Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Improved Training of An optimal Sparse Least Squares Support Vector Machine

Xiao-Lei (Celina) Xia, Kang Li & George W. Irwin

School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast, Belfast BT9 5AH, UK,
(Tel: 44-28-90974164; e-mail: {xxia01, k.li, g.irwin}@qub.ac.uk)

Abstract: The optimal separating hyperplane of a typical Least Squares Support Vector Machine (LS-SVM) is constructed using most of the training samples. A consequent disadvantage is the slowdown of the LS-SVM classification process on the test samples. Previous methods address this issue by simplifying the decision rule established after training, which risks a loss in generalization ability and imposes extra computation cost. This paper presents a novel optimal sparse LS-SVM whose decision rule is parameterized by the optimal set of training examples, in addition to having an optimal generalization capability. For a large number of classification problems, the new LS-SVM requires a significantly reduced number of training samples, a property referred to as the sparseness of the solution. The training of the LS-SVM method is implemented using a modified two-stage regression algorithm. Experiments on two-spiral data confirms the advantages described. Simulation results on checkerboard data further illustrate that the proposed LS-SVM can effectively produce an optimal hyperplane which is sparse in training examples.

1. INTRODUCTION

Recently, a new formulation of the standard SVM - Least Squares SVM (LS-SVM) [Suykens and Vandewalle (1999)], has been proposed to avoid the use of the quadratic programming (QP) involved in training. This considers only equality constraints rather than the inequality ones in a standard SVM. As a result, an LS-SVM classifier can be built by solving a set of linear equations. Empirical studies have suggested that LS-SVMs have very competitive generalization abilities to standard ones [Gestel et al. (2004)].

As with standard SVMs, the decision hyperplane, resulting from the training of an LS-SVM, is constructed from a set of weighted training samples. The SVM solution generally requires a small number of training samples. However, in an LS-SVM the training samples, which are the parameters of the solution, usually constitute a large proportion of the whole training set, in contrast to the sparseness of the SVM solution[Müller et al. (2001)]. Consequently, an LS-SVM involves more computation, which slows the class label prediction on test samples. Several alternative approaches have been proposed to address this issue, which search for good approximate solutions. One alternative formulation is parameterized by a much-reduced number of vectors, and not necessarily by the training samples only. Unfortunately, the actual implementation of the algorithms can be very time-consuming. More specifically, approaches in Burges (1996), Downs et al. (2001), Schölkopf et al. (1999), Schölkopf and Smola (2002) are all computationally very expensive due to the iterative searches involved. For example, the one proposed in Downs et al. (2001) is particularly unrealistic due to its dependency on the feature space which is usually of high, or even infinite dimension.

In this paper, an optimal sparse LS-SVM is proposed which can simultaneously locate the optimal separating hyperplane and identify the optimal set of training samples for the training of its parameters. These goals are achieved by an adaption of the original LS-SVM algorithm, leading to a good generalization performance and a compact formulation. Training is efficiently implemented by application of a two-stage nonlinear regression algorithm [Li et al. (2005), Li et al. (2006)], originally used in system identification. The resultant regression function is the best approximation to the input data in terms of the least sum-squared errors. Additionally, one parameter of the regression (the dimension of the weighting vector) is adjustable. This parameter, which is based on the optimal set of training samples required for the optimal hyperplane, is tuned by ten-fold cross validation at the expense of kernel based pre-processing of the training data. The class labels of -1 and 1 are interpreted as the target values for the regression. Experiments on difficult classification tasks such as the two spiral [Fahlman (1993)] and checkerboard datasets [Ho and Kleinberg (1996)] suggest that the optimal sparse LS-SVM implemented by two-stage regression maintains good generalization capacity and is a competitive alternative to conventional LS-SVMs. Simulation results on the checkerboard data further show that, the proposed LS-SVM can effectively produce an optimal hyperplane which is sparse in the number of training samples required.

