Progress in Artificial Intelligence (2018) 7:167–175
https://doi.org/10.1007/s13748-018-0141-0

REGULAR PAPER

Improvements on twin-hypersphere support vector machine using local density information

Qing Ai[1,2] · Anna Wang[1] · Yang Wang[1] · Haijing Sun[1]

Received: 17 July 2017 / Accepted: 3 January 2018 / Published online: 15 January 2018

Abstract
In this paper, we propose a novel binary classifier called twin-hypersphere support vector machine with local density information (LDTHSVM). Firstly we extract local density for each training sample and treat it as the weight of that sample, next prune training dataset according to these local density degrees, finally introduce these local density degrees into twin-hypersphere support vector machine (THSVM) and reconstruct classification model on the pruned training dataset. LDTHSVM not only inherits good properties from THSVM, but also gives more robust description for dataset. The experimental results on synthetic and publicly available benchmark datasets show the excellent performance of the LDTHSVM classifier in terms of classification accuracy and learning time.

Keywords Twin-hypersphere support vector machine · Local density · Pruning dataset · Pattern recognition

1 Introduction

Support vector machine (SVM) [27,34], as computationally powerful tools for pattern recognition, has already obtained excellent performance in many fields [12,13,17,29,35]. The main idea of SVM is to seek an optimal hyperplane that can separate two classes of samples with maximal margin. The hyperplane can be obtained by solving a quadratic programming problem (QPP). SVM can also successfully solve nonlinear classification problems by using kernel trick. If the size of training set is n, the learning complexity of classical SVM is $O(n^3)$. Therefore, one of the key issues for SVM is the slow learning speed for large-scale training datasets. To improve the learning speed of the classical SVM, many efficient training algorithms have been proposed with comparable classification accuracy, such as sequential minimal optimization (SMO) [3,8,23], decomposition method [6,16], geometric algorithms [7,15], etc.

✉ Qing Ai
lyaiqing@126.com

1 College of Information Science and Engineering, Northeastern University, Shenyang 110819, Liaoning, People's Republic of China

2 School of Software, University of Science and Technology Liaoning, Anshan 114051, Liaoning, People's Republic of China

Recently, a generalized eigenvalue proximal SVM (GEPSVM) was proposed [14], whose main idea aims at constructing a pair of nonparallel hyperplanes, each hyperplane is proximal to the samples of the corresponding class and is far from the others. By solving two generalized eigenvalue problems, the two nonparallel hyperplanes of the GEPSVM can be efficiently obtained. But its classification accuracy is poor in many practical problems, compared with the classical SVM. A twin SVM (TWSVM) was proposed by Jayadeva for binary classification [5]. Similar to GEPSVM in spirit, TWSVM also aims at seeking two nonparallel hyperplanes, and each hyperplane is closer to one class and is at a distance of at least one from the other. The two nonparallel hyperplanes can be obtained in the TWSVM by solving two smaller sized QPPs. The experimental results [5] show that the TWSVM works faster than the classical SVM and compares favorable with the classical SVM in the light of classification accuracy. Some extensions to TWSVM include the smooth TWSVM [9], least squares TWSVM [10], localized TWSVM [33], twin bounded SVM [25], twin parametric-margin SVM [19], ν-TWSVM [18], structural TWSVM [24], nonparallel SVM [26], twin mahalanobis distance-based SVM [20], multi-label TSVM [2], twin support vector clustering [28], etc.

Different from TWSVM which seeks two nonparallel hyperplanes, Peng proposed twin-hypersphere support vector machine (THSVM) [22], which uses two hyperspheres to

depict two classes of samples. The idea may be more reasonable for many practical datasets. The two hyperspheres can be obtained in the THSVM by solving two QPPs. The THSVM can avoid the inversions of two matrices that appear in the TWSVM, which makes the THSVM be more efficient than the TWSVM. Recently the THKSVM [30] and Pin-M3HM [31] as extensions of the THSVM were also proposed, respectively.

In this paper, we propose a novel classifier called THSVM with local density information (LDTHSVM) for binary classification. Firstly we extract local density for each sample and treat it as the weight of that sample, then prune training dataset according to these local density degrees, finally introduce these local density degrees into THSVM and reconstruct more robust classification model. Computational comparisons with some classical classification algorithms have been made on the synthetic and publicly available benchmark datasets, indicating that the LDTHSVM has better classification performance.

The remaining parts of this paper are organized as follows. Section 2 introduces the classical THSVM. Section 3 discusses the local density degrees of training samples and pruning method of training dataset. Section 4 deduces LDTHSVM in detail. Section 5 gives the computational complexity of LDTHSVM. In Sect. 6, experimental results on the synthetic and publicly available benchmark datasets are shown and conclusions are outlined in Sect. 7.

