Int. J. Mach. Learn. & Cyber. (2017) 8:295–308
DOI 10.1007/s13042-014-0323-4

ORIGINAL ARTICLE

A structural information-based twin-hypersphere support vector machine classifier

Xinjun Peng · Lingyan Kong · Dongjing Chen

Received: 28 July 2014 / Accepted: 21 December 2014 / Published online: 11 January 2015

Abstract Twin-hypersphere support vector machine (THSVM) for binary pattern recognition aims at generating two hyperspheres in the feature space such that each hypersphere contains as many as possible samples in one class and is as far as possible from the other one. THSVM has a fast learning speed since it solves two small sized support vector machine (SVM)-type quadratic programming problems (QPPs). However, it only simply considers the prior class-based structural information in the optimization problems. In this paper, a structural information-based THSVM (STHSVM) classifier for binary classification is presented. This proposed STHSVM focuses on the cluster-based structural information of the corresponding class in each optimization problem, which is vital for designing a good classifier in different real-world problems. In addition, it also leads to a fast learning speed since this STHSVM solves a series of smaller-sized QPPs compared with THSVM. Experimental results demonstrate that STHSVM is superior in generalization performance to other classifiers.

Keywords Binary classification · Quadratic programming problem · Twin-hypersphere support vector machine · Structural information

Electronic supplementary material The online version of this article (doi:10.1007/s13042-014-0323-4) contains supplementary material, which is available to authorized users.

X. Peng (✉) · L. Kong · D. Chen
Department of Mathematics, Shanghai Normal University, Shanghai 200234, Peoples Republic of China
e-mail: xjpeng@shnu.edu.cn

X. Peng
Scientific Computing Key Laboratory of Shanghai Universities, Shanghai 200234, Peoples Republic of China

1 Introduction

Support vector machine (SVM) [1–3] finds the maximal margin between two classes [4] by solving a quadratic programming problem (QPP) in the dual space based on the structural risk minimization principle. Within a few years after its introduction SVM not only has a serious of improvements [5, 6], but also has already outperformed most other systems in a wide variety of applications [7–9]. However, classical SVM not only has the large computational cost, but also usually pays more attention to the separation between classes than the prior structural information within classes. In fact, for different real-world problems, different classes may have different underlying data structures.

Recently, a class of nonparallel hyperplane classifiers have been developed. For instance, TWSVM [10] aims at generating a pair of nonparallel planes such that each plane is as close as possible to the corresponding class and is at least one far from the other class. To this end, it solves a pair of smaller-sized QPPs, instead of a large one in SVM, making the learning speed of TWSVM be approximately four times faster than that of SVM in theory [10]. Some extensions to TWSVM include the least squares TWSVM (LSTWSVM) [11], nonparallel-plane proximal classifier (NPPC) [12], v-TWSVM [13], twin parametric-margin SVM (TPMSVM) [14], projection twin support vector machine (PTSVM) [15], nonparallel hyperplane SVM (NHSVM) [16], twin support vector regression (TSVR) [17], and twin parametric insensitive support vector regression (TPISVR) [18].

Different from TWSVM which seeks a hyperplane for each class using a SVM-type formulation, Peng and Xu [19] proposed a twin-hypersphere support vector machine (THSVM) classifier for binary classification, which aims at

generating two hyperspheres in the feature space such that each hypersphere contains as many as possible samples in one class and is as far as possible from the other one. The THSVM not only has a faster learning speed than classical SVM since it solves two smaller sized QPPs instead of a large QPP as in classical SVM, but also successfully avoids the shortcomings in TWSVM [19], such as the matrix inversion problem. In this paper, we mainly focus on this THSVM.

As the relation between the structural information of data and SVM, it is desirable that an SVM classifier be adaptable to the discriminant boundaries to fit the structures in the data, especially for increasing the generalization capacities of the classifier. Fortunately, some algorithms have been developed to focus more attention on the structural information than SVM recently. They provide a novel view in which to design a classifier, that is, a classifier should be sensitive to the structure of data distribution [20]. These algorithms can be mainly divided into two kinds of approaches. The first one is manifold assumption-based, which assumes that the data actually lie on a sub-manifold in the input space. A typical model is Laplacian SVM (LapSVM) [21, 22]. LapSVM constructs a Laplacian graph for each class on top of the local neighborhood of each point to form the corresponding Laplacian (matrix) to reflect the manifold structure of individual-class data. They are then embedded into the traditional framework of SVM as additional manifold regularization terms. The second approach is cluster assumption-based [23], which assumes that the data contains clusters. For instance, structured large margin machine (SLMM) [20], ellipsoidal kernel machine (EKM) [24], minimax probability machine (MPM) [25], and maxi-min margin machine (M^4) [26]. However, the computational cost of these approaches is larger than classical SVM. More recently, Xue et al. [27] proposed a structural regularized SVM (SRSVM). This SRSVM embeds a cluster granularity into the regularization term to capture the data structure, Peng et al. [28] proposed a structural regularized PTSVM (SRPTSVM) for data classification in the spirit of this SRSVM.

THSVM only considers the relationship between two classes, i.e., it finds two hyperspheres to respectively cover the classes of points. In other words, it embeds the class granularity-based structural information [27] into the optimization problems, but not the covariance matrices of two classes. However, this structural information is too rough for real-world problems, which makes THSVM can not find the reasonable projection for each class, then reduce the generalization performance. To overcome this shortcoming, we present an improvement version for THSVM in this paper, called the structural-information-based THSVM (STHSVM) classifier. This STHSVM respectively embeds the data structures of two classes into the optimization problems based on the cluster granularity [27]. That is, in the pair of optimization problems of STHSVM, it considers the cluster-based structural-information constraints for each class, i.e., it introduces a series of hyperspheres but not a single hypersphere to respectively cover the corresponding class of points. Further, for each point in the opposite class, this STHSVM wishes it be as far as possible from the centers of all hypersphres under the given probability values. This STHSVM only needs to solve a series of much smaller sized QPPs compared with THSVM, indicating it has a much faster learning speed than THSVM for solving their QPPs. The experiment results show that this STHSVM obtains the better generalization than THSVM and the other classifiers.

The rest of this paper is organized as follows: Sect. 2 briefly introduces the structural granularities of data and THSVM. Section 3 presents the proposed STHSVM. Experimental results both on the toy and real-world problems are given in Sect. 4. Some conclusions and possible further work are drawn in Sect. 5.

2 Background

In this paper, the training samples are denoted by a set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{l}$, where $x_i \in \mathcal{X} \subset \mathcal{R}^m$ and $y_i \in \{+1, -1\}$, $i = 1, \ldots, l$. For simplicity, we use $\mathcal{I}_\pm$ to denote the sets of index i such as $y_i = \pm$, $k = 1, 2$, use the set $\mathcal{I}$ to denote all point indices, i.e., $\mathcal{I} = \mathcal{I}_+ \cup \mathcal{I}_-$, and use the matrices $C \in \mathcal{R}^{m\times l}$, $C_+ \in \mathcal{R}^{m\times l_+}$ and $C_- \in \mathcal{R}^{m\times l_-}$ to represent all training points, and points belonging to classes ± 1, respectively, where $l_\pm = |\mathcal{I}_\pm|$.

2.1 Structural granularity

Let $\mathcal{S}_1, \ldots, \mathcal{S}_t$ be a partition of $\mathcal{D}$ according to some relation measure, where the partition characterizes the whole data in the form of some structures such as cluster, and $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_t = \mathcal{D}$. Here $\mathcal{S}_i$, $i = 1, \ldots, t$ is called *structural granularity* [27]. In general, four granularity layers can be differentiated:

Global granularity The granularity refers to the dataset $\mathcal{D}$. With this granularity, the whole data are characterized or enclosed by a single ellipsoid with center μ and covariance matrix Σ obtained by minimizing its volume [24]:

$$\begin{aligned} &\min_{\mu,\Sigma} \ \ln|\Sigma| \\ &\text{s.t. } \left|\left|(x_i-\mu)^{-1}\Sigma^{-1}(x_i-\mu)\right|\right| \le 1, \ \forall i, \\ &\quad \Sigma \ge 0. \end{aligned} \tag{1}$$

The corresponding classifier, such as EKM [24], aims to utilize such global data structure, or more precisely, global data scatter in its design.

Class granularity The granularities are the class partitioned data subsets. Single ellipsoid can be used to describe an individual class to form the so called class structure. The covariance matrices of two classes are defined as

$$\Sigma_{\pm} = \frac{1}{l_{\pm}} \sum_{i \in \mathcal{I}_{\pm}} [x_i - \mu_{\pm}][x_i - \mu_{\pm}]^T = C_{\pm} J_{\pm} J_{\pm}^T C_{\pm}^T, \tag{2}$$

where $J_{\pm} = \frac{1}{\sqrt{l_{\pm}}} \left(I - \frac{1}{l_{\pm}} ee^T \right)$, $\mu_{\pm} = \frac{1}{l_{\pm}} \sum_{i \in \mathcal{I}_{\pm}} x_i$ are the means of two classes, e is the vector of ones with appropriate dimensions, and I is the identity matrix with appropriate dimension. For example, PTSVM [15], respectively embeds the class granularities into the two optimization problems.

Cluster granularity The granularities are the data subsets within each class. The data structures within each class are depicted by a certain amount ellipsoids that are obtained by some clustering techniques. The corresponding covariance matrix in cluster i is: $\Sigma_{\mathcal{C}_i} = C_{\mathcal{C}_i} J_{\mathcal{C}_i} J_{\mathcal{C}_i}^T C_{\mathcal{C}_i}^T$, where $\mathcal{C}_i$ is the index set of cluster i and $J_{\mathcal{C}_i} = \frac{1}{\sqrt{|\mathcal{C}_i|}} \left(I - \frac{1}{|\mathcal{C}_i|} ee^T \right)$. For example, SLMM [20] considers the cluster assumption about the data.