The paper is organized as follows. Section 2 introduces the LS-SVM algorithm in addition to a brief review of the essential ideas of standard SVMs. The optimal sparse LS-SVM, realized by a modified two-stage linear regression, is detailed in Section 3. Section 4 gives the experimental results obtained from application to the two-spiral and checkerboard datasets. The paper is concluded in Section

5, which also addresses the use of the proposed algorithm for large-scale learning tasks.

2. FROM SVMS TO LEAST SQUARE SVMS

2.1 SVM Classifiers

Given n pairs of training samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x} \in R^N$ and $y \in \{-1, 1\}$, training a SVM to locate the optimal hyperplane leads to the following quadratic programming (QP) problem:

$$\min_{\mathbf{w},b,\mathbf{e}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$$
$$\xi_i \in R \geq 0 \quad i = 1, \ldots, n$$

where the regularization constants $C \in R > 0$ and $b \in R$. The function ϕ maps training data to a feature space so that they are linearly separable.

The optimal hyperplane is formulated as

$$\mathbf{w} = \sum_{i=1}^{N_s}\alpha_i y_i \phi(\mathbf{x}_i) \tag{2}$$

where $\alpha_i \in R > 0$ and N_s is the number of training samples x_i which corresponds to $\alpha_i > 0$.

The introduction of a kernel function enables the implicit mapping of the training data from the input space to the feature space without requiring knowledge of ϕ. One of the most widely-used kernels is the Gaussian radial basis function (RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2} \tag{3}$$

In this case the decision function of the SVM becomes:

$$\begin{aligned} f(\mathbf{x}) &= \text{sign}(\mathbf{w}^T\phi(\mathbf{x}) + b) \\ &= \text{sign}(\sum_{i=1}^{N_s}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x})) \end{aligned} \tag{4}$$

An important attribute of the solution (4) is its sparseness in α_i — N_s,the number of non-zero α_i is relatively small compared to n, the total number of training data samples. On the other hand, (4) implies that N_s is a major factor in the computational complexity of SVM classification on test samples.

2.2 Least-Squares SVM Classifiers

The least-squares SVM is the solution to the following optimization problem (1) as:

$$\min_{\mathbf{w},b,\mathbf{e}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\sum_{i=1}^{n}e_i^2 \tag{5}$$
$$\text{s.t.} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) = 1 - e_i \quad i = 1, \ldots, n$$

Introducing Lagrange multipliers $\alpha_i (i = 1, \ldots, n)$ for each of the equality constraints gives:

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{e}, \alpha) &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\sum_{i=1}^{n}e_i^2 \\ &\quad - \sum_{i=1}^{n}\alpha_i\{y_i[\mathbf{w}^T\phi(\mathbf{x}_i) + b] - 1 + e_i\} \end{aligned} \tag{6}$$

Due to the equality constraints, α_i can either be positive or negative according to the Karush-Kuhn-Tucker (KKT) conditions [Fletcher (1987)].

The optimality of (6) requires that:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \longrightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i) \tag{7}$$

$$\frac{\partial L}{\partial b} = 0 \longrightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{8}$$

$$\frac{\partial L}{\partial e_i} = 0 \longrightarrow \alpha_i = \gamma e_i, \quad i = 1, \ldots, n \tag{9}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \longrightarrow y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) = 1 - e_i \tag{10}$$

With the adoption of the equality rather than the inequality constraints, the training of an LS-SVM reduces to solving a linear system. Empirical studies have shown that LS-SVMs provide competitive generalization performance to SVMs.

However, in most cases, the total number of non-zero α_i in (7) could be as large as n. Thus the separating hyperplane in an LS-SVM is spanned by a large proportion of the training data set from (4). This non-sparseness in the solution for LS-SVMs directly produces a prolonged classification procedure on test samples.

