http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (2): 370-373, 2008
ISSN 1812-5638

An Improved Algorithm on Least Squares Support Vector Machines

[1,2]Wang Liejun, [2]Lai Huicheng and [1]Zhang Taiyi
[1]Department of Information and Communication Engineering,
Xi'an Jiaotong University, People's Republic of China
[2]School of Information Science and Engineering, Xinjiang University, People's Republic of China

Abstract: The Least Squares Support Vector Machines (LS-SVM) is an improvement on the Support Vector Machines (SVM). Combined the LS-SVM with the Multi-Resolution Analysis (MRA), an improved algorithm-the Multi-resolution Least Squares Support Vector Machines (MLS-SVM) algorithm is proposed in this study. With better approximation ability, the proposed algorithm has the same theoretical framework as the MRA. At a fixed scale the MLS-SVM is a classical LS-SVM. However, the MLS-SVM can gradually approximate the target function at different scales. In experiment, the MLS-SVM is used as nonlinear system's identification, with better identification accuracy achieved.

Key words: Support vector machines, least square method, multi-resolution analysis, nonlinear system identification

INTRODUCTION

Support Vector Machines (SVM) comprises an important class of learning algorithms based on statistical learning theory (Vapnik, 1995, 1999). By minimizing the upper bound of expectation risk, SVM can trade-off the model complexity and empirical risk, enjoying high generalization performance (Vapnik *et al*., 1997). Moreover, the kernel functions which meet Mercer's theory can be incorporated into SVM, so the nonlinear problems in low dimensional space can be solved in linear scheme in the corresponding high dimensional space (Burges, 1998; Suykens and Vandewalle, 1999). As an improvement on SVM, least squares support Vector machines (LS-SVM) takes the equality constraints in place of the inequality counterparts with SVM and employs the quadratic loss function instead of Vapnik's ε insensitive one (Scholkopf *et al*., 1999). So the question can be settled from the solution to a set of linear equations in LS-SVM instead of a time-consuming quadratic programming with the classical SVM (Williamson *et al*., 1999). After that, a recurrent neural network version of LS-SVM has been introduced (Mallat, 1989) and the sparseness of LS-SVM also has been investigated (Williamson *et al*., 1999).

One of the earliest reports of multi-resolution SVM (M-SVM) was made by Shao. Different from the classical SVM with a single kernel, M-SVM uses several kernels of different scales to approximate the target function in a progressive fashion: making use of a wide kernel to approximate the target function at a coarse level; then with a narrower kernel to approximate the residual parts at a more refined level. From the view of the Multi-Resolution Analysis (MRA), the wide kernel and the narrow kernel in Shao and Vladimir (1999) M-SVM can be seen as the scale function and the wavelet function respectively. The wide kernel approximation and the narrow kernel approximation correspond to the smooth approximation and smooth details in the MRA. However, Shao's method needs to be viewed as several limitations in mind. Firstly, it is not consistent with the theory framework of the MRA. This is because the Gaussian kernel in the M-SVM is not a strict wavelet function. In addition, the wide kernel and the narrow kernel don't satisfy dilation equations. Consequently the subspace spanned by them can not be segmented under the MRA theory. Secondly, the definition of the scale in the M-SVM is ambiguous because there is not an explicit expression between the width of the kernel function and the scale of the subspaces. Motivated by Shao's method, the multi-resolution LS-SVM (MLS-SVM) is presented in this study.

LS-SVM AND MRA

The LS-SVM for approximation: The scheme behind the LS-SVM for approximation is reviewed here. The basic idea of the LS-SVM is: mapping the data set $\{x_n, y(n)\}_{n=1}^{N}$ into a high dimensional feature space F via a nonlinear mapping function φ and linear approximation

Corresponding Author: Wang Liejun, School of Information Science and Engineering, Xinjiang University, People's Republic of China

can be done in this space. Thus linear approximation in F is equivalent to nonlinear approximation in low dimensional input space $L_2(R)$ and the linear approximation hyperplane in F is defined as:

$$\tilde{f}(x_n) = w^T \varphi(x_n) + b \tag{1}$$

An optimization problem can be derived when introducing slack variables ξ_n.