Point granularity The granularities are the neighborhoods $ne(x_i)$ of each point x_i, which are described by overlapped local ellipsoids surrounding the data in each class, whose covariance matrix can be viewed as a kind of local generalized covariance $\Sigma_i = \sum_{j \in ne(x_i)} s_{ij} (x_i - x_j)(x_i - x_j)^T$, where $s_{ij} = \exp(-\gamma ||x_i - x_j||^2)$, $\gamma > 0$. One of the most successful classifier under this granularity is LapSVM [21], which is successfully applied into semi-supervised problems.

2.2 Twin-hypersphere support vector machine

For binary pattern recognition, the THSVM uses a pair of hyperspheres, one for each class, to describe the samples in two classes, and classifies points according to which hypersphere a given point is relatively closest to. To this end, it obtains two optimization problems, and each one has an SVM-type formulation. Specifically, the THSVM is obtained by solving the following pair of optimization problems:

$$\begin{aligned} \min \ & R_+^2 - \frac{v_+}{l^-} \sum_{j \in \mathcal{I}^-} \left|\left| \varphi(x_j) - c_+ \right|\right|^2 + \frac{\gamma_+}{l^+} \sum_{i \in \mathcal{I}^+} \xi_i \\ \text{s.t.} \ & ||\varphi(x_i) - c_+||^2 \le R_+^2 + \xi_i, \\ & R_+^2 \ge 0, \ \xi_i \ge 0, \ i \in \mathcal{I}_+, \end{aligned} \tag{3}$$

$$\begin{aligned} \min \ & R_-^2 - \frac{v_-}{l^+} \sum_{i \in \mathcal{I}^+} ||\varphi(x_i) - c_+||^2 + \frac{\gamma_-}{l^-} \sum_{j \in \mathcal{I}^-} \xi_j \\ \text{s.t.} \ & \left|\left| \varphi(x_j) - c_- \right|\right|^2 \le R_-^2 + \xi_j, \\ & R_-^2 \ge 0, \ \xi_j \ge 0, \ j \in \mathcal{I}_-, \end{aligned} \tag{4}$$

where $\gamma_{\pm} > 0$ and $v_{\pm} > 0$ are pre-specified penalty factors, and $c_{\pm} \in \mathcal{H}$ and $R_{\pm}$ are the centers and radiuses of the hyperspheres, respectively.

Clearly, the first term of (3) or (4) minimizes the squares radius of the hypersphere to keep the hypersphere as compact as possible. The second term in the objective function of (3) or (4) maximizes the sum of squared distances from the center of hypersphere to the points of the opposite class, which leads to keep the center of this hypersphere far from the samples of the opposite class. The constraints require that the samples of the corresponding class be covered by this hypersphere. Otherwise, a set of error variables is used to measure the errors wherever these points are not covered by this hypersphere. The last term of (3) or (4) minimizes the sum of error variables, thus attempting to minimize misclassification due to points belonging to the opposite class.

Introducing the Lagrangian functions for the problems (3) and (4) and considering the Karush–Kuhn–Tucker (KKT) necessary and sufficient optimality conditions, we obtain their dual problems

$$\begin{aligned} \max \ & \sum_{i \in \mathcal{I}_+} \alpha_i \left[\frac{2v_+}{l_-} \sum_{j \in \mathcal{I}_-} k(x_j, x_i) + (1 - v_+) k(x_i, x_i) \right] \\ & - \sum_{i_1, i_2 \in \mathcal{I}_+} \alpha_{i_1} \alpha_{i_2} k(x_{i_1}, x_{i_2}) \\ \text{s.t.} \ & \sum_{i \in \mathcal{I}_+} \alpha_i = 1, \ 0 \le \alpha_i \le \frac{\gamma_+}{l_+}, \quad i \in \mathcal{I}_+. \end{aligned} \tag{5}$$

$$\begin{aligned} \max \ & \sum_{j \in \mathcal{I}_-} \alpha_j \left[\frac{2v_-}{l_+} \sum_{i \in \mathcal{I}_+} k(x_i, x_j) + (1 - v_-) k(x_j, x_j) \right] \\ & - \sum_{j_1, j_2 \in \mathcal{I}_-} \alpha_{j_1} \alpha_{j_2} k(x_{j_1}, x_{j_2}) \\ \text{s.t.} \ & \sum_{j \in \mathcal{I}_-} \alpha_j = 1, \ 0 \le \beta_j \le \frac{\gamma_-}{l_-}, \quad j \in \mathcal{I}_-, \end{aligned} \tag{6}$$

where α_i's are the nonnegative Lagrangian multipliers, and $k(u, v)$ is a kernel function: $k(u, v) = u^T v$ for the linear case, and $k(u, v) = \varphi(u)^T \varphi(v)$ for the nonlinear case, such as the Gauss kernel $k(u, v) = \exp\{-\sigma ||u - v||^2\}, \sigma > 0$.

After solving (5) and (6), we will obtain the two hyperspheres $||\varphi(x) - c_{\pm}||^2 \le R_{\pm}^2$, where the $c_{\pm}$ and $R_{\pm}^2$ values are computed by the KKT necessary and sufficient optimality conditions, which are:

$$c_{\pm} = \frac{1}{1-v_{\pm}}\left(\sum_{i\in\mathcal{I}_{\pm}}\alpha_i\varphi(x_i) - \frac{v_{\pm}}{l_{\mp}}\sum_{j\in\mathcal{I}_{\mp}}\varphi(x_j)\right), \tag{7}$$

$$R_{\pm}^2 = \frac{1}{|\mathcal{I}'_{\pm}|}\sum_{i\in\mathcal{I}'_{\pm}}||\varphi(x_i) - c_{\pm}||^2, \tag{8}$$

where the index sets $\mathcal{I}'_{\pm} = \left\{i \middle| 0<\alpha_i<\frac{c_{\pm}}{l_{\pm}},\ i\in\mathcal{I}_{\pm}\right\}$.

Then, a new test sample x is assigned to the class $+$ or $-$, depending on which of the two hyperspheres it lies relatively closest to, i.e.:

$$f(x) = \arg\min_{+,-}\left\{\frac{||\varphi(\mathrm{x}) - \mathrm{c}_+||^2}{\mathrm{R}_+^2}, \frac{||\varphi(\mathrm{x}) - \mathrm{c}_-||^2}{\mathrm{R}_-^2}\right\}. \tag{9}$$

3 Structural information-based twin-hypersphere support vector machine

Following the line of the cluster granularity model, the structural information-based twin-hypersphere support vector machine (STHSVM) classifier has two steps: clustering and learning. STHSVM first adopts some clustering techniques to capture the data distribution within classes, then respectively embeds the minimization of the compactness between the estimated clusters into the objective functions. In the following subsections, we will discuss these steps concretely.

3.1 Clustering

In this step, many clustering methods, such as k-means [29], nearest neighbor clustering [30], and fuzzy clustering [31], can be employed. The aim of clustering is to investigate the underlying data distribution within classes in SRPTSVM. After clustering, the structural information is introduced into the optimization problem by the covariance matrices of the clusters. So the clusters should be compact and spherical for the computation. Following this objective, we consider the following Wards linkage clustering (WLC) technique [32]. Here we only show the linear case, while this clustering method can also be applicable in the kernel space.

Concretely, if $\mathcal{C}_1$ and $\mathcal{C}_2$ are two clusters, also are the index sets of the points in the two clusters, then their Wards linkage $W(\mathcal{C}_1,\mathcal{C}_2)$ can be calculated as

$$W(\mathcal{C}_1,\mathcal{C}_2) = \frac{|\mathcal{C}_1|\cdot|\mathcal{C}_2|\cdot||\mu_{\mathcal{C}_1} - \mu_{\mathcal{C}_2}||^2}{|\mathcal{C}_1| + |\mathcal{C}_2|}, \tag{10}$$

where $\mu_{\mathcal{C}_\infty}$ and $\mu_{\mathcal{C}_\in}$ are the means of the two clusters, respectively.

Initially, each sample is a cluster in the clustering algorithm. The Wards linkage of two samples x_i and x_j is defined as $W(x_i,x_j) = ||x_i - x_j||^2/2$. During clustering, the two clusters with the smallest Wards linkage value are merged. When two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ are being merged to a new cluster $\mathcal{C}'$, the linkage $W(\mathcal{C}',\mathcal{C})$ of $\mathcal{C}'$ and other cluster $\mathcal{C}$ can be conveniently derived from $W(\mathcal{C}_1,\mathcal{C})$, $W(\mathcal{C}_2,\mathcal{C})$, and $W(\mathcal{C}_1,\mathcal{C}_2)$ by

$$W(\mathcal{C}',\mathcal{C}) = \frac{(|\mathcal{C}_1|+|\mathcal{C}|)W(\mathcal{C}_1,\mathcal{C}) + (|\mathcal{C}_2|+|\mathcal{C}|)W(\mathcal{C}_2,\mathcal{C}) - |\mathcal{C}|W(\mathcal{C}_1,\mathcal{C}_2)}{|\mathcal{C}_1|+|\mathcal{C}_2|+|\mathcal{C}|}. \tag{11}$$

To simply determine the cluster number, this WLC uses kernels to measure the similarity between clusters. Salvador and Chan [33] provided a method to automatically determine the number of clusters that selects the number corresponding to the knee point, i.e., the point of maximum curvature, on the curve.

