(i-3) - x(i-1)x(i-2)/2 - \\ & f(i-1)x(i-3) + 0.2f(i-1)f(i-2) + \\ & 0.8x(i-1)f(i-2) + f(i-1)e(i-1)/2 + \\ & 0.1e(i) \end{aligned} \quad (16)$$

where $e(\cdot)$ is Gaussian white noise, its mean is 0 and variance is 0.04. The input $x(i)$ is random series which is independent distributing in $(-1, 1)$. Figure 1 shows the input.

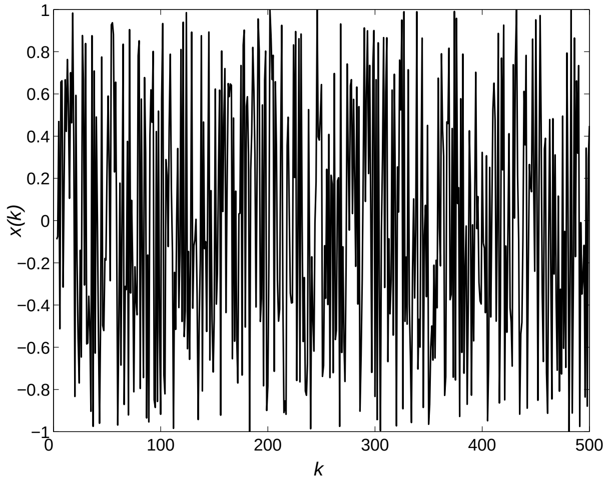

Fig. 1: The input of the nonlinear system.

We take 50 samples as training samples, and then the other samples are sequentially obtained. Figure 2 shows the approximation results obtained using the online LS-SVM.

Example 2. Consider the following system.

$$z(k+1) = \frac{z(k)}{1 + 0.68\sin(0.0005k\pi)z^2(k)} + 0.78u^3(k) \quad (17)$$

where $u(k) = sin(0.01k\pi)$. we takes 20 data points as the training samples, 500 of which are taken as testing samples. The online estimation result is the same as the batch learning, and MSE is 0.0076. But the online algorithm is fast than the batch algorithm.

From the simulation experiment, the online LS-SVM has high efficiency, and is suitable for system identification.

5. Conclusions

System identification plays an important role for control theory and engineering. As the samples in

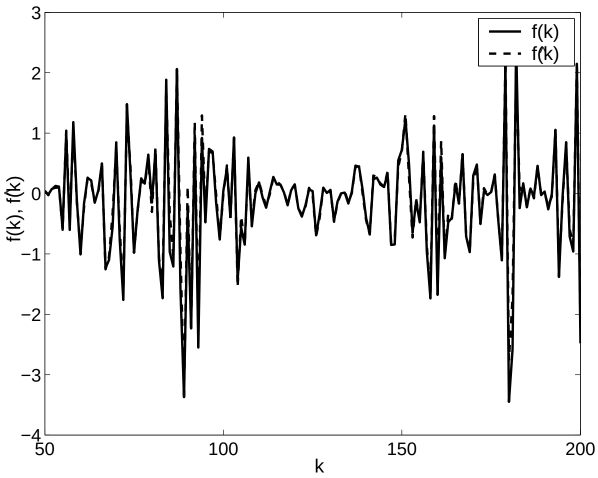

Fig. 2: Original function (solid line) and approximation result by online LS-SVM (dotted line).

system identification are online obtained, the conventional training algorithms of LS-SVM are not suitable for online systems identification. In this paper, an online training algorithm for LS-SVM is employed to system identification. Simulation results show that online LS-SVM has a much faster convergence and is suitable for system identification.

References

[1] V.N. Vapnik, *The ature of Statistical Learning Theory*, Springer-Verlag, 1995.

[2] C.J.C. Burges, Geometry and invariance in kernel based methods. *Proc. of Advance in Kernel Methods-Support Vector Learning*, pp. 89-116, 1999.

[3] H. Chi and O.K. Ersoy, Recursive update algorithm for least squares support vector machines. *Neural Processing Letters*, 17: 165-173, 2003.

[4] J.A.K. Suykens and J. Vandewalle, Least squares support vector machine classifiers. *Neural Processing Letters*, 9: 293-300, 1999.

[5] K.S. Chua, Efficient computations for large least square support vector machine classifiers. *Pattern Recognition Letters*, 24: 75-80, 2003.

[6] J.A.K. Suykens, J.D. Brabanter, L. Lukas and J. Vandewalle, Weighted least squares support vector machines robustness and sparse approximation. *Neurocomputing*, 48: 85-105, 2002.

[7] C. Saunders, A. Gammerman and V. Vovk, Ridge regression learning algorithm in dual variables. *Proc. of the 15th International Conference on Machine Learning*, pp. 515-521, 1998.

[8] J.A.K. Suykens, J. Vandewalle and B.D. Moor, Optimal control by least squares support vector machines. *Neural Networks*, 14: 23-35, 2001.

[9] K.W. Lau and Q.H. Wu, Online training of support vector classifier. *Pattern Recognition*, 36: 1913-1920, 2003.

[10] A. Kuh, Adaptive least square kernel algorithms and applications. *Proc. of International Joint Conference on Neural Networks*, pp. 2104-2107, 2002.

[11] Y. Zhu and Z. Mao, Online optimal modeling of LS-SVM based on time window. *Proc. of IEEE International Conference on Industrial Technology*, pp. 1325-1330, 2004.

[12] Z. Hao, S. Yu, X. Yang, F. Zhao, R. Hu and Y. Liang, Online LS-SVM learning for classification problems based on incremental chunk. *Lecture Notes in Computer Science*, 3173: 558-564, 2004.

[13] J. Liu, J. Chen, S. Jiang and J. Cheng, Online LS-SVM for function estimation and classification. *Journal of University of Science and Technology Beijing*, 10: 73-77, 2003.

[14] G.H. Golub and L.C.F. Van, *Matrix computation*, Johs Hopkins University Press, 1989.