Improved Sparse Least Square Support Vector Machines for the Function Estimation

Yafeng.Zhang, Songnian.Yu

Shanghai University, ShangHai, 200072,China

zhangyafeng6362@gmail.com

***Abstract*—Least square support vector machines (LS-SVMs) are deemed good methods for classification and function estimation. Compared with the standard support vector machines (SVMs), a drawback is that the sparseness is lost in the LS-SVMs. The sparseness is imposed by omitting the less important data in the training process, and retraining the remaining data. Iterative retraining requires more intensive computations than training the non-sparse LS-SVMs. In this paper, we will describe a new pruning algorithm for LS-SVMs: the width of ε–insensitive zone is introduced in the process of the training; in addition, the amount of the pruning points is adjusted according to the performance of training, not specified using the fixed percentage of training data; furthermore, the cross training is applied in the training. The performance of improved LS-SVMs pruning algorithm, in terms of computational cost and regress accuracy, is shown by several experiments based on the same data sets of chaotic time series.**

Keywords-LS-SVMs,improved sparse LS-SVMs, ε–insensitive zone, pruning, function estimation

I. INTRODUCTION

Support vector machines (SVMs) have been introduced for classification and function estimation by Vapnik (1995), which are based on VC dimension and structural risk minimization. SVMs use the kernel to deal with the nonlinear problems, where the kernel is to map training data to the high-dimension by the map function ¢. At the same time, SVMs use the quadratic programming to deal with convex optimization problems. Therefore, for the SVM, the computation is complex [7, 8]. To solve the problem, the least square support vector machines, which use a set of liner equation constrains instead of computational costly quadratic programming problem, have been investigated by Suykens. The optimization problem reduces to solving a linear set equation. Thus, LS-SVMs are more computationally attractive and easier to realize. However, the shortcoming of LS-SVMs is that sparseness is lost [7, 8, 11]. We know, the sparseness not only improve model's generalization ability, but also reduce the training time. So it is important to implement the sparseness of LS-SVMs.

To impose sparseness in the LS-SVMs solution, several pruning algorithms were investigated. Suykens *et al.* proposed a simple pruning algorithm based on the smallest weighted support values [6,7]. Kruif *et al.* presented a more sophisticated pruning method to omit the samples that bear the smallest error in the next pass [5]. Zeng *et al.* presented a SMO-based pruning method [4]. Jiao *et al.* presented a new fast sparse approximation for LS-SVMs using the back fitting scheme [1]. Although, the size of the linear system decreases from step to step, pruning is much more costly than non-sparseness LS-SVMs.

In this paper, we present a new pruning method to improve Suykens' pruning algorithm, which is more efficient at the function estimation than other LS-SVMs pruning methods. The rest of the paper is organized into six sections. In section 2, we will introduce the LS-SVMs algorithm. In section 3, we present the pruning method. And in Section 4, we provide experimental results and discussion for function estimation. In last section, we conclude the paper.

II. LS-SVMS FOR FUNCTION APPROXIMATION

In this section, we will review the basic principles of LS-SVMs for the function approximation. Suppose a training data set $\{x_k, y_k\}_{k=1}^{N}$ has been given, where $x_k \in R_n$ represents an n-dimensional input vector and $y_k \in R$ is the corresponding target. So the LS-SVMs model is defined in the original space by

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

where $\phi(\bullet)$ is a nonlinear function, which puts the data points into a high dimensional Hibert space, b is the bias, and ω is a weighted vector[6,7,8].

In the LS-SVMs prediction model, ω and b are estimated by minimizing a primal space error cost function:

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

$$\text{S.T } y_k = \varphi(x_k) + b + e_k, k = 1,.....N \,, \tag{2}$$

where e_k is the error. In order to resolute this optimization problem, the Lagrange function is given by

$$L(\omega,b,e,\alpha) = J(\omega,e) - \sum \alpha_k \{\omega^T \varphi(x_k) + b + e_k - y_k\} \tag{3}$$

with the Lagrange multipliers α (support values)[7]. According to optimization conditions, we can obtain Karush-Kuhn-Tucker(KKT) System as follows:

$$\frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{k=1}^{N} \alpha_k \varphi(x_k)$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k = 0$$