3. A NOVEL SPARSE LS-SVM USING TWO-STAGE REGRESSION

To solve the non-sparsity of LS-SVM solutions, an optimal set LS-SVM is developed by reforming the set of linear equations produced by conventional LS-SVMs. The aim of the novel LS-SVM is two-fold: (1) to locate the optimal separating hyperplane; (2) to identify the optimal set of the training examples which construct the hyperplane. This is achieved using a two-stage algorithm, initially designed for use in the system identification. With appropriate preprocessing of the input data, this can be effectively used to train an optimal LS-SVM such that the non-sparseness issue is less severe than conventional ones. This optimal sparse LS-SVM is now introduced, followed by a brief look at the two-stage regression approach and a detailed description of its use in the new LS-SVM algorithm.

3.1 A Novel Sparse LS-SVM Algorithm

In the optimization formulation of (5), the orientation w of the separating hyperplane is optimized but not its location b. To address this issue, the term b^2 is appended to (5) according to Mangasarian and Musicant (1999). Also note that the equivalence of the equality constraint in (5) is:

$$\mathbf{w}^T\phi(\mathbf{x}_i) + b = y_i - e_i \tag{11}$$

Hence the optimization formulation becomes:

$$\min_{\mathbf{w},b,\mathbf{e}} \quad \frac{1}{2}(\mathbf{w}^T\mathbf{w}+b^2)+\gamma\sum_{i=1}^{n}e_i^2 \tag{12}$$
$$\text{s.t.} \quad \mathbf{w}^T\phi(\mathbf{x}_i)+b=y_i-e_i \quad i=1,\ldots,n$$

where γ is the penalty constant.

The Lagrangian of (12) is:

$$L'(\mathbf{w},b,\mathbf{e},\alpha) = \frac{1}{2}(\mathbf{w}^T\mathbf{w}+b^2)+\gamma\sum_{i=1}^{n}e_i^2 - \sum_{i=1}^{n}\alpha_i[\mathbf{w}^T\phi(\mathbf{x}_i)+b-y_i+e_i] \tag{13}$$

The optimality of (13) leads to the following conditions:

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i) \tag{14}$$
$$\sum_{i=1}^{n}\alpha_i=b \tag{15}$$
$$\alpha_i=\gamma e_i \tag{16}$$
$$\mathbf{w}^T\phi(\mathbf{x}_i)+b=y_i-e_i \tag{17}$$

To ease the non-sparseness of (14) in α, the optimal sparse LS-SVM seeks $m(m << n)$ training examples $x_i(i=1,2,\ldots,m)$, and corresponding m weights $\beta_i(i=1,2,\ldots,m)$ to reformulate the norm vector w and the bias term b of the optimal hyperplane:

$$\mathbf{w}=\sum_{j=1}^{m}\beta_j\phi(\mathbf{x}_j) \tag{18}$$
$$b=\sum_{j=1}^{m}\beta_j=\sum_{i=1}^{n}\alpha_i \tag{19}$$

Note that, although each training example $\mathbf{x}_i$ doesn't necessarily satisfy (16), the following equation can still be drawn from (15) and (19):

$$\sum_{j=1}^{m}\beta_j=\gamma\sum_{i=1}^{n}e_i \tag{20}$$

The sparse LS-SVM is also subject to:

$$\mathbf{w}^T\phi(\mathbf{x}_i)+b=y_i-e_i, \quad i=1,2,\ldots,n \tag{21}$$

These conditions lead to a linear system which is formulated as:

$$\begin{bmatrix} E & 0 & 0 & -\Phi_m \\ 0 & -1 & 0 & I^T \\ 0 & 0 & -1 & I^T \\ \Phi_n^T & E & E & 0 \end{bmatrix}\begin{bmatrix} \mathbf{w} \\ b \\ \mathbf{e} \\ \beta \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{y} \end{bmatrix} \tag{22}$$

where $\Phi_n=[\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_n)], \Phi_m=[\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_m)]$, $\mathbf{y}=[y_1,\ldots,y_n]^T$, $\mathbf{e}=[e_1,\ldots,e_n]^T$, $\beta=[\beta_1,\ldots,\beta_m]^T$, I is an m-column vector of 1s and E is an unity matrix of appropriate dimension.