3.2 STHSVM classifier

Without loss of generality, we denote the clusters in two classes as $\mathcal{P}_1,\ldots,\mathcal{P}_{c_1}$ and $\mathcal{N}_1,\ldots,\mathcal{N}_{c_2}$, respectively. To find the hyperspheres for each class, which show the compactness within classes, i.e., the clusters that cover the different structural information in different classes, the STHSVM classifier optimizes the following two optimization problems:

$$\begin{aligned}
&\min \sum_{s=1}^{c_1}\frac{l_{+,s}}{l_+}R_{+,s}^2 - \frac{v_+}{l_-}\sum_{j\in\mathcal{I}_-}\sum_{s=1}^{c_1}p_{j,s}\left|\left|\varphi(x_j) - c_{+,s}\right|\right|^2 + \frac{\gamma_+}{l_+}\sum_{i\in\mathcal{I}_+}\xi_i\\
&\text{s.t. } ||\varphi(x_i) - c_{+,s}||^2 \le R_{+,s}^2 + \xi_i, \text{ if i}\in\mathcal{P}_\text{s},\\
&\quad \xi_i \ge 0, R_{+,s}^2 \ge 0, i\in\mathcal{I}_+,\ \ s=1,\ldots,c_1,
\end{aligned} \tag{12}$$

$$\begin{aligned}
&\min \sum_{t=1}^{c_2}\frac{l_{-,t}}{l_-}R_{-,t}^2 - \frac{v_-}{l_+}\sum_{i\in\mathcal{I}_+}\sum_{t=1}^{c_2}p_{i,t}\left|\left|\varphi(x_i) - c_{-,t}\right|\right|^2 + \frac{\gamma_-}{l_-}\sum_{j\in\mathcal{I}_-}\xi_j\\
&\text{s.t. } ||\varphi(x_j) - c_{-,t}||^2 \le R_{-,t}^2 + \xi_j, \text{ if j}\in\mathcal{N}_\text{t},\\
&\quad \xi_j \ge 0, R_{-,t}^2 \ge 0, j\in\mathcal{I}_-,\ \ t=1,\ldots,c_2,
\end{aligned} \tag{13}$$

where $\gamma_{\pm}, v_{\pm}$, $k=1,2$ are penalty factors given by users, $l_{+,s} = |\mathcal{P}_s|$, $s=1,\ldots,c_1$ and $l_{-,t} = |\mathcal{N}_t|$, $t=1,\ldots,c_2$ denote the sizes of the clusters $\mathcal{P}_s$ and $\mathcal{N}_t$, $l_+ = \sum_{s=1}^{c_1} l_{+,s}$, $l_- = \sum_{t=1}^{c_2} l_{-,t}$, $c_{+,s}, R_{+,s}$, $s=1,\ldots,c_1$ and $c_{-,t}, R_{-,t}$, $t=1,\ldots,c_2$, are the centers and radiuses of clusters $\mathcal{P}_s$ and $\mathcal{N}_t$, respectively. In addition, $p_{j,s}$, $s=1,\ldots,c_1$, $j\in\mathcal{I}_-$ are the probabilities of x_j belonging to clusters $\mathcal{P}_s$, and $p_{i,t}$, $t=1,\ldots,c_2$, $i\in\mathcal{I}_+$ are the probabilities of x_i belonging to clusters $\mathcal{N}_t$. In this work, we define them as

$$p_{i,t} = \frac{\kappa\left(||x_i - \mu_{-,t}||\right)}{\sum_{t'=1}^{c_2}\kappa\left(||x_i - \mu_{-,t'}||\right)}, \quad i\in\mathcal{I}_+, \quad t=1,\ldots,c_2, \tag{14}$$

and

$$p_{j,s} = \frac{\kappa\left(||x_j - \mu_{+,s}||\right)}{\sum_{s'=1}^{c_1} \kappa\left(||x_j - \mu_{+,s'}||\right)}, \quad j \in \mathcal{I}_-, \quad s = 1, \ldots, c_1, \tag{15}$$

where $\kappa(u) = \exp(-\tau u^2)$, $\tau > 0$, $\mu_{+,s}$, $s = 1, \ldots, c_1$ and $\mu_{-,t}$, $t = 1, \ldots, c_2$ are the means of clusters $\mathcal{P}_s$ and $\mathcal{N}_t$, respectively. We have $\sum_{t=1}^{c_2} p_{i,t} = 1$ for all $i \in \mathcal{I}_+$ and $\sum_{s=1}^{c_1} p_{j,s} = 1$ for all $j \in \mathcal{I}_-$. Obviously, for any point x_i, $i \in \mathcal{I}_+$, we have a larger value $p_{i,t}$ if x_i, $i \in \mathcal{I}_+$ is nearer to cluster t than the other clusters in Class $-$.

First of all, we consider the illustrations of the optimization problems (12) and (13) before optimizing them. First, the constraints require that the samples in a cluster of the corresponding class be covered by one hypersphere. Otherwise, a set of error variables $\{\xi_i, i \in \mathcal{I}_+\}$ or $\{\xi_j, j \in \mathcal{I}_-\}$ is used to measure the errors wherever these points are not covered by this hypersphere. Compared with the corresponding term of THSVM, this term makes STHSVM pay more attention to the cluster granularity-based structure information, which are more reasonable for real-world problems. Clearly, for many real-world problems, it is not suitable to use one hypersphere to only cover the points in one class. The last term of the objective function of (12) or (13) minimizes the sum of error variables, thus attempting to minimize misclassification due to points belonging to the opposite class. Second, the first term in the objective function of (12) or (13) minimizes the squares radiuses of the hyperspheres. Hence, minimizing them tends to keep these hyperspheres as compact as possible, i.e., makes these hyperspheres be as small as possible. Note that this term gives the different weights for these hyperspheres according to the sizes of clusters. This definition for weights is reasonable since the clusters with larger sizes should have larger influence than those with smaller sizes. Last, the second term in the objective function of (12) or (13) maximizes the sum of squared distances from the centers of hyperspheres to the points of the opposite class, which leads to keep the centers of this hyperspheres far from the samples of the opposite class. However, by considering the cluster granularity-based structure information, this term introduces the different weights $p_{j,s}$ or $p_{i,t}$. In fact, if one point in Class $-$ is nearer to a cluster of Class $+$, we will hope it is as far as possible from the corresponding hyperphere's center of this cluster than the other centers. In this STHSVM, we use (14) or (15) to depict this end, which can describe the above design.

We now consider to optimize the primal optimization problems (12) and (13). The corresponding Lagrangian function of the problem (12) is

$$\begin{aligned}
&\mathcal{L}(c_{+,s}, R_{+,s}^2, \xi_i, \alpha_i, \beta_i, \lambda_s) \\
&= \sum_{s=1}^{c_1} \frac{l_{+,s}}{l_+} R_{+,s}^2 - \frac{v_+}{l_-} \sum_{j \in \mathcal{I}_-} \sum_{s=1}^{c_1} p_{j,s} \left|\left|\varphi(x_j) - c_{+,s}\right|\right|^2 \\
&\quad + \frac{\gamma_+}{l_+} \sum_{i \in \mathcal{I}_+} \xi_i - \sum_{i \in \mathcal{I}_+} \beta_i \xi_i - \sum_{s=1}^{c_1} \lambda_s R_{+,s}^2 \\
&\quad + \sum_{s=1}^{c_1} \sum_{i \in \mathcal{P}_s} \alpha_i \left(||\varphi(x_i) - c_{+,s}||^2 - R_{+,s}^2 - \xi_i \right),
\end{aligned} \tag{16}$$

where $\lambda_s \geq 0$, $s = 1, \ldots, c_1$, $\alpha_i \geq 0$, $\beta_i \geq 0$, $i \in \mathcal{I}_+$ are the Lagrangian multipliers. Differentiating the Lagrangian function () with respect to $c_{+,s}$, $R_{+,s}^2$, and ξ_i, $i \in \mathcal{I}_+$ yields the following Karush–Kuhn–Tucker (KKT) necessary and sufficient optimality conditions:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_{+,s}} &= -\frac{2v_+}{l_-} \sum_{j \in \mathcal{I}_-} p_{j,s} \left(c_{+,s} - \varphi(x_j)\right) \\
&\quad + 2 \sum_{i \in \mathcal{P}_s} \alpha_i \left(c_{+,s} - \varphi(x_i)\right) = 0 \\
\Rightarrow c_{+,s} &= \frac{1}{\sum_{i \in \mathcal{P}_s} \alpha_i - \frac{v_+}{l_-} \sum_{j \in \mathcal{I}_-} p_{j,s}} \\
&\quad \left(\sum_{i \in \mathcal{P}_s} \alpha_i \varphi(x_i) - \frac{v_+}{l_-} \sum_{j \in \mathcal{I}_-} p_{j,s} \varphi(x_j) \right) \\
&\quad s = 1, \ldots, c_1,
\end{aligned} \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial R_{+,s}^2} = \frac{l_{+,s}}{l_+} - \sum_{i \in \mathcal{P}_s} \alpha_i - \lambda_s = 0 \Rightarrow \sum_{i \in \mathcal{P}_s} \alpha_i \leq \frac{l_{+,s}}{l_+}, \quad s = 1, \ldots, c_1, \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = \frac{\gamma_+}{l_+} - \alpha_i - \beta_i = 0 \Rightarrow 0 \leq \alpha_i \leq \frac{\gamma_+}{l_+}, \quad i \in \mathcal{I}_+, \tag{19}$$

$$\left|\left|\varphi(x_i) - c_{+,s}\right|\right|^2 \leq R_{+,s}^2 + \xi_i, \quad i \in \mathcal{P}_s, \tag{20}$$

$$\alpha_i \left(||\varphi(x_i) - c_{+,s}||^2 - R_{+,s}^2 - \xi_i \right) = 0, \quad \alpha_i \geq 0, \quad i \in \mathcal{P}_s, \quad s = 1, \ldots, c_1, \tag{21}$$

$$\beta_i \xi_i = 0, \quad \xi_i \geq 0, \quad \beta_i \geq 0, \quad i \in \mathcal{I}_+, \tag{22}$$

$$\lambda_s R_{+,s}^2 = 0, \quad R_{+,s}^2 \geq 0, \quad \lambda_s \geq 0. \tag{23}$$