2 Related works

2.1 Notations

In this paper, we consider the binary classification problem with the dataset $D = \{(x_i, y_i)\}_{i=1}^{l}$, where $x_i \in R^d$ is a training sample labeled $y_i \in \{1, -1\}$. Further, we denote by matrix $\mathrm{A} \in R^{l_+ \times d}$ and $\mathrm{B} \in R^{l_- \times d}$ the positive and negative samples, respectively. Finally a mapping $\varphi(\cdot)$ is introduced to map R^d into some feature space Z. It is possible to use some kernel function $K(x_i, x_j)$ to represent the inner product in Z, i.e., $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$.

2.2 Review of THSVM

The THSVM [22] determines two hyperspheres, rather than two nonparallel hyperplanes, to describe two classes of samples in the feature space Z:

$$\|\varphi(x) - a_+\|^2 = R_+^2 \quad \text{and} \quad \|\varphi(x) - a_-\|^2 = R_-^2, \tag{1}$$

where $a_\pm$ and $R_\pm$ are, respectively, the centers and radii of the corresponding hyperspheres.

The THSVM classifier is obtained by solving a pair of QPPs as follows:

$$\begin{aligned} \min\ & R_+^2 - \frac{v_1}{l_-}\sum_{j=1}^{l_-}\|\varphi(B_j) - a_+\|^2 + \frac{c_1}{l_+}\sum_{i=1}^{l_+}\xi_i, \\ \text{s.t.}\ & \|\varphi(A_i) - a_+\|^2 \le R_+^2 + \xi_i, \\ & R_+^2 \ge 0, \xi_i \ge 0, i = 1, \ldots, l_+, \end{aligned} \tag{2}$$

$$\begin{aligned} \min\ & R_-^2 - \frac{v_2}{l_+}\sum_{i=1}^{l_+}\|\varphi(A_i) - a_-\|^2 + \frac{c_2}{l_-}\sum_{j=1}^{l_-}\xi_j, \\ \text{s.t.}\ & \|\varphi(B_j) - a_-\|^2 \le R_-^2 + \xi_j, \\ & R_-^2 \ge 0, \xi_j \ge 0, j = 1, \ldots, l_-, \end{aligned} \tag{3}$$

where $c_1, c_2, v_1, v_2 > 0$ are the penalty factors prespecified in advance, and ξ_i, ξ_j are the slack variables.

The dual QPPs of (2) and (3) can be obtained:

$$\begin{aligned} \min\ & \sum_{i_1, i_2 = 1}^{l_+}\alpha_{i_1}\alpha_{i_2}K(A_{i_1}, A_{i_2}) \\ & -\sum_{i=1}^{l_+}\alpha_i\Big[\frac{2v_1}{l_-}\sum_{j=1}^{l_-}K(B_j, A_i) + (1 - v_1)K(A_i, A_i)\Big], \\ \text{s.t.}\ & \sum_{i=1}^{l_+}\alpha_i = 1, \\ & 0 \le \alpha_i \le \frac{c_1}{l_+}, i = 1, \ldots, l_+, \end{aligned} \tag{4}$$

$$\begin{aligned} \min\ & \sum_{j_1, j_2 = 1}^{l_-}\beta_{j_1}\beta_{j_2}K(B_{j_1}, B_{j_2}) \\ & -\sum_{j=1}^{l_-}\beta_j\Big[\frac{2v_2}{l_+}\sum_{i=1}^{l_+}K(A_i, B_j) + (1 - v_2)K(B_j, B_j)\Big], \\ \text{s.t.}\ & \sum_{j=1}^{l_-}\beta_j = 1, \\ & 0 \le \beta_j \le \frac{c_2}{l_-}, j = 1, \ldots, l_-. \end{aligned} \tag{5}$$

Once QPPs (4) and (5) are solved, the decision function of THSVM can be written as:

$$f(x) = \mathrm{sgn}\left\{\frac{(\varphi(x) - a_+)^{\mathrm{T}}(\varphi(x) - a_+)}{R_+^2} - \frac{(\varphi(x) - a_-)^{\mathrm{T}}(\varphi(x) - a_-)}{R_-^2}\right\}. \tag{6}$$

3 Pruning method of training dataset

3.1 Local density of training dataset

The noise samples, which may be caused by sampling or instrument error, have many effects on classification ability of THSVM. That is to say, the classification accuracy will be reduced if there are many noise samples in the training set. The noise samples mainly have two types, one is isolated samples with abnormal features, and the other is the samples with wrong label. In this paper, we use the weight of sample to reduce the effect. The weight of sample can be obtained by estimating its local density degree. The process of calculating the weights of samples is as follows:

