A Sparse Twin SVM for multi-classification problems

HONG-XING YAO

Faculty of Science，Jiangsu University，Zhenjiang，Jiangsu 212013，China;

Faculty of Finance and Economics，Jiangsu University，

Zhenjiang，Jiangsu 212013，China

XIAO-WEI LIU

Faculty of Science，Jiangsu University，Zhenjiang，Jiangsu 212013，China

Abstract—We propose Sparse TSVM, a multi-class SVM classifier that determines k nonparallel planes by solving k related SVM-type problems. The Sparse TSVM promotes Twin SVM to one-versus-rest approach. And it capture classes' main feature better with the sparse algorithm. On several benchmark data sets, Sparse TSVM is not only fast, but shows good generalization.

Index Terms—Data Mining, pattern classification, machine learning, sparse, Twin support vector machine.

1 INTRODUCTION

Standard support vector machines (SVMs), which are powerful tools for data classification, classify 2-category points by assigning them to one of two disjoint halfspaces in either the original input space of the problem for linear classifiers, or in a higher dimensional feature space for nonlinear classifiers [1], [2], [3], [4]. However, real world problems often require the discrimination for more than two categories. Thus, the multi-class pattern recognition has a wide range of applications including Optical Character Recognition [5], Intrusion Detection [6], Speech Recognition [7], and Bioinformatics [8]. Actually, the problem of pattern recognition for the complex information systems comes down to categorization issues [9]. Qi Wu and Rob Law has used SVM in complex nonlinear fault system classifying problems [10]. To improve the classification accuracy, Li Zhang and Wei-Da Zhou proposed a density- induced margin support vector machines [12]. Also, for the large computational complexity of multi-class problem, a simplified multi-class support vector machine with reduced dual optimization [13] and Sparse Multi-class Least-Squares Support Vector Machine [14] are presented to speed-up multi-class training process.

In this paper, we propose a simple but effective nonparallel plane classifier, termed as the Sparse Twin Support Vector Machine (STSVM) for multi-class classification. It followed the TWSVM [11] idea and applied it to multi-class problem. Our algorithm is one-versus-rest approach, however, do not become unbalanced when there are many more samples of some classes than others. What's more, instead of training all the given samples, we proposed a algorithm which yields sparse data points together with its weight to describe the initial training set approximately. After Sparsification, training became very faster while maintaining a higher classification accuracy.

The paper is organized as follows: Section 2 briefly dwells on SVMs, and section 3 on twin support vector machine for binary data classification. Section 4 discusses our multi-classifier, and

give its higher dimensional form with kernel function. Section 5 deals with experimental results and Section 6 contains concluding remarks.

2 SUPPORT VECTOR MACHINES

Let the patterns to be classified be denoted by a set of m row vectors A_i $(i = 1,2,\ldots,m)$ in the n -dimensional real space R^n, where $A_i = (A_{i1}, A_{i2}, \cdots, A_{in})$. Also, let $y_i \in \{1,-1\}$ denote the class to which the i th pattern belongs. We first consider the case when the patterns belonging to the two classes are strictly linearly separable. Then, we need to determine $w \in R^n$ and $b \in R$ such that

$$A_i w + b \geq 1 \quad \text{for} \quad y_i = 1, \text{ and } A_i w + b \leq -1 \quad \text{for} \quad y_i = -1. \tag{1}$$

The plane described by

$$w^T x + b = 0 \tag{2}$$

lies midway between the bounding planes given by

$$w^T x + b = 1 \quad \text{and} \quad w^T x + b = -1. \tag{3}$$

and separates the two classes from each other with margin of $1/\|w\|_2$ on each side. In other words, the margin of separation between the two classes is given by $2/\|w\|_2$. Here, $\|w\|_2$ denotes the L_2 norm of a vector w. Data samples which lie on the planes given by (3) are termed as support vectors. The maximum margin classifier, which is the standard SVM, is obtained by maximizing this margin and is equivalent to the following problem:

$$\text{(SVM1)} \quad \min_{w,b} \ \frac{1}{2} w^T w$$

$$s.t. \quad A_i w \geq 1 - b \quad \text{for} \quad y_i = 1, \text{ and } A_i w \leq -1 - b \quad \text{for} \quad y_i = -1. \tag{4}$$