Note that $R_{+,s}^2 > 0$ will hold in the optimality result of problem (12) if the suitable parameters v_+ and γ_+ are given. Then, we have $\sum_{i \in \mathcal{P}_s} \alpha_i = \frac{l_{+,s}}{l_+}$, $s = 1, \ldots, c_1$, according to the KKT conditions (18) and (23), and

$$c_{+,s} = \frac{1}{\frac{l_{+,s}}{l_+} - \frac{v_+}{l_-} \sum_{j \in \mathcal{I}_-} p_{j,s}} \left(\sum_{i \in \mathcal{P}_s} \alpha_i \varphi(x_i) - \frac{v_+}{l_-} \sum_{j \in \mathcal{I}_-} p_{j,s} \varphi(x_j) \right), \quad s = 1, \ldots, c_1. \tag{24}$$

Substituting (18), (19) and (24) into (16), we obtain the dual optimization problem of (12) as following:

$$\begin{aligned}\max\ & -\sum_{s=1}^{c_1}\left[\frac{1}{\frac{l_{+,s}}{l_+}-\frac{v_+}{l_-}\sum_{j\in\mathcal{I}_-}p_{j,s}}\sum_{i_1\in\mathcal{P}_s}\sum_{i_2\in\mathcal{P}_s}\alpha_{i_1}\alpha_{i_2}k(x_{i_1},x_{i_2})\right.\\ & -2\frac{1}{\frac{l_{+,s}}{l_+}-\frac{v_+}{l_-}\sum_{j\in\mathcal{I}_-}p_{j,s}}\sum_{i\in\mathcal{P}_s}\alpha_i\sum_{j\in\mathcal{I}_-}\frac{v_+}{l_-}p_{j,s}k(x_i,x_j)\\ & \left.-\sum_{i\in\mathcal{P}_s}\alpha_i k(x_i,x_i)\right]\\ & +\text{constant}\\ \text{s.t.}\ & \sum_{i\in\mathcal{P}_s}\alpha_i=\frac{l_{+,s}}{l_+},\ 0\le\alpha_i\le\frac{\gamma_+}{l_+},\ i\in\mathcal{P}_s,\quad s=1,\ldots,c_1,\end{aligned}\tag{25}$$

where the constant term does not influence the solution of this optimization problem. Hence we can omit this term in the optimization process.

This optimization problem can be broken down into a series of small-sized optimization problems by multiplying the objective functions by $\left(\frac{l_{+,s}}{l_+}-\frac{v_+}{l_-}\sum_{j\in\mathcal{I}_-}p_{j,s}\right)$ for $s=1,\ldots,c_1$:

$$\begin{aligned}\min\ & \sum_{i_1\in\mathcal{P}_s}\sum_{i_2\in\mathcal{P}_s}\alpha_{i_1}\alpha_{i_2}k(x_{i_1},x_{i_2})\\ & -\sum_{i\in\mathcal{P}_s}\alpha_i\left[\sum_{j\in\mathcal{I}_-}\frac{2v_+}{l_-}p_{j,s}k(x_i,x_j)\right.\\ & \left.+\left(\frac{l_{+,s}}{l_+}-\frac{v_+}{l_-}\sum_{j\in\mathcal{I}_-}p_{j,s}\right)k(x_i,x_i)\right]\\ \text{s.t.}\ & \sum_{i\in\mathcal{P}_s}\alpha_i=\frac{l_{+,s}}{l_+},\quad 0\le\alpha_i\le\frac{\gamma_+}{l_+},\\ & i\in\mathcal{P}_s,\quad s=1,\ldots,c_1.\end{aligned}\tag{26}$$

Next, notice that $\left|\left|\varphi(x_i)-c_{+,s}\right|\right|^2=R_{+,s}^2$ if $0<\alpha_i<\frac{\gamma_+}{l_+}, i\in\mathcal{P}_s$ according to the KKT conditions (19)–(23). Thus, we compute the square radiuses $R_{+,s}^2$, $s=1,\ldots,c_1$ by the following formula:

$$R_{+,s}^2=\frac{1}{|\mathcal{I}_{+,s}^R|}\sum_{i\in\mathcal{I}_{+,s}^R}\left|\left|\varphi(x_i)-c_{+,s}\right|\right|^2,\quad s=1,\ldots,c_1,\tag{27}$$

where the index sets $\mathcal{I}_{+,s}^R=\left\{i\middle|0<\alpha_i<\frac{\gamma_+}{l_+},\ i\in\mathcal{P}_s\right\}$, $s=1,\ldots,c_1$.

Similarly, we obtain the c_2 simplified dual of problems (13) as following:

$$\begin{aligned}\min\ & \sum_{j_1\in\mathcal{N}_t}\sum_{j_2\in\mathcal{N}_t}\alpha_{j_1}\alpha_{j_2}k(x_{j_1},x_{j_2})\\ & -\sum_{j\in\mathcal{N}_t}\alpha_j\left[\sum_{i\in\mathcal{I}_+}\frac{2v_-}{l_+}p_{i,t}k(x_j,x_i)\right.\\ & \left.+\left(\frac{l_{-,t}}{l_-}-\frac{v_-}{l_+}\sum_{i\in\mathcal{I}_+}p_{i,t}\right)k(x_j,x_j)\right]\\ \text{s.t.}\ & \sum_{j\in\mathcal{N}_t}\alpha_j=\frac{l_{-,t}}{l_-},\quad 0\le\alpha_j\le\frac{\gamma_-}{l_-},\\ & j\in\mathcal{N}_t,\quad t=1,\ldots,c_2,\end{aligned}\tag{28}$$

where $\alpha_j, j\in\mathcal{I}_-$ are the nonnegative Lagrangian multipliers, and the centers $c_{-,t}$, $t=1,\ldots,c_2$ are

$$c_{-,t}=\frac{1}{\frac{l_{-,t}}{l_-}-\frac{v_-}{l_+}\sum_{i\in\mathcal{I}_+}p_{i,t}}\left(\sum_{j\in\mathcal{N}_t}\alpha_j\varphi(x_j)-\frac{v_-}{l_+}\sum_{i\in\mathcal{I}_+}p_{i,t}\varphi(x_i)\right),\quad t=1,\ldots,c_2.\tag{29}$$

Also, the square radiuses $R_{-,t}^2$, $t=1,\ldots,c_2$ are

$$R_{-,t}^2=\frac{1}{|\mathcal{I}_{-,t}^R|}\sum_{j\in\mathcal{I}_{-,t}^R}\left|\left|\varphi(x_j)-c_{-,t}\right|\right|^2,\quad t=1,\ldots,c_2,\tag{30}$$

where the index sets $\mathcal{I}_{-,t}^R$, $t=1,\ldots,c_2$ are

$$\mathcal{I}_{-,t}^R=\left\{j\middle|0<\alpha_j<\frac{\gamma_-}{l_-},\ j\in\mathcal{N}_t\right\},\quad t=1,\ldots,c_2.$$

Once the elements $(c_{+,s},R_{+,s}^2)$, $s=1,\ldots,c_1$ and $(c_{-,t},R_{-,t}^2)$, $t=1,\ldots,c_2$ are calculated by (24), (27), (29) and (30), a series of hyperspheres

$$\begin{aligned}&\left|\left|\varphi(x)-c_{+,s}\right|\right|^2\le R_{+,s}^2,\quad s=1,\ldots,c_1,\\ &\left|\left|\varphi(x)-c_{-,t}\right|\right|^2\le R_{-,t}^2,\quad t=1,\ldots,c_2\end{aligned}\tag{31}$$

are obtained. A new test sample x is assigned to the class $+$ or $-$, depending on which of these hyperspheres given by (31) it lies relatively closest to, i.e.,

$$\begin{aligned}f(x)=\arg\min_{+,-}\Bigg\{&\min_{s=1,\ldots,c_1}\left[\frac{\left|\left|\varphi(x)-c_{+,s}\right|\right|^2}{\frac{l_{+,s}}{l_+}R_{+,s}^2}\right],\\ &\min_{t=1,\ldots,c_2}\left[\frac{\left|\left|\varphi(x)-c_{-,t}\right|\right|^2}{\frac{l_{-,t}}{l_-}R_{-,t}^2}\right]\Bigg\}.\end{aligned}\tag{32}$$

In summary, our STHSVM algorithm for pattern recognition is listed as follows:

Table 1 Attributes of the toy *XOR* and *Hex* datasets

Set	Class	Distribution	Prob.	Mean	Covariance
XOR	Class I	Gauss distr. I_1	0.5	$[2.5; 2.5]$	$[1.5, 0; 0, 1.5]$
		Gauss distr. I_2	0.5	$[-2.5; -2.5]$	$[1.0, 0; 0, 1.0]$
	Class II	Gauss distr. II_1	0.5	$[-2.5; 2.5]$	$[1.5, 0; 0, 1.5]$
		Gauss distr. II_2	0.5	$[2.5; -2.5]$	$[1.0, 0; 0, 1.0]$
Hex	Class I	Gauss distr. I_1	$0.\dot{3}$	$T_0[3; 0]^a$	$T_0[1, 0; 0, 0.25]T_0^T$
		Gauss distr. I_2	$0.\dot{3}$	$T_2[3; 0]$	$T_2[1, 0; 0, 0.25]T_2^T$
		Gauss distr. I_3	$0.\dot{3}$	$T_4[3; 0]$	$T_4[1, 0; 0, 0.25]T_4^T$
	Class II	Gauss distr. II_1	$0.\dot{3}$	$T_1[3; 0]$	$T_1[1, 0; 0, 0.25]T_1^T$
		Gauss distr. II_2	$0.\dot{3}$	$T_3[3; 0]$	$T_3[1, 0; 0, 0.25]T_3^T$
		Gauss distr. II_3	$0.\dot{3}$	$T_5[3; 0]$	$T_5[1, 0; 0, 0.25]T_5^T$

[a] $T_k = [\cos(k\pi/3), -\sin(k\pi/3); \sin(k\pi/3), \cos(k\pi/3)]$, $k = 0, \ldots, 5$

Algorithm 1 The STHSVM algorithm

1. Set the parameters $v_{\pm}$, $\gamma_{\pm}$, and kernel function;
2. Determine the clusters $\mathcal{P}_1, \ldots, \mathcal{P}_{c_1}$ for Class $+$ and the clusters $\mathcal{N}_1, \ldots, \mathcal{N}_{c_2}$ for Class $-$ according to some clustering technique;
3. Optimize the optimization problems (26) and (28) by some optimization technique;
4. Compute $(c_{+,s}, R^2_{+,s})$, $s = 1, \ldots, c_1$ and $(c_{-,t}, R^2_{-,t})$, $t = 1, \ldots, c_2$ by (24), (27), (29) and (30);
5. Predict the label for a new point x by (32).