Firstly, the Euclidean distances d_{ij} between samples x_i and x_j in the feature space are calculated:

$$\begin{aligned} d_{ij}^2 &= \left\|\varphi(x_i) - \varphi(x_j)\right\|^2 \\ &= K(x_i, x_i) - 2K(x_i, x_j) \\ &\quad + K(x_j, x_j), \quad i, j = 1, \ldots, l \text{ and } i \neq j. \end{aligned} \tag{7}$$

Secondly, seek k nearest neighbor region Ω_i of sample x_i and the radius of the region r_i according to (7):

$$\Omega_i = \{x_j | x_j \ is \ k \text{ nearest neighbor of } x_i\}, \tag{8}$$

$$r_i = \max(d_{ij}^2), \ \ j \in \Omega_i, \tag{9}$$

where k is predetermined value.

Then, obtain intra-nearest neighbors of sample x_i:

$$\Lambda_i = \{x_j | x_j \in \Omega_i \ \& \ x_j \text{ and } x_i \text{ belong to the same class}\}. \tag{10}$$

Finally, calculate local density degree d_i of sample x_i according to the following formula:

$$d_i = \sum_{j \in \Lambda_i} \exp\{-\omega * d_{ij}^2 / r\}, \tag{11}$$

where $r = \frac{1}{l}\sum_{i=1}^{l} r_i$ and ω is a weight.

Clearly, this method gives the higher local density degree d_i for the sample in a higher density region: the sample with lower distances from its k nearest neighbors has higher d_i. Moreover, a smaller ω produces higher local density degrees. Generally ω and k are, respectively, set 1 and 7 [11,21].

3.2 Pruning training dataset

To improve classification efficiency, it is necessary to prune samples for large-scale training set. In this paper, the pruning method that uses local density information is proposed as follows:

Firstly, use algorithm presented in Sect. 3.1 to obtain local density degrees d_i of sample x_i.

Then, prune the sample from training dataset whose local density degree is smaller than σ, and remain the sample whose local density degree is bigger than or equal to σ, where pruning threshold σ is a determined by users according to the real problems.

Finally, suppose the pruned training dataset $\bar{D} = \{(\bar{x}_i, \bar{y}_i, d_i^{'})\}_{i=1}^{\bar{l}}$, then denote by matrix $\bar{A} \in R^{\bar{l}_+ \times d}$ and $\bar{B} \in R^{\bar{l}_- \times d}$ the positive and negative samples in the $\bar{D}$, respectively, further let $d_+^{'}$ and $d_-^{'}$ be the weights of $\bar{A}$ and $\bar{B}$, respectively. Scale the weights of remaining samples as follows:

$$\bar{d}_i^{\pm} = \bar{l}_{\pm} \times (d_{\pm}^{'})_i / \sum_{j=1}^{\bar{l}_{\pm}} (d_{\pm}^{'})_j. \tag{12}$$

4 The LDTHSVM classifier

In order to obtain more accurate and efficient classifier for large-scale training datasets with noise, THSVM is improved to be the enhanced version using local density information, called LDTHSVM. Similar to THSVM in spirit, LDTHSVM also tries to construct a pair of hyperspheres, one for each class, such that each hypersphere can cover the samples of the corresponding class as many as possible.

Consider the binary classification problem with the pruned dataset $\bar{D} = \{(\bar{x}_i, \bar{y}_i, \bar{d}_i)\}_{i=1}^{\bar{l}}$, which can be obtained by using the pruning method in Sect. 3.2. Further denote by matrix $\bar{A} \in R^{\bar{l}_+ \times d}$ and $\bar{B} \in R^{\bar{l}_- \times d}$ the positive and negative samples in the $\bar{D}$, respectively, and let $\bar{d}^+ \in R^{\bar{l}_+}$ and $\bar{d}^- \in R^{\bar{l}_-}$ be the weight vectors of $\bar{A}$ and $\bar{B}$. LDTHSVM can be formulated as follows:

$$\begin{aligned} \min \ & R_+^2 - \frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-} \bar{d}_j^- \left\|\varphi(\bar{B}_j) - a_+\right\|^2 + \frac{c_1}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+} \bar{d}_i^+ \xi_i, \\ \text{s.t.} \ & \left\|\varphi(\bar{A}_i) - a_+\right\|^2 \leq R_+^2 + \xi_i, \\ & R_+^2 \geq 0, \xi_i \geq 0, i = 1, \ldots, \bar{l}_+, \end{aligned} \tag{13}$$

$$\begin{aligned} \min \ & R_-^2 - \frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+} \bar{d}_i^+ \left\|\varphi(\bar{A}_i) - a_-\right\|^2 + \frac{c_2}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-} \bar{d}_j^- \xi_j, \\ \text{s.t.} \ & \left\|\varphi(\bar{B}_j) - a_-\right\|^2 \leq R_-^2 + \xi_j, \\ & R_-^2 \geq 0, \xi_j \geq 0, j = 1, \ldots, \bar{l}_-. \end{aligned} \tag{14}$$