When the two classes are not strictly linearly separable, there will be an error in satisfying the inequalities (1) for some patterns and we can modify (1) to

$$A_i w + q_i \geq 1 - b \quad \text{for} \quad y_i = 1, \text{ and } A_i w - q_i \leq -1 - b \quad \text{for} \quad y_i = -1. \ q_i \geq 0, i = 1,2,\ldots,m, \tag{5}$$

where q_i denotes the error variable associated with the ith data sample. In this case, the classifier is termed as a " soft margin " one, and it approximately classifies points into two classes with some error. The classification of a given test sample x is obtained by determining the sign of $w^T x + b$. The soft margin depends on the value of the nonnegative error variables q_i . In this case, one needs to choose a trade-off between the margin and the error and the standard SVM formulation for classification of the data points with a linear kernel is given by

$$\text{(SVM2)} \quad \min_{w,b,q} \ ce^T q + \frac{1}{2} w^T w$$

$$s.t. \quad A_i w + q_i \geq 1 - b \quad \text{for} \quad y_i = 1,$$

$$A_i w - q_i \geq -1 - b \quad \text{for} \quad y_i = -1$$

$$q_i \geq 0, \quad i = 1,2,\ldots,m. \tag{6}$$

Here, c denotes a scalar whose value determines the trade-off; a larger value of c emphasizes

the classification error, while a smaller one places more importance on the classification margin. In practice, rather than solving (SVM1) and (SVM2), we solve their dual problems to get the appropriate hard or soft margin classifier. The case of nonlinear kernels is handled on lines similar to linear kernels [17].

3 TWIN SUPPORT VECTOR MACHINES

In this section, we give a brief outline of TSVM [11]. Here, data points belonging to classes 1 and -1 are represented by matrices A and B, respectively. Let the number of patterns in classes 1 and -1 be given by m_1 and m_2, respectively. Therefore, the sizes of matrices A and B are $(m_1 \times n)$ and $(m_2 \times n)$, respectively. The TWSVM classifier aims to determine two nonparallel planes

$$x^T w^{(1)} + b^{(1)} = 0, \text{ and } x^T w^{(2)} + b^{(2)} = 0 \quad , \tag{7}$$

so as to minimize the Euclidean distance of the planes from the data points of classes 1 and -1, respectively. This leads to the following optimization problems:

$$\text{(TSVM1)} \qquad \underset{w^{(1)},b^{(1)},q}{Min} \quad \frac{1}{2}(Aw^{(1)} + e_1 b^{(1)})^T (Aw^{(1)} + e_1 b^{(1)}) + c_1 e_2^T q$$

$$s.t. \qquad -(Bw^{(1)} + e_2 b^{(1)}) + q \geq e_2, \quad q \geq 0, \tag{8}$$

and

$$\text{(TSVM2)} \qquad \underset{w^{(2)},b^{(2)},q}{Min} \quad \frac{1}{2}(Bw^{(2)} + e_2 b^{(2)})^T (Bw^{(2)} + e_2 b^{(2)}) + c_2 e_1^T q$$

$$s.t. \qquad -(Aw^{(2)} + e_1 b^{(2)}) + q \geq e_1, \quad q \geq 0, \tag{9}$$

where $c_1, c_2 > 0$ are parameters and e_1 and e_2 are vectors of ones of appropriate dimensions.

The Wolfe dual of TWSVM1 as follows:

$$\text{(DTSVM1)} \qquad \underset{\alpha}{Max} \quad e_2^T \alpha - \frac{1}{2}\alpha^T G(H^T H)^{-1} G^T \alpha$$

$$s.t. \quad 0 \leq \alpha \leq c_1. \tag{10}$$

Similarly, consider TWSVM2 and obtain its dual as

$$\text{(DTSVM2)} \qquad \underset{\gamma}{Max} \quad e_1^T \gamma - \frac{1}{2}\gamma^T P(Q^T Q)^{-1} P^T \gamma$$

$$s.t. \quad 0 \leq \gamma \leq c_2 \tag{11}$$

Here, $P = [A \;\; e_1]$, $Q = [B \;\; e_2]$, and the augmented vector $u = [w^{(1)}, b^{(1)}]^T$, which is given by $u = (H^T H)^{-1} G^T \alpha$, $v = [w^{(2)}, b^{(2)}]^T$ is given by $v = (Q^T Q)^{-1} P^T \gamma$.

Once vectors u and v are known, the separating planes (7) are obtained. A new data sample $x \in R^n$ is assigned to class h, depending on which of the two planes given by (7) it lies closest to, i.e.,

$$x^T w^{(h)} + b^{(h)} = \underset{l=1,2}{Min} \left| x^T w^{(l)} + b^{(l)} \right| \tag{12}$$

where $|\cdot|$ is the perpendicular distance of point x from the plane $x^T w^{(l)} + b^{(l)} = 0$, $l = 1,2$.