The linear equations can be further simplified to:

$$\left[\begin{array}{c|c} I^T+H & \gamma^{-1} \\ \hline I^T & -1 \end{array}\right]\begin{bmatrix} \beta \\ b \end{bmatrix}=\begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \tag{23}$$

where $H=\Phi_n^T\Phi_m$.

Using the kernel technique, a Hessian matrix H is introduced:

$$H_{ij}=\phi(x_i)^T\phi(x_j)=K(x_i,x_j) \tag{24}$$

Training the optimal sparse LS-SVM then amounts to solving (23), which relies on the use of a two-stage algorithm from the field of system identification.

3.2 Two-Stage Regression [Li et al. (2005), Li et al. (2006)]

This algorithm originates from nonlinear system identification, though in broad terms, it can be considered an linear regression approach. Applying two-stage in linear regression also enables the size of the model to be varied in that the algorithm can simultaneously perform feature selection on the input data. The two-stage method finds a model of a pre-set size, with the least approximation errors for the input data which contains only the selected features. The algorithm is now described from a regression perspective.

Consider the problem of approximating a set of n data pairs, $(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)$ where $\mathbf{x}\in R^d(d\geq n)$ and $y\in R$. The model size (or features in the classification context) is supposed to be k, where k can be either predetermined or determined by a model (feature) selection criterion along the regression process. The two-stage algorithm can effectively select k $(k\leq d)$ features and a corresponding weight vector which contains k coefficients to construct a regression function:

$$f(\mathbf{x})=\mathbf{w}^T\mathbf{x}'+b \tag{25}$$

Here $\mathbf{x}'\in R^k$ with k selected features from $\mathbf{x}\in R^d$ and $\mathbf{w}=[c_1,\ldots,c_k]^T\in R^k$. The loss function for the regression model is then:

$$J=(\mathbf{P}\mathbf{w}-\mathbf{y})^T(\mathbf{P}\mathbf{w}-\mathbf{y}) \tag{26}$$

where $\mathbf{P}=[\mathbf{x'}_1,\ldots,\mathbf{x'}_n]^T=[\mathbf{p}_1,\ldots,\mathbf{p}_k]\in R^{n\times k}$, $\mathbf{y}=[\mathbf{y}_1,\ldots,\mathbf{y}_n]^T\in R^n$ is the target value vector.

For a data matrix $\mathbf{P}$ whose rank is k, an estimate of $\mathbf{w}$ which minimizes J is:

$$\mathbf{w}=(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{y} \tag{27}$$

Starting from (27), the two-stage algorithm searches for the optimal regression function in two steps. First, one feature is chosen at each step to update $\mathbf{P}$. Without loss of generality, assume that the feature index chosen is the m-th one:

$$\mathbf{P}_m=[\mathbf{p}_1,\ldots,\mathbf{p}_m] \tag{28}$$

where $\mathbf{p}_j\in R^n$ $(j=1,\ldots,m)$ is the j-th selected column vector from the input data matrix $\mathbf{X}$. Using (26) and (27), the loss function at this stage is :

$$J(\mathbf{P}_m)=\mathbf{y}^T[\mathbf{y}-(\mathbf{P}_m^T\mathbf{P}_m)^{-1}\mathbf{P}_m^T\mathbf{y}] \tag{29}$$

Next $\mathbf{p}_{m+1}$ is selected from the remaining $(d-m)$ column vectors of X, to minimize $\mathbf{R}(\mathbf{P}_{m+1})$ where $\mathbf{P}_{m+1} = (\mathbf{P}_m, \mathbf{p}_{m+1})$. Thus:

$$\min_{\mathbf{p}_{m+1}} J(\mathbf{P}_m, \mathbf{p}_{m+1}) \tag{30}$$

where $\mathbf{p}_{m+1} \in \mathbf{X}$ and $\mathbf{p}_{m+1} \notin \mathbf{P}_m$.