From the primal problems (13), we notice that, unlike THSVM, firstly LDTHSVM does not employ all training samples, but use pruned training dataset, which makes classifier more robust and efficient when there exist many noise

samples in the large-scale training set. Secondly, the local density degrees of negative and positive samples are, respectively, added to the second and third terms in the objective function which contributes to the positive center far away from the negative samples with higher local density degrees, in addition, to the positive samples with higher local density degrees covered by the positive hypersphere. Furthermore, we can get similar conclusions from the primal problem (14). Obviously, the LDTHSVM classifier is more reasonable for the practical applications.

The Lagrangian function of (13) is given by

$$\begin{aligned} L = R_+^2 - \frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-\left\|\varphi(\bar{B}_j)-a_+\right\|^2+\frac{c_1}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+\xi_i \\ +\sum_{i=1}^{\bar{l}_+}\alpha_i\left(\left\|\varphi(\bar{A}_i)-a_+\right\|^2-R_+^2-\xi_i\right) \\ -\sum_{i=1}^{\bar{l}_+}r_i\xi_i-\lambda R_+^2, \end{aligned} \tag{15}$$

where $\alpha_i \geq 0$, $r_i \geq 0$, $\lambda \geq 0$, $i = 1, \ldots, \bar{l}_+$ are the Lagrangian multipliers. According to the Karush–Kuhn–Tucker Theorem, the following conditions are satisfied:

$$\frac{2v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-(\varphi(\bar{B}_j)-a_+)-2\sum_{i=1}^{\bar{l}_+}\alpha_i(\varphi(\bar{A}_i)-a_+)=0, \tag{16}$$

$$1-\sum_{i=1}^{\bar{l}_+}\alpha_i-\lambda=0, \tag{17}$$

$$\frac{c_1}{\bar{l}_+}\bar{d}_i^+-\alpha_i-r_i=0\Rightarrow 0\leq\alpha_i\leq\frac{c_1}{\bar{l}_+}\bar{d}_i^+,\ i=1,\ldots,\bar{l}_+, \tag{18}$$

$$\left\|\varphi(\bar{A}_i)-a_+\right\|^2\leq R_+^2+\xi_i,\ i=1,\ldots,\bar{l}_+, \tag{19}$$

$$\alpha_i\left(\left\|\varphi(\bar{A}_i)-a_+\right\|-R_+^2-\xi_i\right)=0, \alpha_i\geq 0, i=1,\ldots,\bar{l}_+, \tag{20}$$

$$r_i\xi_i=0,\ \xi_i\geq 0,\ r_i\geq 0,\ i=1,\ldots,\bar{l}_+, \tag{21}$$

$$\lambda R_+^2=0,\ R_+^2\geq 0,\ \lambda\geq 0. \tag{22}$$

According to (16), (17) and (22), the center of positive hypersphere can be obtained as follows:

$$a_+=\frac{1}{1-\frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-}\left(\sum_{i=1}^{\bar{l}_+}\alpha_i\varphi(\bar{A}_i)-\frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-\varphi(\bar{B}_j)\right). \tag{23}$$

Substituting (17), (18) and (23) into (15) and discarding the constant items, we can obtain the following dual problem of (13):

$$\begin{aligned} \max\ & -t_1\sum_{i_1,i_2=1}^{\bar{l}_+}\alpha_{i_1}\alpha_{i_2}K(\bar{A}_{i_1},\bar{A}_{i_2}) \\ & +t_2\sum_{i=1}^{\bar{l}_+}\alpha_i\left[\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-K(\bar{B}_j,\bar{A}_i)+(1/t_2)K(\bar{A}_i,\bar{A}_i)\right], \\ \text{s.t.}\ & \sum_{i=1}^{\bar{l}_+}\alpha_i=1, \\ & 0\leq\alpha_i\leq\frac{c_1}{\bar{l}_+}\bar{d}_i^+, i=1,\ldots,\bar{l}_+, \end{aligned} \tag{24}$$

where $t_1=\frac{1+\frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^--2v_1}{(1-v_1)^2}$ and $t_2=\frac{\frac{2v_1}{\bar{l}_-}-\frac{4v_1^2}{\bar{l}_-}+2(\frac{v_1}{\bar{l}_-})^2\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-}{(1-v_1)^2}$.