978-0-7695-4077-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICICTA.2010.269

$$\frac{\partial L}{\partial e_k} = 0 \to \alpha_k = \gamma e_k (k = 1, \ldots\ldots N) \tag{4}$$
$$\frac{\partial L}{\partial \alpha_k} = 0 \to \omega \varphi^T (x_k) + b + e_k - y_k = 0 \cdot$$

By eliminating *e, ω,* we can attain the following set of linear equations

$$\begin{pmatrix} 0 & \tilde{1}^T \\ \tilde{1} & \Omega + I/\gamma \end{pmatrix} \begin{pmatrix} b \\ \tilde{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{y} \end{pmatrix}, \tag{5}$$

where $\tilde{y} = [y_1, y_2, y_3, \ldots\ldots, y_{N-1}, y_N]^T$ ($\tilde{y}$ is a vector with target value), $\tilde{1} = [1, 1, 1, \ldots\ldots, 1, 1]^T$, $\tilde{a} = [a_1, a_2, a_3, \ldots\ldots, a_{N-1}, a_N]^T$, I is a square matrix, $\Omega_{i,j} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ for $i,j = 1, 2, \ldots\ldots, N$ and $K(x_i, x_j)$ is the kernel function that satisfies Mercer's condition [1,4, 7,8] .

So we can obtain the LS-SVMs model for foundation estimation

$$y(x) = \sum_{k=1}^{N} \alpha_k \varphi(x, x_k) + b \,, \tag{6}$$

where $\tilde{a}$, b are solved by (5) [6,7, 8].

III. PRUNING ALGORITHM

A. Suykens' pruning algorithm

Compared with the standard SVMs, the drawback of LS-SVMs is that the sparseness is lost. The sparseness came up due to the choice of the *2*-norm and be also revealed by the fact that the support values are proportional to errors *($a_k = \gamma e_k$)* at the data points. Thus the training error is reposed by the support values [7, 8, 10]. The Suykens' pruning algorithm is imposed by omitting irrelevant points in the training sets, and re-estimating the LS-SVMs as follow:

a) train LS-SVMs on the *N* points,
b) remove some points with the smallest support values (e.g. 5% of the set) according to the values of the a_k,
c) retrain the LS-SVMs based on the remaining data points,
d) go to section *b* until the performance degrades.

This procedure corresponds in fact to the pruning of the LS-SVMs [7, 8,11]. In fact, pruning is an iterative scheme where in each step one has to resolute a KKT system. Although, the size of the linear system decreases from step to step, pruning is much more costly than non-sparseness LS-SVMs. To the pruning algorithm, computational cost and performance are related to the determination of the pruning points and the implication of the iterative retraining. In the following section, we will describe a new method to resolute the determination of pruning points in each retraining of the LS-SVMs [7, 8, 10].

B. Improved pruning algorithm

To the determination of pruning points, the criterion plays on an important role. In the Suykens' pruning algorithm, the amount of pruning points is specified using the fixed percentage of training data in each retraining, and it cannot be adjust according to the performance. Thus it is inefficient and inflexible [4,7]. In addition, the data set of removed from the training set is given up on the later training, it is not reasonable exactly. In this section, we will detail a new criterion that is automatically adjusting the amount of the training data set according to the training performance, and using the cross-training to train the data set.

1) Amount of removing the data points

To standard SVMs, the sparseness is achieved by the use of the ε–insensitive loss function, where error smaller than ε is ignored and parameter ε control the width of ε–insensitive zone. The value of ε has an effect on the smoothness of SVMs' response and affects the number of support vectors used to construct the regression function. Therefore, the complexity and the generalization capability of the SVMs depend on the choice of ε [4, 5, 9]. In addition, it is known that the value of ε should be proportional to the input noise level, that is $\varepsilon \propto \sigma$ (σ is the standard deviation of noise)[1, 3, 9]. Furthermore, in the Suykens' pruning, it is not difficult to find that the omitting of the data points implicitly corresponds to creating a ε–insensitive zone, because they have similar effects [2]. According to this, it may be not difficult to find that the determination of pruning points has an important influence on achieving sparseness in LS-SVMs. Meanwhile, the determination of pruning points is proportional to the training error. Hence, on the same training environment, the efficiency of the network is better on the good training data set than the bad, and there are more insignificant points in the data set, when the performance is descend to the beforetime. To derive the suitable criterion for pruning points, we put the theory of the ε–insensitive zone into the pruning of data set, which the amount of data points is proportional to the training error. Therefore, we remove amount of unimportant points on the bigger scale of the sample, when the training performance fall, On the contrary, we remove on the smaller scale.