Finally, k column vectors are chosen from the input data $\mathbf{X}$, producing a matrix $\mathbf{P}_k$. Meanwhile, $\mathbf{X}$ is split into ($\mathbf{P}_k$, $\mathbf{P}_{d-k}$), where $\mathbf{P}_k$ consists of the selected vectors in the input matrix, while $\mathbf{P}_{d-k}$ represents the unselected ones.

It should be noted that the first step is a forward subset selection approach and the result is not optimal. In other words, the selected vectors are not sparse enough due to the correlations between vectors. Therefore, the second part of the two-stage algorithm optimizes $\mathbf{P}_k$, replacing previously selected $\mathbf{p}_i$ vectors with new column vectors from $\mathbf{P}_{d-k}$ to produce a further reduction in $J(\mathbf{P}_k)$. At each step, a $\mathbf{p}_i (i = 1, \ldots, k)$ is selected from $\mathbf{P}_k$ for examination and replaced by a $\mathbf{p}_j (j = 1, \ldots, d-k)$ from $\mathbf{P}_{d-k}$ in order to compare the estimation performances of their resulting $\mathbf{P}_k$. The original $\mathbf{p}_i$ vector is finally replaced by the one from $\mathbf{P}_{d-k}$ which produces less estimation errors than $\mathbf{p}_i$ and the smallest errors of all the column vectors from $\mathbf{P}_{d-k}$. Otherwise, the $\mathbf{p}_i$ being examined stays unchanged. This process ensures an optimal regression function which contains the weight vector $\mathbf{w}$ and selected input data.

3.3 Training of Optimal Sparse LS-SVMs

Equation (23) is equivalent to:

$$(\mathbf{I}_{n\times m} + \mathbf{H})\beta + \mathbf{e} = \mathbf{y} \tag{31}$$

where $\mathbf{I}_{n\times m}$ is an n-by-m matrix of ones.

Prior the application of the two-stage regression method to (31), the training data $\{\mathbf{x}_i, y_i\}$ where $i = 1, \ldots, n$ and $y \in \{-1, 1\}$ are preprocessed, producing a matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]^T \in R^{n\times n}$:

$$D_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + 1 \tag{32}$$

In addition to the Gaussian RBF in (3), alternatives for the kernel function K include:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^p \tag{33}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i^T \mathbf{x}_j + \delta) \tag{34}$$

where (33) is a polynomial with degree up to p and (34) describes a two-layer sigmoidal neural network.

Replacing K in (32) with a specific kernel function, for example, the Gaussian RBF, gives:

$$D_{ij} = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2} + 1 \tag{35}$$

The problem of finding the optimal hyperplane with the norm vector of (18) is then reduced to approximating the data set $(\mathbf{d}_1, y_1), \ldots, (\mathbf{d}_n, y_n)$ by generating a model of a designated size $m (m \leq n)$. The two-stage linear regression algorithm can be immediately applied, which results in k selected column indexes of $\mathbf{D}$ and a weight vector $\beta = (\beta_1, \ldots, \beta_m)$ with β_i for the i-th chosen column. These indexes are, in fact, those of m chosen examples from the original training set $\mathbf{x}_i (i = 1, \ldots, n)$ used to construct the separating hyperplane of the optimal sparse LS-SVM.

Denoting these m training samples $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_m)$, the resulting optimal sparse LS-SVM classifier is:

$$f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{m} \beta_i K(\mathbf{s}_i, \mathbf{x}) + b) \tag{36}$$

where $b = \sum_{i=1}^{m} \beta_i$.

More specifically, in the case of a Gaussian RBF, the decision rule on a test sample $\mathbf{x}$ is:

$$f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{m} \beta_i e^{-||\mathbf{s}_i - \mathbf{x}||^2/2\sigma^2} + \sum_{i=1}^{m} \beta_i) \tag{37}$$

The parameters σ, m in (37) can be determined by leave-one-out cross validation. Note that the tuning of the parameter γ in (12) becomes a search of the optimal model size m, and the two-stage algorithm is used to implement optimal sparse LS-SVMs.