According to (18)–(21), we obtain

$$R_+^2=\frac{1}{|\bar{I}_R^+|}\sum_{i=1}^{|\bar{I}_R^+|}\left\|\varphi(\bar{A}_i)-a_+\right\|^2, \tag{25}$$

where $\bar{I}_R^+=\left\{i|0<\alpha_i<\frac{c_1}{\bar{l}_+}\bar{d}_i^+, i=1,\ldots,\bar{l}_+\right\}$.

Similarly, we can get the simplified dual optimal problem of (14) as follows:

$$\begin{aligned} \max\ & -t_3\sum_{j_1,j_2=1}^{\bar{l}_-}\beta_{j_1}\beta_{j_2}K(\bar{B}_{j_1},\bar{B}_{j_2}) \\ & +t_4\sum_{j=1}^{\bar{l}_-}\beta_j\left[\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+K(\bar{A}_i,\bar{B}_j)+(1/t_4)K(\bar{B}_j,\bar{B}_j)\right], \\ \text{s.t.}\ & \sum_{j=1}^{\bar{l}_-}\beta_j=1, \\ & 0\leq\beta_j\leq\frac{c_2}{\bar{l}_-}\bar{d}_j^-, j=1,\ldots,\bar{l}_-, \end{aligned} \tag{26}$$

where $t_3=\frac{1+\frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+-2v_2}{(1-v_2)^2}$ and $t_4=\frac{\frac{2v_2}{\bar{l}_+}-\frac{4v_2^2}{\bar{l}_+}+2(\frac{v_2}{\bar{l}_+})^2\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+}{(1-v_2)^2}$.

Also, the center a_- and radius R_- of negative class are, respectively, calculated as follows:

$$a_-=\frac{1}{1-\frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+}\left(\sum_{j=1}^{\bar{l}_-}\beta_j\varphi(\bar{B}_j)-\frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+\varphi(\bar{A}_i)\right), \tag{27}$$

$$R_-^2=\frac{1}{|\bar{I}_R^-|}\sum_{j=1}^{|\bar{I}_R^-|}\left\|\varphi(\bar{B}_j)-a_-\right\|^2, \tag{28}$$

where $\bar{I}_R^-=\left\{j|0<\beta_j<\frac{c_2}{\bar{l}_-}\bar{d}_j^-, j=1,\ldots,\bar{l}_-\right\}$.

A new sample $x \in R^d$ is assigned to the positive class or negative class, depending on which of the two hyperspheres it lies closest to. Therefore, the decision function is defined as follows:

$$f(x) = \text{sgn}\left\{\frac{(\varphi(x)-a_+)^T(\varphi(x)-a_+)}{R_+^2} - \frac{(\varphi(x)-a_-)^T(\varphi(x)-a_-)}{R_-^2}\right\}, \tag{29}$$

where

$$\begin{aligned}
&(\varphi(x)-a_+)^{\mathrm{T}}(\varphi(x)-a_+)\\
&= K(x,x) - \frac{2}{1-\frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-}\left(\sum_{i=1}^{\bar{l}_+}\alpha_i K(\bar{A}_i,x) - \frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^- K(\bar{B}_j,x)\right) + \left(\frac{1}{1-\frac{v_1}{\bar{l}_-}\sum_{j=1}^{\bar{l}_-}\bar{d}_j^-}\right)^2\\
&\left(\sum_{i_1=1}^{\bar{l}_+}\sum_{i_2=1}^{\bar{l}_+}\alpha_{i_1}\alpha_{i_2}K(\bar{A}_{i_1},\bar{A}_{i_2}) - \frac{2v_1}{\bar{l}_-}\sum_{i=1}^{\bar{l}_+}\sum_{j=1}^{\bar{l}_-}\alpha_i\bar{d}_j^- K(\bar{A}_i,\bar{B}_j) + \left(\frac{v_1}{\bar{l}_-}\right)^2\sum_{j_1=1}^{\bar{l}_-}\sum_{j_2=1}^{\bar{l}_-}\bar{d}_{j_1}^-\bar{d}_{j_2}^- K(\bar{B}_{j_1},\bar{B}_{j_2})\right),
\end{aligned} \tag{30}$$