4 Experiments

In this section, we run a series of experiments systematically on both toy and real-world classification problems to evaluate the proposed STHSVM algorithm. First, we present two synthetic datasets, i.e., the *XOR* and *Hex* datasets, to intuitively compare STHSVM with THSVM. On real-world problems, several datasets in the UCI database are used to evaluate the classification accuracies derived from STHSVM in comparison to some other algorithms, including the TWSVM [10], TPMSVM [14], SRSVM [27], SRPTSVM [28], and THSVM [19]. Remark that the regularization terms are introduced in the TPMSVM, which is helpful to the generalization performance. Here only Gaussian kernel is employed for nonlinear problems. For these classifiers, the kernel widths and the regularization parameters are selected from the set $\{2^{-9}, \ldots, 2^{10}\}$ by cross-validation. While the v/c values in TPMSVM are selected from the set $\{0.05, 0.1, \ldots, 0.95\}$. All methods are implemented in MATLAB[1] on Windows XP running on a PC.

[1] Available at: http://www.mathworks.com.

4.1 Toy datasets

In this part we compare our method with THSVM on the 2-D *XOR* and *Hex* problems, in which the points are randomly generated under Gaussian distributions in each class. Table 1 describes the corresponding attributes of the *XOR* and *Hex* problems. It can be easily seen that, for the two problems, the two classes are composed of some clusters and these clusters have totally different distributions. Thus, in these cases, the structural information within the classes may be more important than the discriminative information between the classes. For each cluster of the two problems, we randomly generate 100 training points and 500 test points, respectively.

For the *XOR* and *Hex* problems, we use the linear and kernel STHSVM and kernel THSVM classifiers to find the decision bounds. Remark that the linear THSVM can not successfully obtain the suitable separating bound for this problem, Figs. 1 and 2 show the one-run training results obtained by the linear and kernel STHSVM and kernel THSVM on these two problems. Due to the formal neglect of the structural information within the classes, kernel THSVM cannot differentiate the different data occurrence trends, i.e., the clusters in each class. Then, the derived hyperspheres for two classes only as possibly as cover the points in the corresponding classes. Specifically, it can be found from Figs. 1a and 2a that the obtained hyperspheres cover the same area, i.e., they can not successfully depict the data, since the structural information under cluster granularity is ignored. Different to the THSVM classifier, STHSVM embeds the structure information within the classes into the optimization problems. Then, STHSVM should get more reasonable discriminant boundaries than THSVM which basically accord with the data occurrence trend, and thus has the best classification performance than the other classifiers in theory. Figures 1b, c and 2b, c confirm this conclusion, in which the results show that it

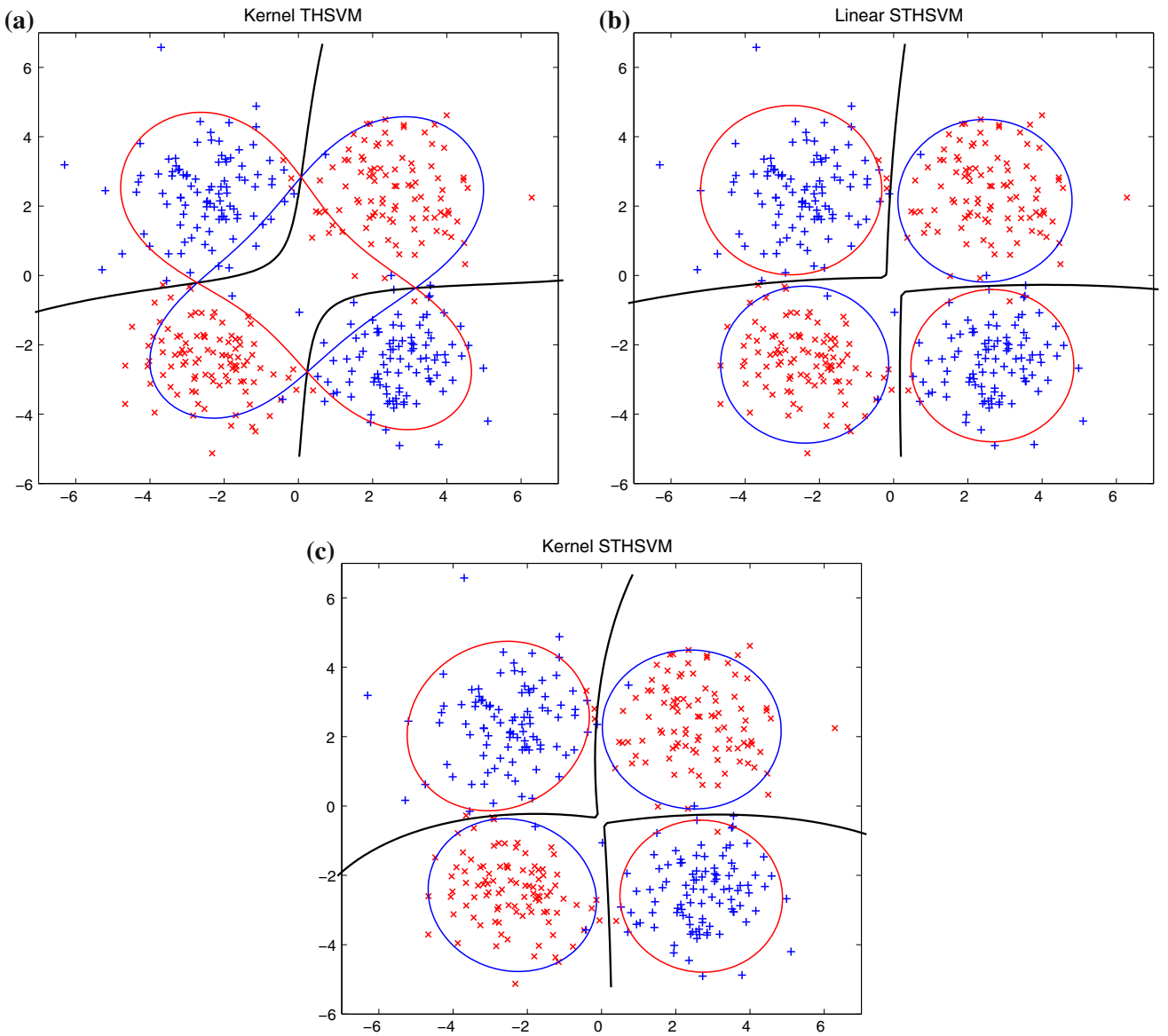

Fig. 1 Training results of kernel THSVM (**a**), linear STHSVM (**b**), and kernel STHSVM (**c**) on the *XOR* problem. The two classes of points are marked by "×" and "+", the decision bounds of these classifiers are marked by *solid curves* in *black*, and the hyperspheres of these classifiers marked by *solid curves* in *blue* and *red*, respectively

obtains a better separating bound than THSVM. To further explain the conclusion, we make ten independent runs on the *XOR* and *Hex* problems and compare with the results of the two classifiers, listed in Table 2. It can be found that our linear and kernel STHSVM obtains the better performance than the kernel THSVM classifier.

To further explain the performance of the proposed STHSVM classifier, in Figs. 3 and 4, we depict the two-dimensional scatter plots for kernel THSVM and linear/nonlinear STHSVM on the *XOR* and *Hex* problem with $100 + 100$ and $150 + 150$ test points, respectively. The plots are obtained by plotting test points with coordinates (d_1, d_2). Here, d_+^i and d_-^i are the relative 'distances' of a test point x_i to the centers of the positive and negative hyperspheres for the THSVM, i.e., $d_\pm^i = ||\varphi(x_i) - c_\pm||/R_\pm$ for the THSVM. While d_+^i and d_-^i are the minimum relative 'distances' of a test point x_i to the centers of the positive and negative hyperspheres for the STHSVM, i.e., $d_+^i = \min_s \left[\sqrt{l_+}||\varphi(x_i) - c_{+,s}||/\sqrt{l_{+,s}}R_{+,s}\right]$ and $d_-^i = \min_t \left[\sqrt{l_-}||\varphi(x_i) - c_{-,t}||/\sqrt{l_{-,t}}R_{-,t}\right]$ for the STHSVM. In short, the point x_i is assigned to class $+1$ if the value of d_+^i is less than d_-^i and vice versa. In Figs. 3 and 4, each point is marked as "○" if its class label is $+1$ and "□" otherwise. Obviously, the two-dimensional projections for test points