4 SPARSE TWIN SUPPORT VECTOR MACHINES

4.1 sparsify training samples

Let $S = \{(\bar{x}_1, y_1),...,(\bar{x}_m, y_m)\}$ be a set of m training samples. We assume that each sample $\bar{x}_i$ is drawn from a domain $X \subseteq \Re^n$ and that each label y_i is an integer from the set $Y = \{1,2,..,k\}$.

First we get some samples to represent S through the following algorithm:

1)All samples must be normalized such that the features locate in [0,1] before learning.

2)Each sample is given a weight w_p. at the beginning, $\forall\ p$ (that is to say for every sample) $w_p = 1$.

3) $R = 0.8$.

4) $\forall\ \bar{x}_p \in S$, set $s(p) = \{\bar{x}_q \mid \left\|\bar{x}_q - \bar{x}_p\right\|_1 \le R, \bar{x}_q \in S\}$, here we use L_1 norm. if $\forall\ \bar{x}_q \in s(p)$, $y_q = y_p$, let $R_p = R$ and go to step 5.

5) $\forall\ \bar{x}_q \in S$,if $\left\|\bar{x}_q - \bar{x}_p\right\|_1 \le R \times v, 0 \le v \le 1$, then, $\bar{x}_p = \dfrac{\bar{x}_p w_p + \bar{x}_q w_q}{w_p + w_q}$, $w_p = w_p + w_q$, Delete $\bar{x}_q$.

(through the experiment we known that usually when $v = 0.75$ the classifier performance better.)

6) $R = R - 0.01$.

7) Repeat step 4~6 until $R = 0.01$.

Now the new samples together with its weight can describe the training set approximately.

Here is an example, iris dataset from the UCI repository (just consider the second and forth attributes). Figure 1 show initial samples of 3 classes (green stars, blue rounds and red dots), and figure 2 show sparse samples we get through the algorithm above. Where $v = 0.75$.

Figure 3 and figure 4 display the classifiers calculated from initial samples and sparse samples respectively.