$$\begin{aligned}
&(\varphi(x)-a_-)^{\mathrm{T}}(\varphi(x)-a_-)\\
&= K(x,x) - \frac{2}{1-\frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i}\left(\sum_{j=1}^{\bar{l}_-}\beta_j K(\bar{B}_j,x) - \frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}d_i^+ K(\bar{A}_i,x)\right) + \left(\frac{1}{1-\frac{v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\bar{d}_i^+}\right)^2\\
&\left(\sum_{j_1=1}^{\bar{l}_-}\sum_{j_2=1}^{\bar{l}_-}\beta_{j_1}\beta_{j_2}K(\bar{B}_{j_1},\bar{B}_{j_2}) - \frac{2v_2}{\bar{l}_+}\sum_{i=1}^{\bar{l}_+}\sum_{j=1}^{\bar{l}_-}\beta_j\bar{d}_i^+ K(\bar{A}_i,\bar{B}_j) + \left(\frac{v_2}{\bar{l}_+}\right)^2\sum_{i_1=1}^{\bar{l}_+}\sum_{i_2=1}^{\bar{l}_+}\bar{d}_{i_1}^+\bar{d}_{i_2}^+ K(\bar{A}_{i_1},\bar{A}_{i_2})\right).
\end{aligned} \tag{31}$$

5 Computational complexity of LDTHSVM

In this section, we further analyze the computational complexity of our LDTHSVM. There are three main steps in Our LDTHSVM, which are

1. Calculating the local density degrees of all training samples,
2. Pruning training dataset,
3. Solving the optimal problems (24) and (26),

where the main computational cost is the calculation of k nearest neighbor of all training samples in step 1 and the solution of the optimal problems in step 3. The computational complexity of calculating k nearest neighbor of all training samples is $O(l^2 \log l)$, and the computational complexity of solving the optimal problems is $O(\bar{l}_+^3 + \bar{l}_-^3)$, where $\bar{l}_+ \ll l_+$ and $\bar{l}_- \ll l_-$, when the training dataset contains many noise samples. Therefore, the computational complexity of LDTHSVM is about $O(l^2 \log l + \bar{l}_+^3 + \bar{l}_-^3)$.

6 Experiments

In this section, we investigate classification performance of our LDTHSVM on publicly available benchmark datasets, as well as a synthetic dataset. In the experiments, we compare LDTHSVM with other classical algorithms including THSVM, WLTSVM [32], TWSVM and SVM. The parameters selection is very important for these algorithms. The exhaustive search is still the most popular method for determining the parameters [5,10,22,24,25]. To reduce computational complexity of parameters selection, we make the parameters $c_1 = c_2 = c$ and $v_1 = v_2 = v$. In each algorithm, the optimal parameter c is searched from set $\{2^i | i = 0, 1, \ldots, 10\}$, v from set $\{0.1, 0.2, \ldots, 0.9\}$ and pruning threshold σ from set $\{0.1, 0.2, \ldots, 0.5\}$ on the validation set comprising of 30% of the training samples. Once all parameters are determined, the validation sets are returned to the training datasets to construct the final classifiers.

6.1 Synthetic data with noise

In this subsection, to show the effectiveness of LDTHSVM intuitively, we use a synthetic dataset. The toy 2-D dataset is randomly generated under two Gaussian distributions: positive class: $N((0,0)^T, \text{diag}\{0.5, 0.5\})$, negative class: $N((2,2)^T, \text{diag}\{0.25, 0.25\})$. The training dataset consists of 440 samples (220 samples for each class, where there are 20 samples with wrong label) and the test dataset consists of 4000 samples (2000 samples for each class).

Figure 1 intuitively shows the classification results of the LDTHSVM, THSVM, TWSVM and SVM on the