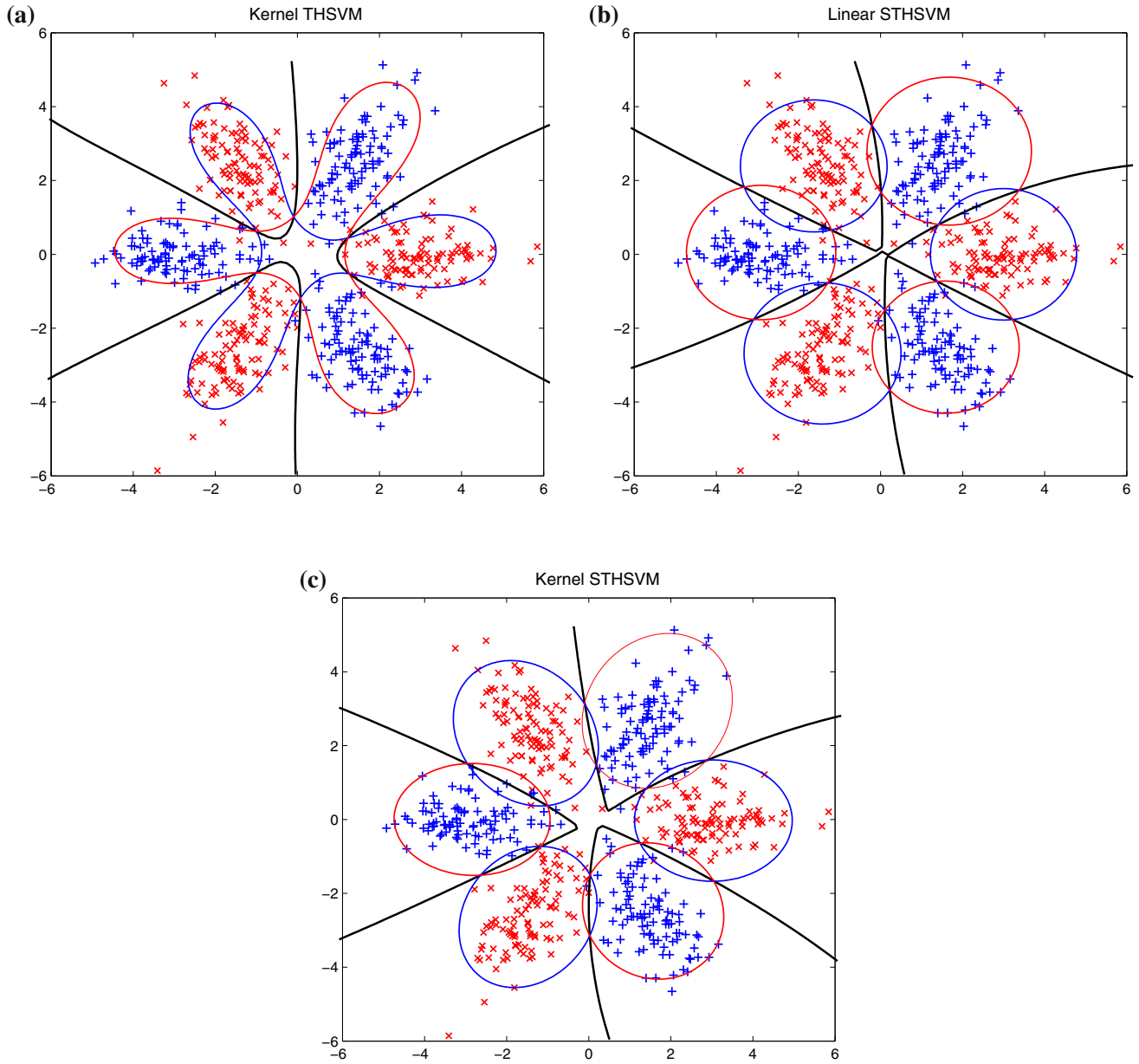

Fig. 2 Training results of kernel THSVM (**a**), linear STHSVM (**b**), and kernel STHSVM (**c**) on the *Hex* problem. The two classes of points are marked by "×" and "+", the decision bounds of these classifiers are marked by *solid curves* in *black*, and the hyperspheres of these classifiers marked by *solid curves* in *blue* and *red*, respectively

Table 2 Results of linear/kernel STHSVM and kernel THSVM on *XOR* and *Hex* datasets

Dataset	Kernel THSVM	Linear STHSVM	Kernel STHSVM
XOR	96.15 ± 0.72	97.76 ± 0.64	97.80 ± 0.50
Hex	96.05 ± 0.85	97.17 ± 0.70	97.20 ± 0.68

indicate how well the classification criterion is able to discriminate between the two classes. It can be seen that for the kernel THSVM, most points are covered by the corresponding hyperspheres and are far from the opposite hyperspheres, i.e., the corresponding d^i_+'s or d^i_-'s are not larger than one. This indicates the hyperspheres in the THSVM can effectively depict the data characteristic of classes. However, for the linear and kernel STHSVM, it not only can be found that most points are covered by the corresponding hyperspheres and are far from the opposite hyperspheres, but also is more robust than the THSVM. In fact, this is because the STHSVM embeds the structure information within the classes into the optimization problems.

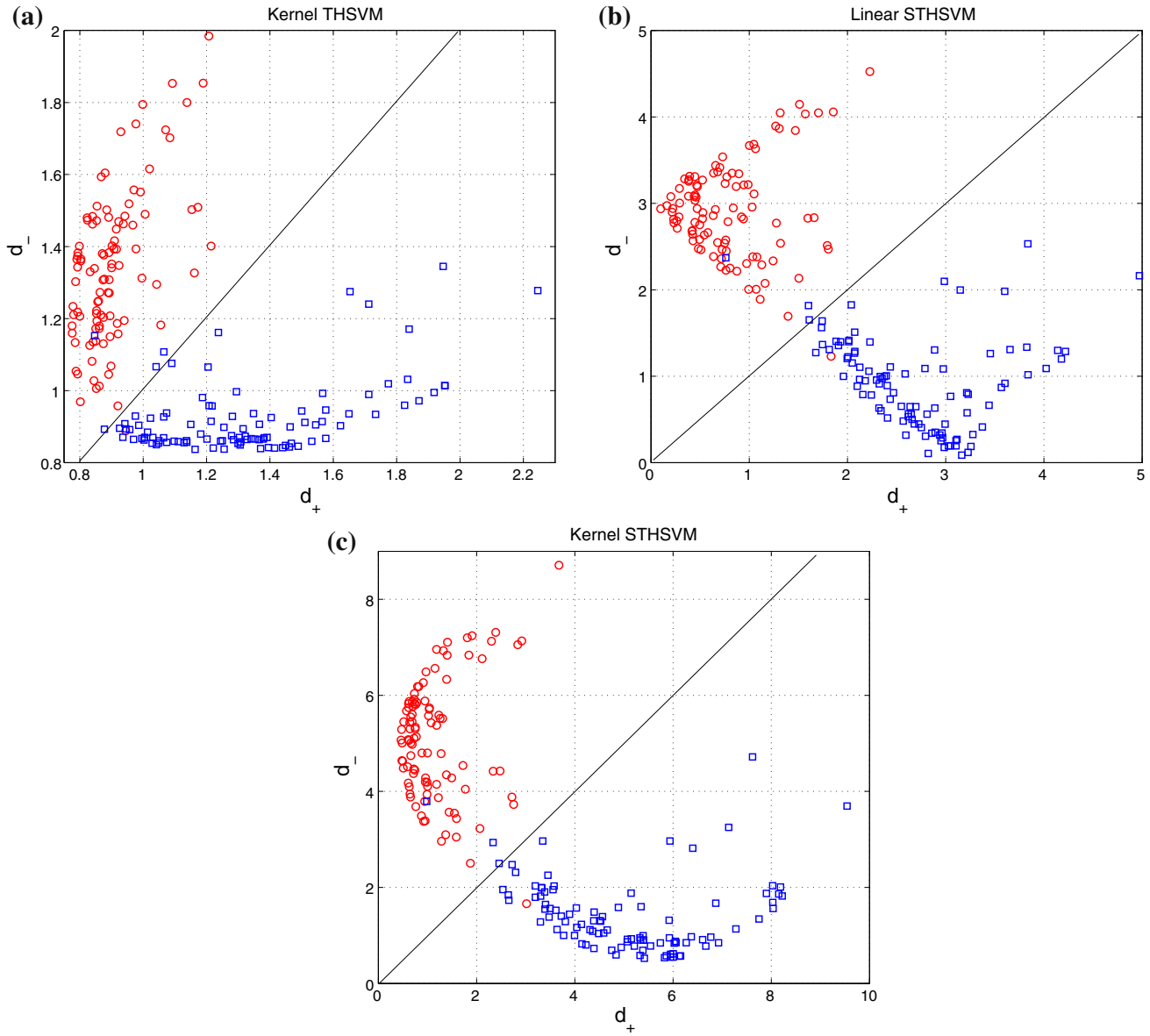

Fig. 3 Two-dimensional projections for test points from the *XOR* dataset with the kernel THSVM (**a**), linear STHSVM (**b**), and kernel STHSVM (**c**)

4.2 Benchmark datasets

In this section, we compare with the performance of TWSVM, TPMSVM, SRSVM, SRPTSVM, THSVM, and STHSVM on the 13 benchmark datasets [34] in that order: *Banana* (B), *Breast Cancer* (BC), *Diabetes* (D), *Flare* (F), *German* (G), *Heart* (H), *Image* (I), *Ringnorm* (R), *Splice* (S), *Thyroid* (Th), *Titanic* (T), *Twonorm* (Tw), and *Waveform* (W). In particular, we use in each problem the train-test splits given in that reference (100 for each dataset except for *Image* and *Splice*, where only 20 splits are given).