4. EXPERIMENTS AND RESULTS

The proposed optimal sparse LS-SVM was evaluated using two classification problems supplied by the two-spiral and checkerboard datasets.

4.1 Two-Spiral Classification

The two-spiral dataset is available from the Carnegie Mellon repository [Fahlman (1993)]. This has two groups of spiral-shaped data points, with 97 points for each, as illustrated by Fig. 1 in which "+" and "-" classes are designated separately by a cross and a circle. The Gaussian RBF is chosen as the kernel function for the experiments in this paper. For the optimal sparse LS-SVM, the parameter settings are: $\sigma = 1$ for the width of the RBF and the optimal model size, or the number of training data points used to construct the separating hyperplane, is 192, which were obtained using the ten-fold cross validation. The standard LS-SVM was also implemented to classify the two spirals, with parameter settings of $\sigma = 11.33$ in the RBF kernel and $\gamma = 0.16$ in (12). The solution is parameterized by 194 training data points, as shown in Fig. 2. Fig. 3, Fig. 4 and Fig. 5 show the two-spiral optimal sparse LS-SVMs with model sizes of 192, 120 and 72 respectively. In these figures, the black solid line is the optimal separating hyperplane. It reveals that even though the "perfect" model size is 192, an optimal sparse LS-SVM with good generalization performance can be achievable from a model of, as few as, 72 training samples. These results demonstrate that the proposed LS-SVM algorithm can effectively improve the sparseness of the solution with desirable generalization performance. The Receiver Operating Characteristic (ROC) curves of the three classifiers are shown in Fig. 6, indicating that they are all very good classifiers.

4.2 Checkerboard Classification

The checkerboard dataset [Ho and Kleinberg (1996)] contains 1,000 data points of two categories, as shown in

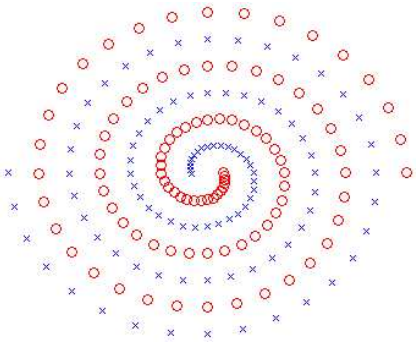

Fig. 1. Two-spiral Dataset

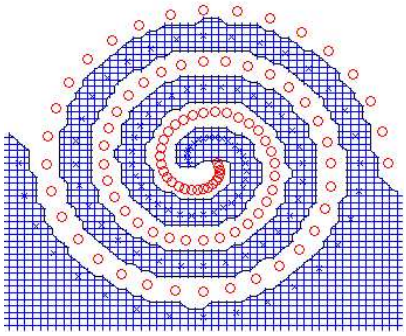

Fig. 2. Two-spiral Conventional LS-SVM Classifier constructed from 192 training data points

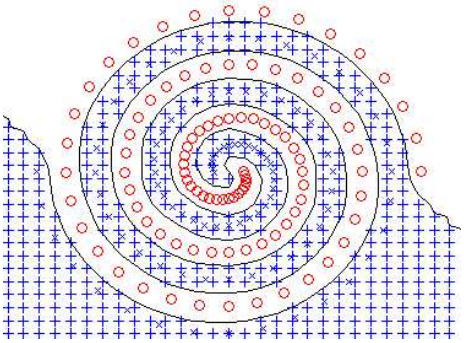

Fig. 3. Two-spiral optimal sparse LS-SVM classifier constructed from 192 training data points

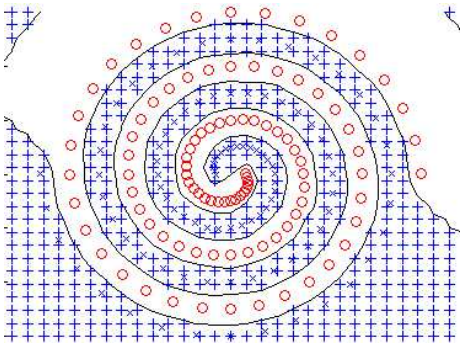

Fig. 4. Two-spiral optimal sparse LS-SVM classifier constructed from 120 training data points