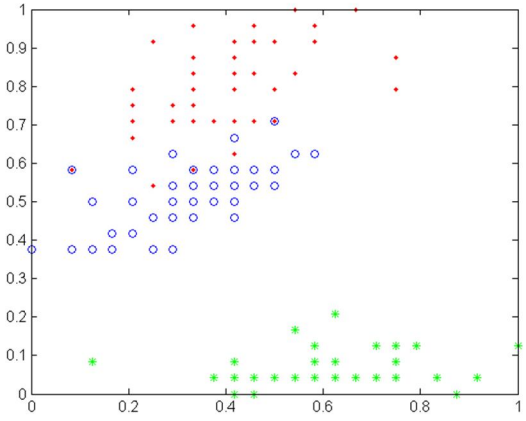

figure 1 initial samples of iris

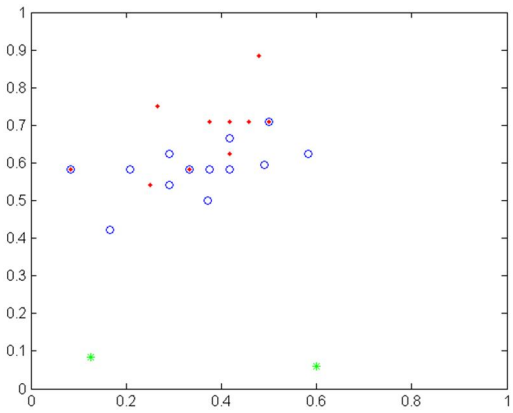

figure 2 sparsified samples of iris

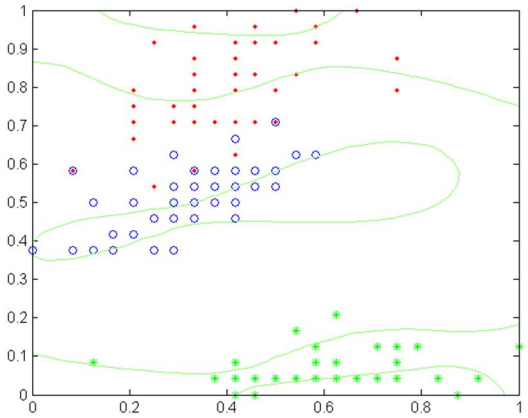

figure 3 classifiers calculated from initial samples

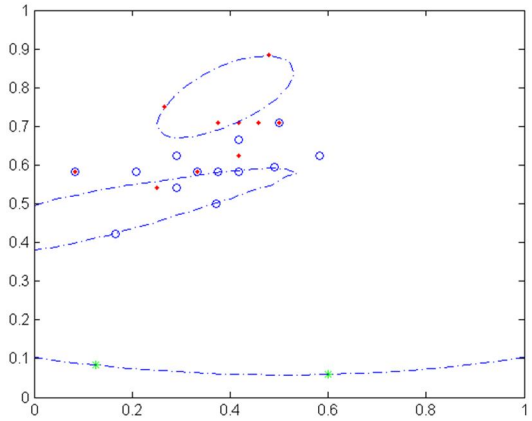

figure 4 classifiers calculated from sparsified samples

4.2 building sparse twin support vector mufti-class cassifier

Here we aimed to get k nonparallel hyperplanes. For each hyperplane, we use a one-versus-rest approach, the k th class and other classes. To typical SVM, this method may yield to data unbalance, but this classifier, a new data sample $\bar{x} \in X$ is assigned to which class depending on which of the hyperplanes we get it lies closest to, so it won't yield to unbalance.

The Sparse twin support vector multi-class classifier is obtained by solving the following k quadratic programming problems

$$\min_{w_r, b_r, \xi_s} \quad \frac{1}{2}(A_r w_r + e_r b_r)^T N_r^3 (A_r w_r + e_r b_r) + c \sum_{s \in Y \setminus \{r\}} e_s^T N_s \xi_s$$

$$s.t. \quad A_s w_r + e_s b_r + \xi_s \geq e_s + N_s \bar{r}_s \ , \ \xi_s \geq 0 .$$

$$r = 1,2,\ldots,k \tag{13}$$

Where $c \geq 0$ are parameters and e_r and e_s are vectors of ones of appropriate dimensions. A_s is a matrix which each row is a data point belonging to class s. Assume that the number of samples in class s is m_s, then

$$A_s = \begin{bmatrix} \bar{x}_{m1} \\ \bar{x}_{m2} \\ \vdots \\ \bar{x}_{m_s} \end{bmatrix}, \quad N_s = \begin{bmatrix} w_{m1} & & & \\ & w_{m2} & & \\ & & \ddots & \\ & & & w_{m_s} \end{bmatrix}. \text{ Vector } \bar{r}_s = [R_{m1}, R_{m2}, \ldots R_{m_s}]' . \tag{14}$$

The Lagrangian corresponding to the problem (13) is given by

$$L(w_r, b_r, \xi_s, \alpha_s, \beta_s) = \frac{1}{2}(A_r w_r + e_r b_r)^T N_r^3 (A_r w_r + e_r b_r) + c \sum_{s \in Y \setminus \{r\}} e_s^T N_s \xi_s - \sum_{s \in Y \setminus \{r\}} \alpha_s^T (A_s w_r + e_s b_r + \xi_s - (e_s + N_s \bar{r}_s)) - \sum_{s \in Y \setminus \{r\}} \beta_s^T \xi_s \tag{15}$$

where α_s and β_s are the vectors of Lagrange multipliers. The Karush-Kuhn-Tucker (K.K.T) necessary and sufficient optimality conditions [18] for (13) are given by

$$A_r^T N_r^3 (A_r w_r + e_r b_r) - \sum_{s \in Y \setminus \{r\}} A_s^T \alpha_s = 0 \tag{16}$$

$$e_r^T N_r^3 (A_r w_r + e_r b_r) - \sum_{s \in Y \setminus \{r\}} e_s^T \alpha_s = 0 \tag{17}$$

$$cN_s e_s - \alpha_s - \beta_s = 0 \tag{18}$$

$$A_s w_r + e_s b_r + \xi_s \geq e_s + N_s \bar{r}_s \quad , \quad \xi_s \geq 0 \tag{19}$$

$$\sum_{s \in Y \setminus \{r\}} \alpha_s^T (A_s w_r + e_s b_r + \xi_s - (e_s + N_s \bar{r}_s)) = 0 \, , \quad \beta_s^T \xi_s = 0 \tag{20}$$

$$\alpha_s \geq 0 \, , \quad \beta_s \geq 0 \tag{21}$$

Where $s \in Y \setminus \{r\}$.
Since $\beta_s \geq 0$, from (18) we have

$$0 \leq \alpha_s \leq cN_s e_s \tag{22}$$

Next, combining (16) and (17) leads to

$$([A_r \; e_r]^T N_r^3 [A_r \; e_r])[w_r^T \; b_r]^T - \sum_{s \in Y \setminus \{r\}} [A_s \; e_s]^T \alpha_s = 0 \tag{23}$$