Along with this idea, we suppose the training data set N_i been given, and the assessing standards of the training performance is defined using the mean squared error ($V_i = \sqrt{\frac{\varepsilon_{i1}^2 + \varepsilon_{i2}^2 + \cdots\cdots + \varepsilon_{in}^2}{n}} = \sqrt{\frac{\sum_{j=1}^{n} \varepsilon_{ij}^2}{n}}$, n is the amount of the data point in the *ith* training), so the pruning function is rewritten using the definition of M_{i+1}

$$M_{i+1} = \left(1 + \frac{V_i - V_{i-1}}{V_i + V_{i-1}}\right) \times S \times N_i \tag{7}$$

where V_i is the mean squared error, S is the defined present of removing data.

2) Cross training

In the Suykens' pruning algorithm, when the data points are put into the removing data set, it will not affect on the later training. In fact, some data points of the removed data set maybe affect the performance of training on the later training. If the training set and the parts of the removed data set are both used at the same time, it will be helpful for the

training [1, 2]. Therefore, in this paper, the cross-training is applied on the training. Given a training data set *P*, a removed data set *Q*, a removing data set *S*, a data set *T*(*T* is consist of the data points removing from the data set *Q* according to some criterion) , the cross training is defined as follow:

a) train the LS-SVMs on the training data set *P*,
b) remove the data points from the *P* according to the defined criterion, and add the data points to *S*, then *P=P-S,*
c) choose the data points from the *Q* according to the defined criterion, and add to the *T,* then *P=P+T* and *Q=Q-T+S.*

The procedure is repeated until defined stopping criterion is satisfied.

3) The improved pruning algorithm

According to the criterion of amount of removing data points and the cross-training, the pruning algorithm is showed as follows:

a) train the initial non-sparse LS-SVMs as described on the section *b* on *N* points,
b) repeat the *c* and *d*, unless the user-defined performance index is satisfied,
c) remove points according the step as follow:
   (1) compute the mean squared error V_i , and Compute the amount of pruning points,
   (2) remove data points with the smallest support values in the sorted $|a_k|$ spectrum, and add to the data points to *S,* then *P=P-S*,
   (3) choose the data points from the *Q* according to the same criterion, and add to the *T*, then *P=P+T*, and *Q=Q-T +S*,
d) retrain the LS-SVMs based on the remaining data set *P*.

In the improved pruning algorithm, the amount of pruning data points is defined as (9) not according the fixed percentage of training data. It is obvious that the pruning data set is proportion to the performance of training so it can adjust the retraining data set flexibly, and speed up the retraining. Meanwhile the cross training is applied on the training, so it is not difficult to find that the performance indicator is improved using the removed data set and training data set at the process of training.

IV. EXPERIMENT

We will perform an experiment for the improved LS-SVMs pruning procedure, which is based on chaotic time series data set. In this experiment, we consider the chaotic time series data generated by the Mackey-Glass delay-differential equation:

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\Delta)}{1+x(t-\Delta)^{10}} \qquad (8)$$

where the Δ=17，Δ=30[12]. We denote the time series by MGS17, MGS30. Fig1 presents the 1000 points of the time series MGS17 and MGS30.

In this paper, we test the performance of improved LS-SVM pruning algorithm (ILS-SVMs), the Suykens' pruning algorithm (SLS-SVM) and Jiao' pruning algorithm (JLS-SVM) using the data set MGS17 and MGS30. In order to compare briefly, we use the Gaussian Radial Basis kernel ($K(x_i, x_j) = \exp(\left|x_i - x_j\right|^2 / 2\partial^2)$) as the kernel function where the σ=3, γ=10. The performance indicator is tested using the root mean squared error ($RMSE = \sqrt{\frac{1}{N}\sum_{I=1}^{N}\left(y_i - \hat{y}_i\right)^2}$) [6,7, 8]. We will divide the each data set into two parts. First 500 points were used as training set, and the others for testing. In the table1 and table 2, we show the training time, the testing time, and the number of support vectors and the prediction accuracy of SLS-SVMs, JLS-SVMs, and ILS-SVMs on the data set MGS17 and MGS30. Meanwhile, in the figer3, the accuracy of the SLS-SVMs, JLS-SVMs, and ILS-SVMs will be showed on MGS17.