In Table 3, we report the training time of one-run and the average test accuracies of linear TWSVM, TPMSVM, SRSVM, SRPTSVM, THSVM, and STHSVM on these benchmark datasets. For the linear SRPTSVM classifier, we adopt the recursive strategy to find the best prediction performance. Table 4 lists the training time of one-run and the average test accuracies of nonlinear TWSVM, TPMSVM, SRSVM, SRPTSVM, THSVM, and STHSVM with Gaussian kernels. From these results, we can find that STHSVM obtains the best learning results than THSVM and other classifiers for most datasets. In fact, this is because STHSVM embeds the structural information of each class under cluster granularity into its two optimization problems, which is more helpful to further improve the learning performance. In addition, it can be found that compared with the other methods, SRSVM and

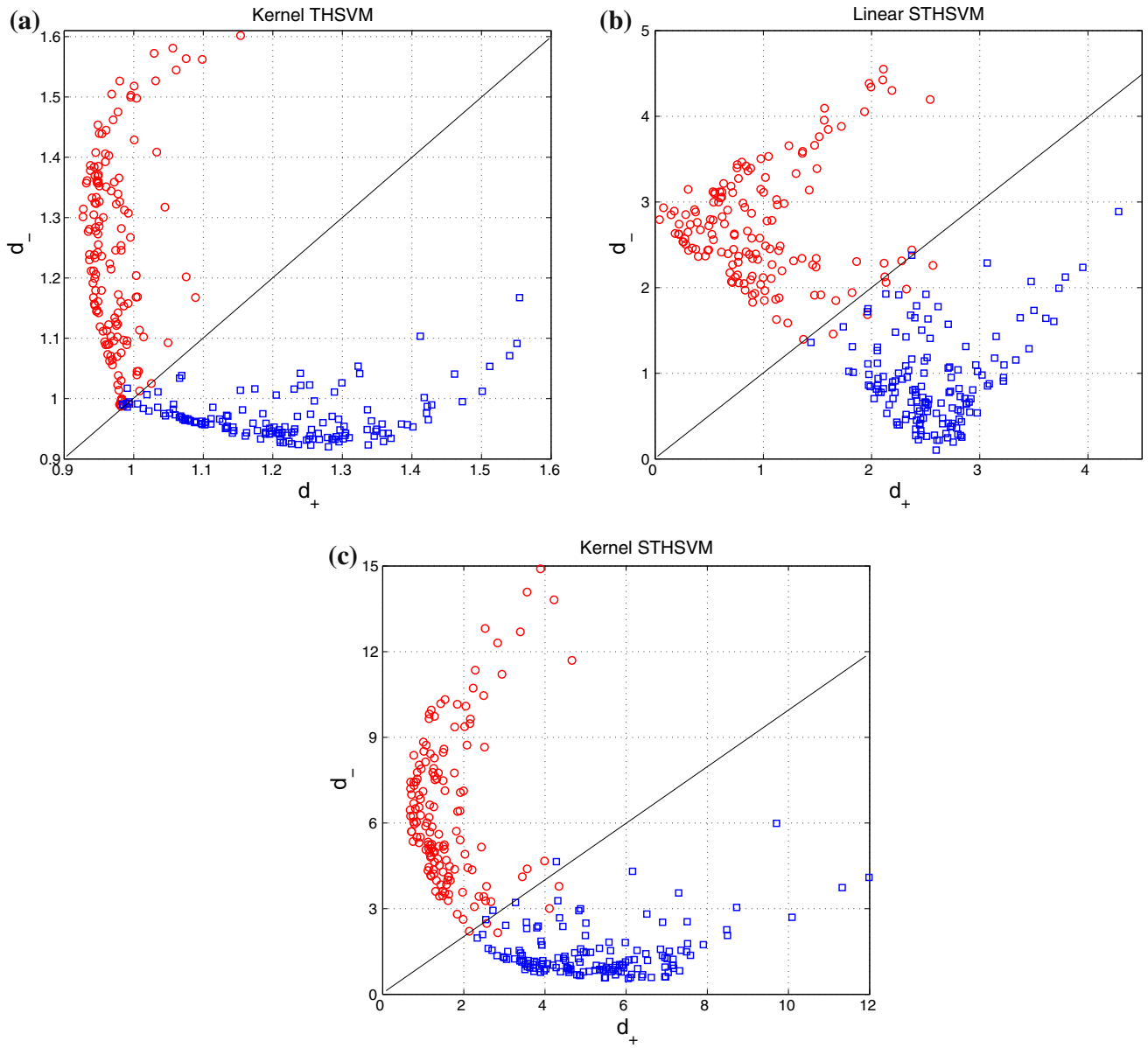

Fig. 4 Two-dimensional projections for test points from the *Hex* dataset with the kernel THSVM (**a**), linear STHSVM (**b**), and kernel STHSVM (**c**)

SRPTSVM obtain better generalization performance than TWSVM and TPMSVM for many datasets. This is also because the two methods successfully embed the data structural information into their optimization problems. However, the results in Tables 3 and 4 show that our method outperforms SRSVM and SRPTSVM for many datasets. A possible reason is that it uses a more reasonable strategy to depict the data structure than the latter. In order to find out whether STHSVM is significantly better than the other algorithms, we perform the t test on the classification results to calculate the statistical significance of STHSVM. The null hypothesis H_0 demonstrates that there is no significant difference between the mean numbers of patterns correctly classified by STHSVM and the other algorithms. If the hypothesis H_0 of each dataset is rejected at the 5 % significance level, i.e., the t test value is more than 1.734, the corresponding results in Tables 3 and 4 is denoted "*". Consequently, as shown in Tables 3 and 4, it can be clearly found that STHSVM possesses significantly superior classification performance compared with the other classifiers on the most datasets. This just accords with our conclusions. As for the learning time of these methods, remark that it need find the cluster-based structural information through some clustering technique, which leads to some extra learning time compared with THSVM. However, it can be seen that, for most datasets, the proposed STHSVM with different kernels has a comparable

Table 3 Prediction accuracies and learning time (in s) of linear TWSVM, TPMSVM, SRPTSVM, SRSVM, THSVM, and STHSVM on benchmark datasets

Set Size	TWSVM *Acc.* (%) *Time* (*s*)	TPMSVM *Acc.* *Time*	SRPTSVM *Acc.* *Time*	SRSVM *Acc.* *Time*	THSVM *Acc.* *Time*	STHSVM *Acc.* *Time*
B	55.78 ± 5.84*[a]	61.90 ± 2.54	62.23 ± 4.68	58.63 ± 2.97*	60.15 ± 3.90*	63.40 ± 2.57
400/4,900 × 2	0.204	0.970	0.912	3.784	0.457	0.414/0.301[b]
BC	71.43 ± 4.80*	71.50 ± 4.10*	72.78 ± 4.76*	71.28 ± 4.35*	72.12 ± 3.64*	74.10 ± 3.12
200/77 × 9	0.092	0.074	0.204	0.422	0.076	0.105/0.052
D	76.68 ± 2.05	75.03 ± 2.40*	77.03 ± 2.41	76.24 ± 2.33*	76.21 ± 2.54*	78.13 ± 2.60
468/300 × 8	0.203	1.317	1.136	3.780	1.035	1.037/0.769
F	66.80 ± 1.60*	67.25 ± 1.54	67.50 ± 3.04	67.16 ± 1.85	67.49 ± 2.87	67.78 ± 2.12
666/400 × 9	0.511	2.779	3.120	5.320	0.903	0.578/0.355
G	75.73 ± 1.90	73.32 ± 2.82*	73.26 ± 3.85*	73.19 ± 3.10*	73.25 ± 3.01*	75.61 ± 3.43
700/300 × 20	0.970	2.612	1.636	5.367	1.115	1.022/0.803
H	84.30 ± 3.13	83.90 ± 3.18*	85.25 ± 3.21	83.76 ± 3.52*	83.01 ± 2.70*	85.16 ± 3.22
170/100 × 13	0.026	0.192	0.292	0.310	0.205	0.212/0.188
I	79.08 ± 2.24	79.45 ± 2.06	78.91 ± 1.30	79.02 ± 1.98	78.75 ± 1.97	79.17 ± 2.30
1300/1,010 × 18	5.307	14.274	28.431	156.253	6.032	5.175/3.016
R	75.50 ± 0.72*	76.38 ± 0.62*	76.67 ± 0.56	76.45 ± 0.70	76.52 ± 0.76	77.12 ± 0.71
400/7,000 × 20	0.391	0.935	2.934	4.490	0.652	0.647/0.408
S	83.45 ± 0.84*	83.72 ± 0.98*	84.62 ± 1.05	84.15 ± 1.68	83.92 ± 1.16	84.48 ± 1.49
1,000/2,175 × 60	2.708	7.221	20.631	17.245	4.287	3.152/2.076
Th	82.80 ± 3.18*	89.13 ± 3.88	92.27 ± 3.72*	89.38 ± 3.90	89.40 ± 3.70	90.32 ± 3.52
140/75 × 5	0.021	0.143	1.024	0.433	0.073	0.078/0.060
T	77.60 ± 0.37*	77.64 ± 0.42*	78.42 ± 0.37	78.59 ± 0.65	77.84 ± 0.61	78.82 ± 0.54
150/2,051 × 3	0.017	0.138	0.426	0.692	0.089	0.105/0.0921
Tw	97.21 ± 0.24	97.66 ± 0.08	97.68 ± 0.15	97.68 ± 0.22	97.56 ± 0.23	97.72 ± 0.25
400/7,000 × 20	0.081	0.939	6.821	5.076	0.242	0.272/0.214
W	82.72 ± 0.81*	87.73 ± 0.37*	87.95 ± 0.41	88.05 ± 0.89	87.70 ± 0.41*	88.78 ± 0.68
400/4,600 × 21	0.077	1.478	3.621	4.873	1.128	1.050/0.982

[a] '*'The difference between STHSVM and this algorithm is significant at 5 % significance level, i.e., t value > 1.734

[b] The time for clustering in STHSVM

speed with THSVM. In fact, this is because this proposed STHSVM only needs to optimize a series of smaller-sized optimization problems compared with THSVM, which leads it to have a much faster speed than THSVM. Tables 3 and 4 also confirm this conclusion. In fact, the time for clustering in this STHSVM is listed in tables indicates this method is much efficient. In summary, these simulation results show that our STHSVM not only obtains a better generalization performance than these related methods, but also has a fast learning speed.