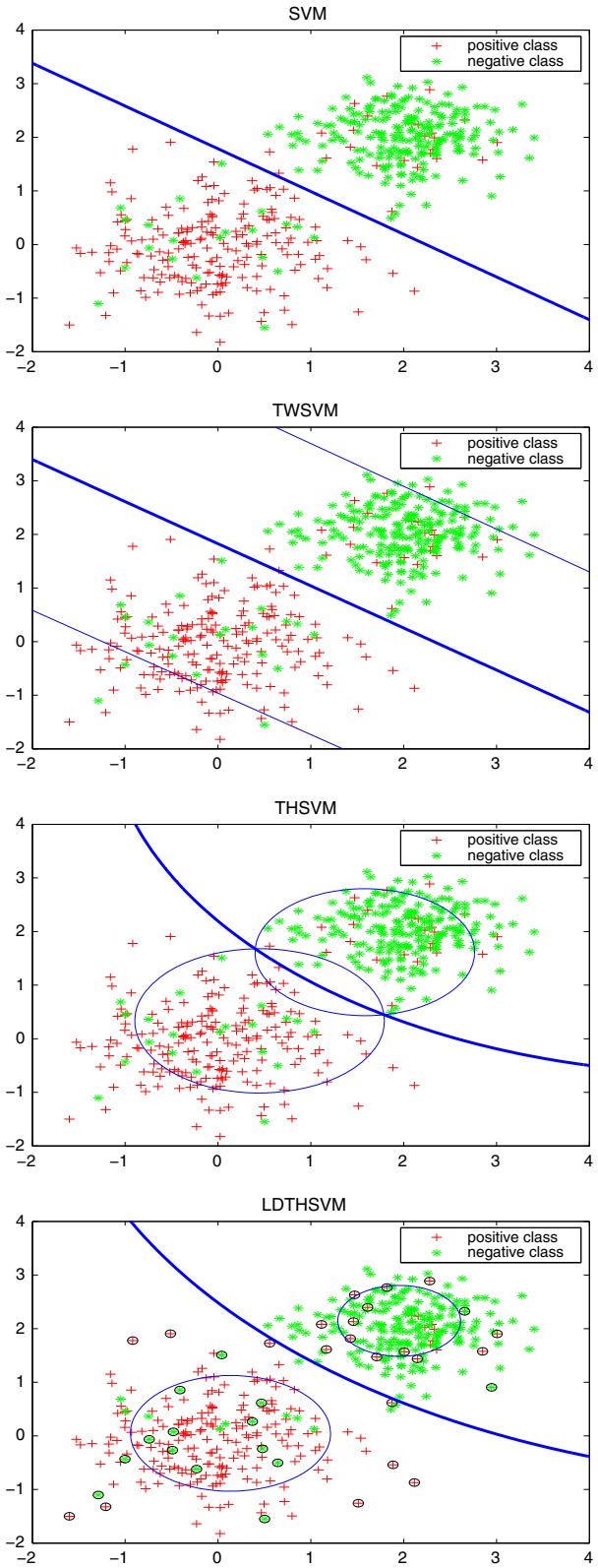

Fig. 1 Classification results of the LDTHSVM, THSVM, TWSVM and SVM for the synthetic dataset with linear kernels. The thick curves represent the separating plane, while the thin curves represent two non-parallel hyperplanes or a pair of hyperspheres. For LDTHSVM, the circled samples are pruned

Table 1 Classification performance of the LDTHSVM, THSVM, TWSVM and SVM for the synthetic dataset with linear kernels

Performance	SVM	TWSVM	THSVM	LDTHSVM
Accuracy (%)	98.75	98.80	96.07	98.95
Training time(s)	2.0343	0.292	0.4955	0.8008

For LDTHSVM, the training time still includes local density calculation

two Gaussian dataset with linear kernels. By inspecting Fig. 1, we can get the following conclusions: Firstly, the THSVM, TWSVM and SVM are significantly influenced by the noise samples, especially by the samples with wrong label. Secondly, LDTHSVM can effectively remove most samples with wrong label and a small part of isolated samples, further suppress the interference of the remaining noise samples by introducing local density of samples into the classifier, which make the separating curves around positive and negative class tighter and the centers of positive and negative hyperspheres closer to means of two Gaussian distributions. In other words, LDTHSVM can effectively depict the true distribution of the two classes of samples. Further we detailedly show the classification results in Table 1. We can observe from Table 1, LDTHSVM obtains better accuracy compared with THSVM, TWSVM and SVM. Although training speed of the LDTHSVM is slightly slower than that of THSVM and TWSVM, its training speed is significantly faster than that of SVM.

6.2 Benchmark datasets

In this subsection, to further investigate the classification performance of LDTHSVM, we perform LDTHSVM, THSVM, WLTSVM, TWSVM and SVM on the publicly available benchmark datasets from UCI Repository. In these simulations, we only consider the Gaussian kernel $K(x_1, x_2) = e^{-\gamma\|x_1-x_2\|^2}$ and the parameter γ is selected from the range $\{2^i|i = -9, -8, \ldots, 10\}$. We use the tenfold cross-validation methodology to estimate the classification accuracy of each algorithm. Table 2 lists the classification accuracies and training time of LDTHSVM, THSVM, WLTSVM, TWSVM and SVM. From Table 2, it can be observed that our LDTHSVM obtains better classification accuracies for most datasets, compared with THSVM, WLTSVM, TWSVM and SVM. This indicates that LDTHSVM can effectively suppress the interference of noise samples. Furthermore, it can be observed that, compared with THSVM, LDTHSVM is inefficient. One of the possible reasons is that there are not a large number of noise samples that can be pruned in the training dataset. Even so, the training time of LDTHSVM is also close to TWSVM.