We define

$$H_r = [A_r \; e_r] \tag{24}$$

and the augmented vector $u_r = [w_r^T \; b_r]^T$. With these notations (23) may be rewritten as

$$(H_r^T N_r^3 H_r) u_r - \sum_{s \in Y \setminus \{r\}} H_s^T \alpha_s = 0 \, , \quad \text{i.e. } u_r = (H_r^T N_r^3 H_r)^{-1} \sum_{s \in Y \setminus \{r\}} H_s^T \alpha_s \tag{25}$$

Although $H_r^T N_r^3 H_r$ is always positive semidefinite, it is possible that it may not be well conditioned in some situations. We use a regularization term εI, $\varepsilon > 0$ [11], to take care of problems due to possible ill-conditioning of $H_r^T N_r^3 H_r$. Here, I is an identity matrix of appropriate dimensions. Therefore, (25) gets modified to

$$u_r = (H_r^T N_r^3 H_r + \varepsilon I)^{-1} \sum_{s \in Y \setminus \{r\}} H_s^T \alpha_s \tag{26}$$

However, in the following, we shall continue to use (25) with the understanding that, if need be, (26) is to be used for the determination of u_r.

Using (15) and the above K.K.T conditions, we obtain the Wolfe dual [18] of (13) as follows:

$$\max_{w_r, b_r, \xi_s, \alpha_s, \beta_s} \quad L(w_r, b_r, \xi_s, \alpha_s, \beta_s)$$

$$s.t. \quad \alpha_s, \beta_s \geq 0$$

$$\frac{\partial L}{\partial w_r} = 0 \, , \quad \frac{\partial L}{\partial b_r} = 0 \, , \quad \frac{\partial L}{\partial \xi_s} = 0 \tag{27}$$

By elimination of w_r, b_r, ξ_s and β_s, the dual problem of (13) is reduced to the following succinct form:

$$\max_{\alpha_s} \quad \sum_{s \in Y \setminus \{r\}} \alpha_s^T (e_s + N_s \bar{r}_s) - \frac{1}{2} \Big(\sum_{s \in Y \setminus \{r\}} \alpha_s^T H_s \Big) (H_r^T N_r^3 H_r)^{-1} \sum_{s \in Y \setminus \{r\}} H_s^T \alpha_s$$

$$s.t. \qquad 0 \le \alpha_s \le cN_s e_s . \tag{28}$$

In the above discussion, the matrix H_r is matrix of size $m_r \times (n+1)$, and matrix $H_r^T N_r^3 H_r$ is of size $(n+1)\times(n+1)$,where, in general, n is much smaller in comparison to the number of patterns of each class.

Once vectors u_r are known from (25), the separating planes

$$\bar{x}^T w_r + b_r = 0, \quad r \in Y \tag{29}$$

are obtained. A new data sample $\bar{x} \in X$ is assigned to which class depending on which of the planes given by (29) it lies closest to, i.e.

$$\bar{x}^T w_h + b_h = \min_{l \in Y} \left| \bar{x}^T w_l + b_l \right| \tag{30}$$

where $|\cdot|$ is the perpendicular distance of point $\bar{x}$ from the plane $\bar{x}^T w_l + b_l = 0$, $l \in Y$.

From the Karush-Kuhn-Tucker conditions (16), (17), (18), (19), (20), (21), and (22), we observe that Patterns of other classes (except class r) for which $0 \le \alpha_s(i) \le cN_s(i)$ $(i = 1,2,\ldots,m_s)$ lie on the hyperplane given by $\bar{x}^T w_r + b_r = e_s + N_s \bar{r}_s$. Taking motivation from standard Twin SVM [11], we can define that such patterns of other classes as support vectors of class r with respect to other classes as they play an important role in determining the required plane.

4.3 The Nonliear Kernel Classifier

Because our classifier is one-versus-rest approach, in many cases we have to face linearly nonseparable problems. So it's very necessary to extend our results to nonlinear classifiers. we consider the following kernel-generated surfaces instead of planes,

$$K(x^T, C^T)v_r + b_r = 0 \tag{31}$$

$$\text{Where } C^T = [A_1^T \; A_2^T \; \ldots \; A_k^T], \tag{32}$$

and K is an appropriately chosen kernel. Note that the planes (29) can be obtained as a special case of (31), by using a linear kernel $K(x^T, C^T) = x^T C$, and by defining $w_r = C^T v_r$. In line with the arguments in 4.2, we construct an optimization problem as follows:

$$\min_{w_r, b_r, \xi_s} \; \frac{1}{2}(K(A_r, C^T)v_r + e_r b_r)^T N_r^3 (K(A_r, C^T)v_r + e_r b_r) + c\sum_{s \in Y\setminus\{r\}} e_s^T N_s \xi_s$$

$$s.t. \;\; K(A_s, C^T)v_r + e_s b_r + \xi_s \ge e_s + N_s \bar{r}_s \;, \;\; \xi_s \ge 0 \tag{33}$$