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{n=1}^{N}\xi_n^2 \\ &\text{s.t. } y(n) - w^T\varphi(x_n) - b = \xi_n, n = 1,2,\cdots N \end{aligned} \tag{2}$$

In (2) C is a regularization constant and b is a threshold. By constructing a Lagrange function from both the objective function and the corresponding constraints, the resultant problem becomes:

$$\min W(\alpha) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n\left(y(n) - w^T\varphi(x_n) - b - \xi_n\right) \tag{3}$$

where, α_n is Lagrange multiplier. The values of α and b can be solved from:

$$\begin{bmatrix} 0 & L^T \\ L & \chi^T\chi + C^{-1}I \end{bmatrix}\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \tag{4}$$

where, $L = (1,1,..., 1)^T$, $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)^T$, $Y = (y_1, y_2, .., y_N)^T$ and $\chi = (\varphi(x_1), \varphi(x_2), ...\varphi(x_N))$. At last the expression of w and linear approximation hyperplane are obtained with

$$w = \sum_{n=1}^{N}\alpha_n\varphi(x_n), \tilde{f}(x_n) = \sum_{i=1}^{N}\alpha_i K(x_n, x_i) + b \tag{5}$$

where, K is defined by

$$K(x_n, x_i) = \varphi(x_n) \bullet \varphi(x_i) \tag{6}$$

K is so called kernel function, which satisfying the Mercer conditions. The approximation criterion of LS-SVM is:

$$\tilde{f}(x_n) = \arg\min_{g\in F}\left\{\frac{C}{2}\sum_{n=1}^{N}\left[y(n) - g(x_n)\right]^2 + \frac{1}{2}\|g\|_F^2\right\} \tag{7}$$

where, $\|\bullet\|_F$ denotes the norm defined on the F.

The multi-resolution analysis (Zhang *et al.*, 2001): Supposing MRA is a multi-resolution analysis such that there exist the sub-spaces V_j and W_j. $\phi_{jk}(x) = 2^{-j/2}\phi(2^{-j}x - k)$ and $\Psi_{jk}(x) = 2^{-j/2}\Psi(2^{-j}x - k)$, which are orthogonal basis of V_j and W_j subspaces, respectively and satisfy dilation equation:

$$\begin{aligned} \phi(2^{-j}x) &= \sqrt{2}\sum_k h_{0k}\phi(2^{-j+1}x - k) \\ \psi(2^{-j}x) &= \sqrt{2}\sum_k h_{1k}\psi(2^{-j+1}x - k) \end{aligned} \tag{8}$$

Then the projection of $f(x)$ on V_{j-1} can be computed as follows:

$$P_{j-1}f(x) = P_jf(x) + D_jf(x) \tag{9}$$

$P_jf(x)$ is smooth approximation of $f(x)$ at resolution j and $D_jf(x)$ is smooth details. They can be expressed:

$$P_jf(x) = \sum_k c_k^{(j)}\phi_{jk}(x), D_jf(x) = \sum_k d_k^{(j)}\psi_{jk}(x) \tag{10}$$

where, $c_k^{(j)}$ and $d_k^{(j)}$ are discrete approximation and discrete details of f (x) at resolution j, formulated as:

$$c_k^{(j)} = \left\langle f(x), \phi_{jk}(x)\right\rangle, d_k^{(j)} = \left\langle f(x), \psi_{jk}(x)\right\rangle \tag{11}$$

The improved algorithm in the framework of multi-resolution analysis: The MLS-SVM algorithm is derived in the subspace V_j. The linear approximation hyper-plane in V_j is elaborated as the following form:

$$\tilde{f}_j(x_n) = \sum_k c_k^{(j)}\phi_{jk}(n) + b \tag{12}$$

Now let $\tilde{f}_j(x_n)$ approximate f (x) at the scale j in the sense of approximation criterion (7). For convenience (7) can be represented as:

$$\min_{\tilde{f}} H(\tilde{f}) = \frac{C}{2}\sum_{n=1}^{N}\left[y(n) - \tilde{f}(x_n)\right]^2 + \frac{1}{2}\|\tilde{f}\|_{V_j}^2 \tag{13}$$

Where:

$$\|\tilde{f}\|_{V_j}^2 = \sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}}$$

is following the Ref. (Scholkopf *et al*., 1999).
After substituted into (13), the above problem becomes:

$$\min_{\tilde{f}} H(\tilde{f}) = \frac{C}{2}\sum_{n=1}^{N}\left[y(n) - \tilde{f}(x_n)\right]^2 + \frac{1}{2}\sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} \tag{14}$$