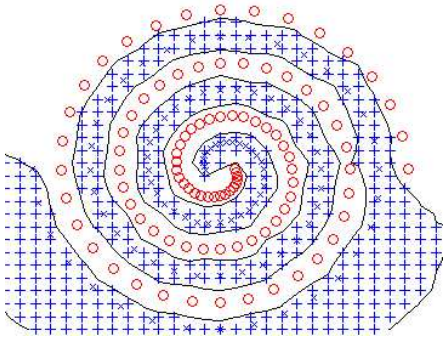

Fig. 5. Two-spiral optimal sparse LS-SVM classifier constructed from 72 training data points

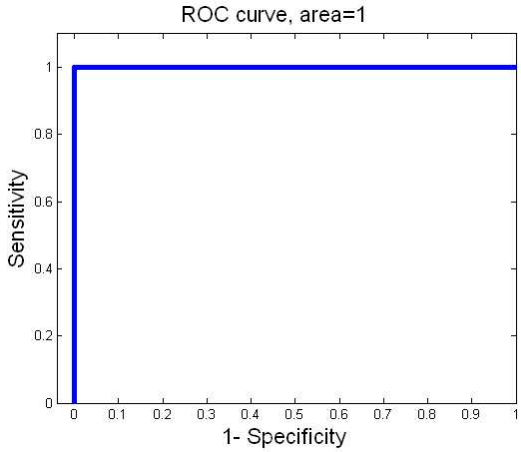

Fig. 6. The ROC Curve for two-spiral optimal LS-SVMs parameterized by over 72 training samples

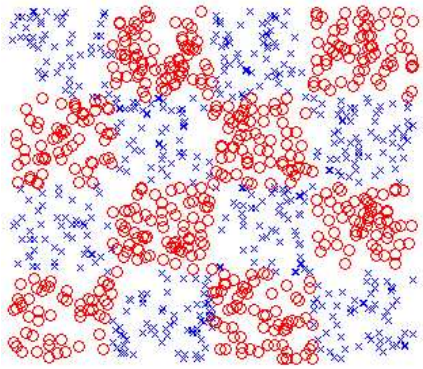

Fig. 7. Checkerboard Dataset

Fig. 7. For the standard LS-SVM, parameter settings derived from an exhaustive grid search are: $\sigma = 1.6372$, $\gamma = 0.12416$. The results from the conventional LS-SVM classifier are depicted in Fig. 8 where the shaded regions represent the "cross" category. The resulting separating hyperplane is constructed from the whole 1000 training data points.

The training of the optimal sparse LS-SVM leads to a classifier constructed from 100 data points with the RBF width $\sigma = 0.06$, as illustrated by Fig. 9. The optimal sparse LS-SVM successfully reduced the training samples of the solution by 90%, while maintaining an excellent generalization capacity.

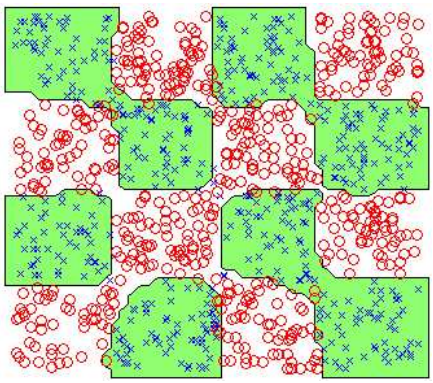

Fig. 8. Checkerboard standard LS-SVM classifier with 1000 training points to parameterize the solution