Where $c \ge 0$ are parameters. The corresponding Lagrangian is

$$\begin{aligned} L(w_r, b_r, \xi_s, \alpha_s, \beta_s) &= (K(A_r, C^T)v_r + e_r b_r)^T N_r^3 (K(A_r, C^T)v_r + e_r b_r) \\ &+ c\sum_{s \in Y\setminus\{r\}} e_s^T N_s \xi_s - \sum_{s \in Y\setminus\{r\}} \alpha_s^T (K(A_s, C^T)v_r + e_s b_r + \xi_s - (e_s + N_s \bar{r}_s)) - \sum_{s \in Y\setminus\{r\}} \beta_s^T \xi_s \end{aligned} \tag{34}$$

We obtain the K.K.T. conditions for (33) as

$$K(A_r, C^T)^T N_r^3 (K(A_r, C^T)v_r + e_r b_r) - \sum_{s \in Y\setminus\{r\}} K(A_s, C^T)\alpha_s = 0, \tag{35}$$

$$e_r^T N_r^3 (K(A_r, C^T)v_r + e_r b_r) - \sum_{s \in Y\setminus\{r\}} e_s^T \alpha_s = 0, \tag{36}$$

$$cN_s e_s - \alpha_s - \beta_s = 0, \tag{37}$$

$$K(A_s, C^T)v_r + e_s b_r + \xi_s \geq e_s \ , \ \xi_s \geq 0, \tag{38}$$

$$\alpha_s^T (K(A_s, C^T)v_r + e_s b_r + \xi_s - (e_s + N_s \bar{r}_s)) = 0, \ \beta_s^T \xi_s = 0, \tag{39}$$

$$\alpha_s \geq 0, \ \beta_s \geq 0, \tag{40}$$

Combining (35) and (36), we obtain

$$([K(A_r, C^T)\ e_r]^T N_r^{\ 3} [K(A_r, C^T)\ e_r])[v_r^T\ b_r]^T - \sum_{s \in Y \setminus \{r\}} [K(A_s, C^T)\ e_s]^T \alpha_s = 0 \tag{41}$$

We define

$$G_r = [K(A_r, C^T)\ \ e_r], \tag{42}$$

and the augmented vector $z_r = [v_r^T\ b_r]^T$. With these notations (41) may be rewritten as

$$(G_r^{\ T} N_r^{\ 3} G_r) z_r - \sum_{s \in Y \setminus \{r\}} G_s^{\ T} \alpha_s = 0, \quad \text{i.e.} \ \ z_r = (G_r^{\ T} N_r^{\ 3} G_r)^{-1} \sum_{s \in Y \setminus \{r\}} G_s^{\ T} \alpha_s \tag{43}$$

The Wolfe dual of (33) is given by

$$\begin{aligned} \max_{\alpha_s} \quad & \sum_{s \in Y \setminus \{r\}} \alpha_s^T (e_s + N_s \bar{r}_s) - \frac{1}{2} \Big(\sum_{s \in Y \setminus \{r\}} \alpha_s^T G_s \Big) (G_r^{\ T} N_r^{\ 3} G_r)^{-1} \sum_{s \in Y \setminus \{r\}} G_s^T \alpha_s \\ s.t. \quad & 0 \leq \alpha_s \leq cN_s e_s . \end{aligned} \tag{44}$$

Once (33) are solved to obtain the surfaces (31), a new pattern $\bar{x} \in X$ is assigned to which class in a manner similar to the linear case.

5 EXPERIMENTAL RESULTS

In this section, we validate the accuracy and efficiency of the proposed STSVM algorithm on several publicly available benchmark datasets including Iris, Wine, Indian Liver Patient, SPECTF Heart, Segmentation, and Wine Quality from the UCI repository. For all the datasets,we linearly scale each attribute to be in the range [0,1] before learning. Table 1 gives a detailed description about these datasets, which "training" indicates training sample size, and "test" indicates test sample size. Here we use RBF-kernel.

Table 1 Dataset description

Data	Training	Test	Class	Attribute
iris	150	0	3	4
wine	130	0	2	13
indian liver patient	150	0	2	10
wine quality	200	0	4	11
SPECTF heart	80	187	2	44
segmentation	210	2100	7	19

5.1. Reliability of sparse algorithm

In the beginning, we display the reliability of STSVM on iris from the UCI repository. Fig.3 shows the initial samples (stars, rounds and dots) and classifiers (solid lines) calculated from them, and fig.4 the sparse samples together with classifiers (dashed) calculated from them (In order to make the observation easier, here we consider the second and forth attributes of each sample only).