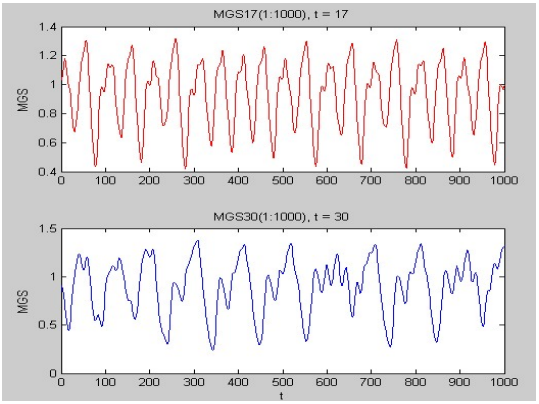


Figure1. The data set of chaotic time series

TABLE1: TRAINING COMPUTATIONAL COSTS, TESTING COMPUTATIONAL COSTS, THE TITLE OF SUPPORT VALUES AND RMSE FOR THE SLS-SVMS, JLS-SVMS, AND ILS-SVMS USING THE DATA SET MGS17

	Training time (s)	Testing time (s)	The title of Support values	RMSE
SLS-SVMs	7.030	0.6406	400	0.0305
JLS-SVMs	6.910	0.5407	315	0.0227
ILS-SVMs	6.540	0.4932	225	0.0245

TABLE2: TRAINING COMPUTATIONAL COSTS, TESTING COMPUTATIONAL COSTS, THE TITLE OF SUPPORT VALUES AND RMSE SLS-SVMS, JLS-SVMS, AND ILS-SVMS USING THE DATA SET MGS30

	Training time (s)	Testing time (s)	The title of Support vectors	RMSE
SLS-SVMs	7.064	0.5786	421	0.0312
JLS-SVMs	6.934	0.5023	324	0.0207
ILS-SVMs	6.785	0.4853	265	0.0216

The table1 and the table2 present that compared to the other pruning algorithm, the ILS-SVMs reduce the amount of support vectors, resolve the sparseness problem and increase generally capacity of the model on the same training

condition. In addition, the prediction accuracy of ILS-SVMs is slightly better that that of SL-SVMs, and is a little bit worse than the JLS-SVMs, however, the training time and testing time of ILS-SVMs is significantly shorter than that of SL-SVMs, JLS-SVMs on these data sets. The figer3 also show that the prediction accuracy of ILS-SVMs is slightly better that that of SL-SVMs, and is a little bit worse than the JLS-SVMs. Above all, we can conclude that the improved pruning algorithm is fit for the function estimate perfectly, and it has well generally capacity and estimate accuracy.

V. CONCLUSION

LS-SVMs are successfully approach to function estimation. However, there are two obvious limitations. First, their computation is complex. Second, the solution of LS-SVMs lost the sparseness. So this paper describes a new pruning algorithm for LS-SVMs: this algorithm is based on the width of the ε–insensitive zone; in addition, a new criterion, which automatically adjusts the amount of the training data set according to the training performance, is proposed to determination the amount of pruning data points in the LS-SVMs algorithm; furthermore, the cross training, which using removed data points reasonable in the training process, not only reused the removed data points in the training, but also improved the performance. Several experiment results, based on the chaotic time series, showed that the improved LS-SVMs pruning algorithm has not only high estimation accuracy but also has very fast training speed.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No.60873129, Shanghai Leading Academic Discipline Project under Grant No.J50103.

REFERENCES

[1] Licheng Jiao, Liefeng Bo, "Fast Sparse Approximation for Least Squares Support Vector Machines", IEEE Trans.on Neural Netw, 2007,18(3): 685-697.

[2] Y.-J. Lee, S.-Y. Huang, "Reduced support vector machines: A statistical theory", IEEE Trans.on Neural Netw, 2007, 18(1): 1–13.

[3] JIANG Tian-han, SHU Jiong, "Multi-step Prediction of Chaotic Time series Using the Least Squares Support Vector Machines", *Control and Decision*, 2006, 21(1).