An additional set of variables ξ_n is introduced, the optimization problem turns into:

$$\begin{aligned} &\min_{\tilde{f}} H(\tilde{f}) = C\sum_{n=1}^{N}\xi_n^2 + \frac{1}{2}\sum_k \frac{c_k^{(j)2}}{\lambda_k^{(j)}} \\ &\text{s.t. } y(n) - \tilde{f}(x_n) = \xi_n, n = 1, \cdots N \end{aligned} \tag{15}$$

For the purpose of solving the above constrained minimization problem, the technique of Lagrange multipliers is used and the lagrangian corresponding to the above problem is defined:

$$W = C\sum_{n=1}^{N}\xi_n^2 + \frac{1}{2}\sum_k \frac{c_k^{(j)^2}}{\lambda_k^{(j)}} + \sum_{n=1}^{N}\alpha_n^{(j)}\left(y(n) - \tilde{f}(x_n) - \xi_n\right) \quad (16)$$

where, $\alpha_n^{(j)}$ is Lagrange multiplier. The $c_k^{(j)}$ can be obtained by setting the derivatives with respect to $c_k^{(j)}$ zero, that is $\frac{\partial W}{\partial c_k^{(j)}} = 0$

$$c_k^{(j)} = \lambda_k^{(j)}\sum_{n=1}^{N}\alpha_n^{(j)}\phi_{jk}(n) \quad (17)$$

Substituting (17) into (12) one arrives at:

$$\tilde{f}_j(x_n) = \sum_{i=1}^{N}\alpha_i^{(j)}K_j(x_n, x_i) + b \quad (18)$$

In (18) the kernel function K_j is given by:

$$K_j(x,y) = \sum_k 2^{-j}\phi(2^{-j}x - k)\phi(2^{-j}y - k) \quad (19)$$

The numerical calculation of Eq. 19 is important to MLS-SVM. In calculation, the function ϕ we choose is the scaling function of the Daubechies wavelet with order 6 (DB6) and order 10 (DB10).

RESULTS AND DISCUSSION

Here the standard questions about nonlinear system identification are taken to verify the MLS-SVM algorithm. The Mean Squares Error (MSE) is used as evaluating the performance.

Considering the nonlinear system with a single input and a single output, which output y(n) and input u(n) can be described as the following difference equation

$$y(n+1) = \frac{y(n)}{1 + y^2(n)} + u^3(n) \quad (20)$$

where, u(n) is a random input whose amplitude is uniformly distributed in the interval [-1,1]. The classical input can be

$$u_{n1} = 0.8\sin(2\pi t/200) + 0.2\cos(2\pi t/40) \quad (21)$$

$$u_{n2} = \begin{cases} \sin(2\pi t/200), t \le 500 \\ 0.8\sin(2\pi t/200) + 0.2\sin(2\pi t/40), t > 500 \end{cases} \quad (22)$$

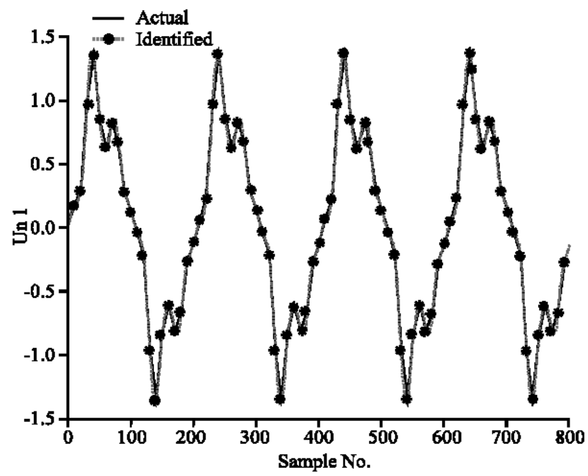

Fig. 1: The actual values (—) and identification values (-*-) with Db10 and the scale 0 with u_{n1}