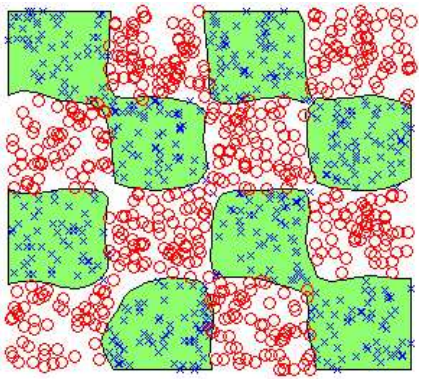

Fig. 9. Checkerboard optimal sparse LS-SVM classifier with 100 training points to parameterize the solution

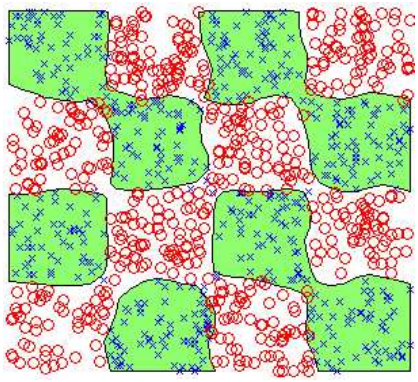

Fig. 10. Checkerboard optimal sparse LS-SVM classifier with 70 training points to parameterize the solution

In fact the sparseness of the optimal LS-SVM solution can be further improved, as demonstrated by Fig. 10 whose hyperplane is built from only 70 training examples with $\sigma = 0.02$. Although its generalization performance is not the optimum, it would provide a satisfactory alternative where test speed, rather than prediction accuracy, is required.

5. DISCUSSIONS AND CONCLUSIONS

This paper proposes a novel algorithm for training an optimal sparse Least Squares Support Vector Machine (LS-SVM). Its decision rule is parameterized by the optimal set of training samples, which provides the optimal generalization capability. Its training is achieved by using a modified two-stage regression algorithm from Li et al. (2005) and Li et al. (2006). Simulation results suggest that the optimal sparse LS-SVM can ease the non-sparseness problem of conventional LS-SVM, without any loss the generalization capability.

The computation of the Hessian matrix for preprocessing the input data, as required for training an optimal sparse LS-SVM, has a major influence on its overall complexity. In this paper, with a lack of prior knowledge, each training sample is considered equally as a candidate to span the decision hyperplane, as the kernel between each pair of training samples is computed. The associated Hessian matrix is square. However, the issue of how many training samples seems to outweigh that of which exact training sample should be selected, as indicated the experiments. This property allows for the simplification of Hessian. A subset of training samples can first be appointed as parameters in the solution, a kernel function is then computed for each sample pair, one being the original full set, the other from the subset. The optimal size of the subset can be found from a grid search. The square Hessian matrix thus reduces to be a rectangle, enabling the algorithm to become more applicable to large scale learning tasks.

REFERENCES

C. J. C. Burges. Simplified support vector decision rules. In L. Saitta, editor, *Proc. of Proc. 13th Int'l Conf. Machine Learning*, pages 71–77, San Mateo, CA, 1996.

T. Downs, K. E. Gates, and A. Masters. Exact simplification of support vector solutions. *J. Mach. Learn. Res.*, 12:293–297, 2001.

R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, Chichester and New York, 1987.

T. Van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle. Benchmarking least-squares support vector machine classifiers. *Machine Learning*, 54(1):5–32, 2004.

T. K. Ho and E. M. Kleinberg. Checkerboard dataset, 1996.

K. Li, J.-X. Peng, and E. Bai. A two-stage algorithm for identification of nonlinear dynamic systems. *Automatica*, 42(7):1189–1197, 2006.

K. Li, J.-X. Peng, and G. W. Irwin. A fast nonlinear model identification method. *IEEE Trans. Automat. Control.*, 50(8):1211–1216, 2005.

O. L. Mangasarian and D. R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10:1032–1037, 1999.

K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12 (2):181–202, 2001.

B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. Miiller, G. Gitsch, and A. J. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

J. A. K. Suykens and J. Vandewalle. Least-squares support vector machine classifiers. *Neural Processing Letters*, 9 (3):293–300, 1999.