As we can see from figure 3 and 4, our algorithm is not only low computation, but also capture main features of training sample better. Here $v = 0.75, c_1 = 0.5,\ c_2 = 1$. To make the observation easier, we plot initial samples and classifiers calculated from sparse samples in figure 5.

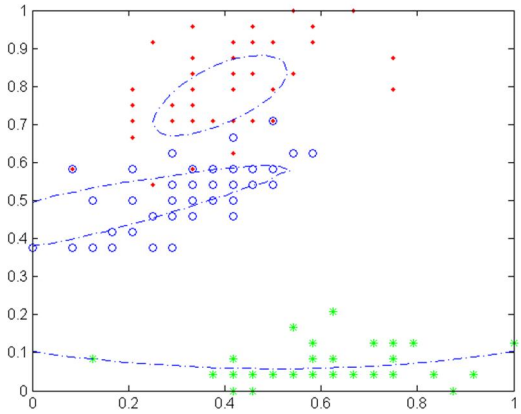

figure 5 initial samples and classifiers calculated from sparsified samples

Next we display the STSVM's performance on iris in table 2. Which training error is the classifying error when use the initial training data as testing data. We first consider the second and forth attributes only, then all attributes are taken into consideration. Mostly, support vector number became less after Sparsification. It is noteworthy that，even if sometimes the SV number increase, training time is shorter, because sparse sample is much smaller.

Table 2 iris

	Attribute 2 and 4			All Attributes			
	STSVM (v=0)	STSVM (v=0.5)	STSVM (v=0.75)	STSVM (v=0)	STSVM (v=0.5)	STSVM (v=0.75)	STSVM (v=1)
Training (initial)	150	150	150	150	150	150	150
Training (sparse)	——	25	21	——	51	24	12
SV of class 1	1	1	1	1	5	1	1
SV of class 2	17	2	2	7	6	4	3
SV of class 3	19	4	2	3	2	5	4
Sum of SV	37	7	5	11	13	10	8
Training Error(%)	0.0467	0.0533	0.0533	0.0133	0.0200	0.0267	0.1000

5.2 Accuracy comparing

In this section, we compared training error and testing error, which is the error rate when classify initial training sample and testing sample respectively. Testing error indicates the accuracy when classify a new point.

Firstly, training accuracy of TSVM and STSVM is compared through Wine and Indian Liver Patient Dataset from the UCI repository in table 3. For the convenience of comparing, we delete the third class of Wine, and consider class 1 and 2 only. The Indian Liver Patient Dataset contains 416 liver patient records and 167 non liver patient records. After deleting the samples which miss values, we have 579 samples in all. Here we use the first 150 samples.

We can see from table 3 that, training error correlates negatively with support vector number usually. But testing error is different. Table 4 shows testing error correlates positively with support vector number. Less SV indicates that it capture classes' main features better to some extent. Actually, STSVM have higher classification accuracy with appropriate parameter v. However, SV number did not always decrease when v increase. Experiments show that mostly STSVM perform better with $v = 0.75$, but how to find the best v still needing future solutions.

5.3 Performance of STSVM on unbalanced data sets

Lastly we show STSVM's performance when training data is unbalanced. Wine quality Dataset from the UCI repository is used, and to reduce calculation we select the first 200 samples.

Table 5 shows that although the training data is very unbalanced, our algorithm did not be overwhelmed by the large-scale classes or ignore the small ones. Where training error of one class equal to the misclassification rate of that class. And "Total" means the misclassification rate of the whole initial sample.

Table 3 training error comparing

	Wine			Indian Liver Patient			
	TSVM	STSVM (v=0.5)	STSVM (v=0.75)	TSVM	STSVM (v=0.5)	STSVM (v=0.75)	STSVM (v=1)
Training (initial)	130	130	130	150	150	150	150
Training (sparse)	——	92	21	——	122	89	47
SV of class1	17	60	3	35	30	28	12
SV of class2	10	4	1	56	84	33	12
Sum of SV	27	64	4	91	114	61	24
Training Error (%)	0	0	0.0077	0.0467	0.0333	0.0733	0.1933

Table 4 testing error comparing

	SPECTF Heart			Segmentation		
	TSVM	STSVM (v=0.75)	STSVM (v=1)	STSVM (v=0.5)	STSVM (v=0.75)	STSVM (v=1)
Training (initial)	80	80	80	210	210	210
Training (sparse)	——	73	23	119	81	51
SV of class 0	40	40	9	——	——	——
SV of class 1	40	33	2	10	10	5
SV of class 2	——	——	——	4	2	2
SV of class 3	——	——	——	16	12	9