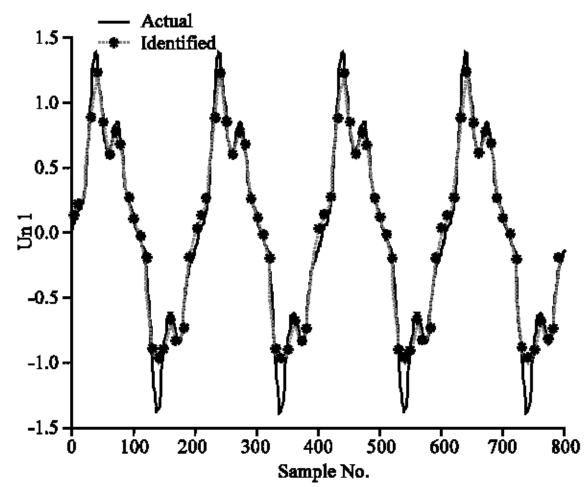

Fig. 2: The actual values (—) and identification values (-*-)with Db10 and the scale 2 with u_{n1}

Equations 20 and 21 can be used as generating 800 data points, from which 80 points are selected in equidistance. The chosen points constitute the training samples. Now MLS-SVM is performed for the system identification. Moreover, the identification result of the signal at the scale j(j = 0, 1, 2) can be obtained. In MLS-SVM the chosen parameters c is 10^6.

The identification results on the conditions of the scales 1, 2 are shown in Fig. 1 and 2, where the solid line is for the actual values and the dotted line with asterisks for the identification values. The corresponding identification MSE is shown in Table 1.

The same procedures can be carried out in the system with input u_{n2}. The identification MSE is shown in Table 2.

Table 1: The identification MSE with different scale kernel and different scales in the u_{n1} system for MLS-SVM

	Different kernel	
The scale j	DB10 ($\times 10^{-4}$)	DB6 ($\times 10^{-4}$)
0	2.7172	3.7902
1	114.00	138.05
2	1025.26	1456.72

Table 2: The identification MSE with different scale kernel and different scales in the u_{n2} system for MLS-SVM

	Different kernel	
The scale j	DB10 ($\times 10^{-4}$)	DB6 ($\times 10^{-4}$)
0	1.3156	1.1878
1	77.829	64.661
2	459.49	584.14

As shown from the results, the identification values fit the actual values very well. In view of Fig. 1 and 2, no difference can be noticed. It can be found that the MLS-SVM can approximate the target function gradually at different scale. For a fixed kernel, the smaller the scale j is, the smaller of the identification MSE of the MLS-SVM is, which is consistent with the conclusion in MRA. However, the MSE of MLS-SVM of DB10 scale kernel is usually smaller than that of DB6 scale kernel at a fixed scale j. It should be pointed out that the different system should be identified with the different scale kernels. To cope with the smooth output system, the higher order Daubechies Wavelets will be used, which compactly supporting scale function has wide interval.

CONCLUSION

Combined the LS-SVM with the MRA, a new algorithm (MLS-SVM) is proposed in this study. New algorithm has the same theoretical framework as the MRA with better approximation ability. At a fixed scale, the MLS-SVM is a classical LS-SVM. However, the MLS-SVM can approximate the target function gradually at different scale. From the implemented identification of nonlinear system, it is evident that the proposed algorithm is feasible and has smaller MSE.

REFERENCES

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2 (2): 23-34.

Mallat, S., 1989. A theory of multiresolution signal decomposition. The wavelet transform. IEEE. Trans., PAMI., 11 (7): 674-693.

Scholkopf, B., S. Mika and C.J.C. Burges, 1999. Input space versus feature space in kernel-based methods. IEEE. Trans. Neural Networks, 10 (5): 1000-1017.

Shao, X. and C. Vladimir, 1999. Multi-resolution support vector machine. International Joint Conference on Neural Networks IJCNN, Washington DC, pp: 1065-1070.

Suykens, J.A.K. and J. Vandewalle, 1999. Least squares support vector machine classifiers. Neural Process. Lett., 9 (2): 293-300.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York.

Vapnik, V., S. Golowich and A.J. Smola, 1997. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. Neural Information Processing System, MIT Press.

Vapnik, V., 1999. An overview of statistical learning theory. IEEE. Trans. Neural Networks, 10 (5): 988-999.

Williamson, R., A. Smola and B. Scholkopf, 1999. Entropy Numbers, Operators and Support Machine Kernel, in Advances in Kernel Methods-Support Learning. Cambridge, M.A., MIT Press, pp: 127-144.

Zhang, Z., G. Liu and D. Enqing, 2001. De-noising via dyadic wavelet transform. J. Elect. Inform. Technol., 23 (11): 1083-1090.