5 Conclusions

The twin-hypersphere support vector machine (THSVM) [19] for binary classification seeks two hypersphres by solving two SVM-type problems, one for each class, to make the points in each class are covered as many as possibly by one hyperphere. Then, it classifies a new point according to which hypersphere is relatively closest to. However, it only considers the global structure information of each class, which leads to hardly extend to real-world problems.

In this paper, under the structural granularity [27], which characterizes a series of data structures involved in the various classifier design ideas, we have introduced an improved THSVM named the structural information-based THSVM (STHSVM) classifier. In each optimization problem of STHSVM, a data structural-information derived from the cluster granularity [27] is absorbed into the learning process. Further, STHSVM introduces a different probability sum of projected center for each

Table 4 Prediction accuracies and learning time of nonlinear TWSVM, TPMSVM, SRPTSVM, SRSVM, THSVM, and STHSVM on benchmark datasets

Set	TWSVM *Acc.* (%) *Time* (*s*)	TPMSVM *Acc.* *Time*	SRPTSVM *Acc.* *Time*	SRSVM *Acc.* *Time*	THSVM *Acc.* *Time*	STHSVM *Acc.* *Time*
B	88.52 ± 0.92*	88.61 ± 0.59*	89.92 ± 0.42	88.72 ± 0.92	88.60 ± 0.60*	90.14 ± 0.58
	0.271	0.116	1.365	4.236	0.312	0.327/0.231
BC	72.71 ± 4.26*	74.15 ± 3.81	74.52 ± 3.17	74.63 ± 4.21	74.25 ± 2.58	74.50 ± 2.74
	0.126	0.023	0.557	0.429	0.048	0.056/0.040
D	75.78 ± 2.60*	76.37 ± 2.23	76.82 ± 1.76	76.82 ± 1.94	75.74 ± 1.96*	76.75 ± 2.35
	0.577	0.130	1.071	5.481	0.521	0.435/0.311
F	65.28 ± 1.96*	67.85* ± 1.93*	69.12 ± 1.84	69.95 ± 2.01	67.80 ± 1.46*	69.84 ± 1.52
	5.134	4.340	6.582	11.097	4.535	4.125/2.906
G	75.98 ± 1.93*	76.82 ± 2.72	77.21 ± 2.10	77.35 ± 2.35	75.00 ± 2.27*	77.32 ± 2.29
	1.457	0.200	7.452	18.863	1.645	1.615/1.412
H	83.40 ± 3.31*	84.32 ± 3.11*	86.46 ± 3.51	84.65 ± 3.52*	84.27 ± 3.25*	86.52 ± 3.49
	0.037	0.013	0.274	0.269	0.049	0.056/0.044
I	95.29 ± 0.77*	96.90 ± 0.96	97.24 ± 0.42	97.53 ± 0.72	96.16 ± 0.60*	97.74 ± 0.62
	17.850	11.604	37.315	289.215	18.745	15.363/7.903
R	98.42 ± 0.16	98.24 ± 0.45	98.47 ± 1.16	98.46 ± 1.22	98.43 ± 0.25	98.47 ± 0.50
	0.578	0.232	1.426	5.434	0.973	1.015/0.902
S	87.80 ± 0.66*	88.74 ± 1.28*	88.69 ± 1.42*	89.91 ± 0.52	88.72 ± 1.12*	89.96 ± 0.95
	8.252	6.323	9.315	156.241	8.160	7.256/3.120
Th	95.52 ± 2.10	94.85 ± 2.15*	95.82 ± 2.21	95.40 ± 2.75	95.56 ± 2.11	96.12 ± 2.02
	0.023	0.011	0.124	0.311	0.075	0.095/0.086
T	77.16 ± 0.45*	77.70 ± 1.51	77.86 ± 1.29	78.16 ± 1.04	77.36 ± 1.36	78.21 ± 1.32
	0.034	0.009	0.112	0.296	0.104	0.115/0.090
Tw	97.42 ± 0.18	96.74 ± 0.79*	97.62 ± 0.50	97.10 ± 0.52	97.21 ± 0.43	97.60 ± 0.59
	1.041	0.219	4.642	5.142	1.110	1.072/0.795
W	89.42 ± 0.97*	90.83 ± 0.48	91.15 ± 0.55	91.01 ± 0.72	89.80 ± 0.64*	91.16 ± 0.82
	0.252	0.248	2.730	6.374	0.959	0.874/0.697

* The difference between STHSVM and this algorithm is significant at 5 % significance level, i.e., t value > 1.734

point to depict the cluster granularity-based structural information. The experiments have confirmed that this STHSVM successfully embeds into the cluster granularity-based structural information and obtains the good performance. The idea in this method can be easily extended to some other TWSVM classifiers. There still exists some future work. For example, we only apply our method into the middle-scale classification problems since it has a large cost to cluster large-scale datasets. In addition, another problem is to discuss the relationship between the performance and the cluster number. Also, the parameter-selection problem is an important further problem.

Acknowledgments The authors would like to genuinely thank the anonymous reviewers for their constructive comments and suggestions. This work is partly supported by the National Natural Science Foundation of China (61202156), the National Natural Science Foundation of Shanghai (12ZR1447100), and the program of Shanghai Normal University (DZL121).

References

1. Boser B, Guyon L, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, ACM Press, Pittsburgh, 1992, pp 144–152
2. Vapnik VN (1995) The natural of statistical learning theory. Springer, New York
3. Vapnik VN (1998) Statistical learning theory. Wiley, New York
4. He Q, Wu C (2011) Separating theorem of samples in Banach space for support vector machine learning. Int J Mach Learn Cybernet 2(1):49–54
5. Wang X, He Q, Chen D, Yeung D (2005) A genetic algorithm for solving the inverse problem of support vector machines. Neurocomputing 68:225–238
6. Wang X, Lu S, Zhai J (2008) Fast fuzzy multi-category SVM based on support vector domain description. Int J Pattern Recognit Artif Intell 22(1):109–120
7. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of IEEE Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997, pp 130–136

8. El-Naqa I, Yang Y, Wernik M, Galatsanos NP, Nishikawa RM (2002) A support vector machine approach for detection of microclassification. IEEE Trans Med Imaging 21(12):1552–1563
9. Schölkopf B, Tsuda K, Vert J-P (2004) Kernel methods in computational biology. MIT Press, Cambridge
10. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
11. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
12. Ghorai S, Mukherjee A, Dutta PK (2009) Nonparallel plane proximal classifier. Signal Process 89(4):510–522
13. Peng X (2010) A *v*-twin support vector machine (*v*-TSVM) classifier and its geometric algorithms. Inform Sci 180(15):3863–3875
14. Peng X (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10–11):2678–2692
15. Chen X, Yang J, Ye Q, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recognit 44(10–11):2643–2655
16. Shao YH, Chen WJ, Deng NY (2014) Nonparallel hyperplane support vector machine for binary classification problems. Inform Sci 263:22–35
17. Peng X (2010) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372
18. Peng X (2012) Efficient twin parametric insensitive support vector regression model. Neurocomputing 79:26–38
19. Peng X, Xu D (2013) A twin-hypersphere support vector machine classifier and the fast learning algorithm. Inform Sci 221(1):12–27
20. Yeung D, Wang D, Ng W, Tsang E, Zhao X (2007) Structured large margin machines: sensitive to data distributions. Mach Learn 68(2):171–200
21. Belkin M, Niyogi P, Sindhwani V (2004) Manifold regularization: a geometric framework for learning from examples, Dept. Comput. Sci., Univ. Chicago, Chicago, IL, Techique report, TR-2004-06, Aug 2004
22. Chen WJ, Shao YH, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. Int J Mach Learn Cybernet 5(3):459–468
23. Rigollet P (2007) Generalization error bounds in semi-supervised classification under the cluster assumption. J Mach Learn Res 8:1369–1392
24. Shivaswamy PK, Jebara T (2007) Ellipsoidal kernel machines. In: Proceeding of 12th International Workshop on Artificial Intelligence Statistic, 2007, pp 1–8
25. Lanckriet GRG, Ghaoui LE, Bhattacharyya C, Jordan MI (2002) A robust minimax approach to classfication. J Mach Learn Res 3:555–582
26. Huang K, Yang H, King I, Lyu MR (2008) Maxi–min margin machine-learning large margin classifiers locally and globally. IEEE Trans Neural Netw 19:260–272
27. Xue H, Chen S, Yang Q (2011) Structural regularized support vector machine: a framework for structural large margin classifier. IEEE Trans Neural Netw 22:573–587
28. Peng X, Xu D (2014) Structural regularized projection twin support vector machine for data classification. Inform Sci 279:416–432
29. Hartigan JA, Wong MA (1979) A *k*-means clustering algorithm. Appl Stat 28(1):100–108
30. Lu S-Y, Fu KS (1978) A sentence-to-sentence clustering procedure for pattern analysis. IEEE Trans Syst Man Cybernet 8(5):381–389
31. Zadeh LA (1965) Fuzzy sets. Inform Control 8:338–353
32. Ward JH (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58(301):236–244
33. Salvador S, Chan P (2004) Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: Proc. 16th IEEE Int. Conf. Tools Artif. Intell., Nov 2004, pp 576584
34. Rätsch G (2000) Benchmark repository, datasets available at http://ida.first.fhg.de/projects/bench/benchmarks.htm