[4] Zeng, X.-w. Chen, "SMO-based pruning methods for sparse least squares support vector machines", IEEE Trans.on Neural Netw, 2005, 16(6): 1541–1546.

[5] B.J.de Kruif, T.J.de Vries, "Pruning error minimization in least squares support machines", IEEE Trans.on Neural Netw, 2003, 14(3): 696-702.

[6] B. Harmers, J.A.K. Suykens, B. De Moor, "A comparison of iterative methods for least squares support vector machine classifiers", Internal report 01-110, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2001.

[7] J.A.K. Suykens, J.Vandewalle,"Least squares support vector machine classifiers", Neural Processing Letters, 1999, 9(3): 293-300.

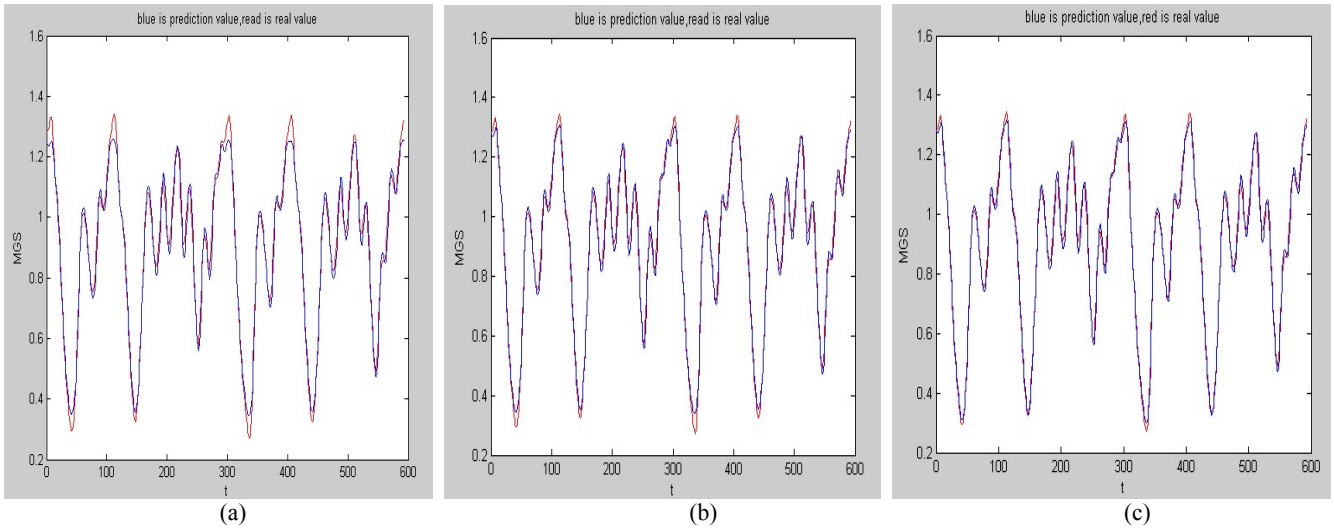
[8] B.Baesens, S.Viaene, T.Van Gestel, J.A.K.Suykens, "An empirical assessment of kernel type performance for least squares support vector machine classifiers", Proceeding of 4th International Conference on Knowledge-Based Intelligent Engineering Systems& Allied Technologies, 313-316.

[9] Scho¨lkopf, B. Smola, "New support vector algorithms", Neutral Computation, 2000, 12(4): 1207–1245.

[10] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using a support vector machine", Neural Networks for Signal Processing VII—Proc. 1997 IEEE Workshop, 1997, 511–520.

[11] J.A.K. Suykens, L. Lukas, J.Vandewalle, "Sparse approximation using least squares support vector machines", IEEE International Symposium on Circuits and Systems (ISCAS 2000), 2000, II757-II760.

[12] Nicholas I.Sapankevych, Ravi Sankar, "Time series prediction using support vector machines: a survey", Computer and Information Science, 2009,4(2): 24-38.



(a) (b) (c)

Figure2. Prediction for Mackey-Glass time series of SLS-SVMs, JLS-SVMs, and ILS-SVMs using the data set MGS17.(a) The result of SLS-SVMs, (b) the result of JLS-SVMs, (c) the result of ILS-SVMs