SV of class 4	——	——	——	13	11	7
SV of class 5	——	——	——	20	12	11
SV of class 6	——	——	——	11	11	43
SV of class 7	——	——	——	6	2	2
Sum of SV	80	73	11	80	60	79
Training error (%)	0	0	0.2750	0	0.0190	0.1000
Test Error (%)	0.4171	0.4118	0.3850	0.2224	0.1881	0.2429

Table 5 red wine quality

	Initial Sample Size	Training Error %		
		STSVM (v=0.5)	STSVM (v=0.75)	STSVM (v=1)
class 4	12	0	0	0
class 5	131	0.0305	0.0916	0.1450
class 6	50	0.0400	0.0200	0.1600
class 7	7	0	0	0.1429
Total	200	0.0300	0.0650	0.1400

6 CONCLUDING REMARKS

In this paper, we introduce a Sparse-TSVM classifier that solves a multi-class classification problem. We proposed a algorithm which sparsify the given training data points, So the new samples together with its weight can describe the initial training set approximately. By promoting TWSVM idea to multi-class problem, and the sparse algorithm, we got STSVM. The experimental evaluations on several real world datasets show that the proposed STSVM approach can greatly speed-up the training process and achieve a higher classification accuracy. Deficiency is how to find the best v still needing future solutions. We just know when $v = 0.75$ it performance better from experiments.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.70871056 and 71271103), the Six Talents Peak Foundation of Jiangsu Province, and A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions(PAPD).

REFERENCES

[1] C. Cortes, V. Vapnik, Support Vector Networks, Mach Learn 20 (1995) 273-297.

[2] V. Cherkassky, F. Mulier, Learning from Data—Concepts, Theory, and Methods., John Wiley and Sons, New Jersey, 2007.

[3] P.S. Bradley, O.L. Mangasarian, Massive Data Discrimination via Linear Support Vector Machines, Optim Methods Softw 13 (2000) 1-10.

[4] C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Min Knowl Discov 2 (1998) 1-43.

[5] S. Mori, C. Suen, K. Yamamoto, Historical Review of OCR Research and Development, Proc IEEE Inst Electr Electron Eng 80 (1995) 244-273.

[6] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and

hierarchical clustering, VLDB J 16 (2007) 507-521.

[7] A. Ganapathiraju, J. E. Hamaker, J. Picone, Applications of support vector machines to speech recognition, IEEE Trans Signal Process 52 (2004) 2348-2355.

[8] P. Baldi, G. Pollastri, A machine-learning strategy for protein analysis, IEEE Intell Syst 17 (2002) 28-35.

[9] L. Liu, X. Wu, L. Cui, A Dynamic Pattern Classifier for Complex Information Systems Based on Fuzzy Petri Nets, Second International Symposium on Intelligent Information Technology Application 2 (2008) 583-587.

[10] Q. Wu, R. Law, Complex system fault diagnosis based on a fuzzy robust wavelet support vector classifier and an adaptive Gaussian particle swarm optimization, Information Sciences 180 (2010) 4514-4528.

[11] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans Pattern Anal Mach Intell 29 (2007) 905-910.

[12] L. Zhang, W. Zhou, Density-induced margin support vector machines, Pattern Recognit 44 (2011) 1448-1460.

[13] X. He, Z. Wang, C. Jin, Y. Zheng, X. Xue, A simplified multi-class support vector machine with reduced dual optimization, Pattern Recognit Lett 33 (2012) 71-82.

[14] X. Xia, K. Li, A Sparse Multi-class Least-Squares Support Vector Machine, IEEE International Symposium on Industrial Electronics (2008) 1230-1235.

[15] K. Crammer, Y. Singer, On the Learnability and Design of Output Codes for Multiclass Problems, Mach Learn 47 (2002) 201-233.

[16] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J Mach Learn Res 2 (2001) 265-292.

[17] S.R. Gunn, Support Vector Machines for Classification and Regression, technical report, School of Electronics and Computer Science, Univ. of Southampton, Southampton, U.K., 1998, http://www.isis.ecs.soton.ac.uk/resources/svminfo/.

[18] O.L. Mangasarian, Nonlinear Programming, SIAM ed., Society for Industrial and Applied Mathematics, philadelphia, 1994.

[19] W. Shuguo, Y. Hongxing, Pinning synchronization of supply chain complex networks with time- varying topological structures and multiple coupling delays, Journal of Jiangsu University (natural science edition) 33 (2012) 239-243.