Table 2 Classification performance of the LDTHSVM, THSVM, WLTSVM, TWSVM and SVM for the benchmark datasets with Gaussian kernels

Dataset	SVM accuracy(%) training time(s) pruned ratio(%)	TWSVM accuracy(%) training time(s) pruned ratio(%)	WLTSVM accuracy(%) training time(s) pruned ratio(%)	THSVM accuracy(%) training time(s) pruned ratio(%)	LDTHSVM accuracy(%) training time(s) pruned ratio(%)
Heart (270 × 13)	83.96 ± 6.70	79.89 ± 8.36	**84.07 ± 6.60**	82.48 ± 7.88	**84.07 ± 7.62**
	2.9071	1.8077	6.6423	0.6607	2.2648
	–	–	–	–	6.67
Ionosphere (351 × 34)	93.22 ± 4.27	86.03 ± 10.94	89.22 ± 5.15	92.29 ± 4.25	**94.06 ± 4.14**
	5.2718	6.653	6.1171	1.5239	3.4447
	–	–	–	–	18.05
Australian (690 × 14)	85.76 ± 4.47	86.30 ± 4.03	86.38 ± 3.39	83.45 ± 3.84	**86.39 ± 3.98**
	28.9792	15.254	16.4124	6.38	12.5349
	–	–	–	–	7.68
WDBC (569 × 30)	97.50 ± 1.86	96.45 ± 1.89	**97.86 ± 1.92**	95.41 ± 2.33	97.36 ± 2.29
	17.7337	7.142	7.7190	4.4224	8.2344
	–	–	–	–	3.22
Vote (435 × 16)	92.88 ± 4.06	94.32 ± 2.97	94.42 ± 3.43	92.94 ± 3.79	**94.59 ± 3.36**
	9.6824	3.3036	6.9228	2.2737	4.7007
	–	–	–	–	3.75
Breast (277 × 9)	73.28 ± 6.81	74.91 ± 6.69	**75.44 ± 7.56**	66.66 ± 9.04	72.20 ± 3.07
	2.3353	8.1016	3.6851	0.7989	1.9445
	–	–	–	–	2.48
Sonar (208 × 60)	89.26 ± 6.62	88.16 ± 7.04	89.45 ± 7.81	83.27 ± 9.43	**89.99 ± 7.49**
	1.4464	1.4147	2.5281	0.4884	1.765
	–	–	–	–	0.5

For LDTHSVM, the training time still includes local density calculation
The accuracy of bold values are the highest one in all algorithms

Table 3 Rank on classification accuracy of five classifiers for benchmark datasets

Dataset	SVM	TWSVM	WLTSVM	THSVM	LDTHSVM
Heart	2	5	3	4	1
Ionosphere	2	5	4	3	1
Australian	4	3	2	5	1
WDBC	2	4	1	5	3
Vote	5	3	2	4	1
Breast	3	2	1	5	4
Sonar	3	4	2	5	1
Average rank	3	3.71	2.14	4.42	1.71

6.3 Friedman test

From Table 2, we can notice that not any algorithm can outperform all others for all datasets in the light of classification accuracy. In this subsection, to analyze the classification performance of five algorithms on multiple datasets statistically, we use Friedman test [1,4]. The ranks of five classifiers on classification accuracy for all datasets are listed in Table 3. We can calculate the Friedman statistic according to (32)

$$\chi_F^2 = \frac{12q}{p(p+1)}\left[\sum_{i=1}^{p} R_i^2 - \frac{p(p+1)^2}{4}\right], \tag{32}$$

where $R_i = \frac{1}{q}\sum_{j=1}^{q} r_i^j$ and r_i^j represents the rank of the ith of p classifiers on the jth of q datasets. Friedmans χ_F^2 is undesirably conservative, and we use the other better statistic

$$F_F = \frac{(q-1)\chi_F^2}{q(p-1)-\chi_F^2}, \tag{33}$$

which is distributed according to the $F(p-1, (p-1)(q-1))$.

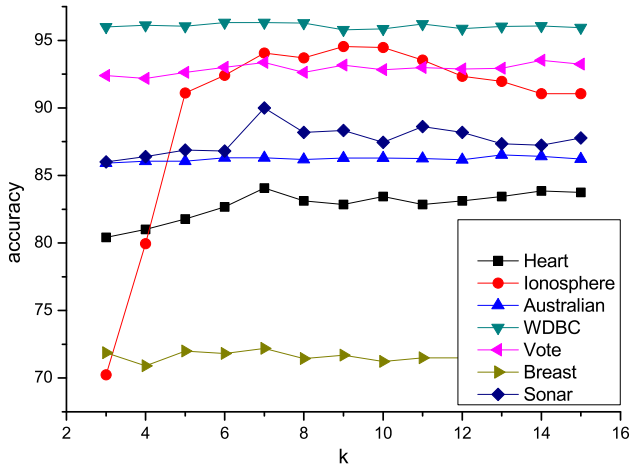

Fig. 2 Classification accuracy for different parameter k on different datasets

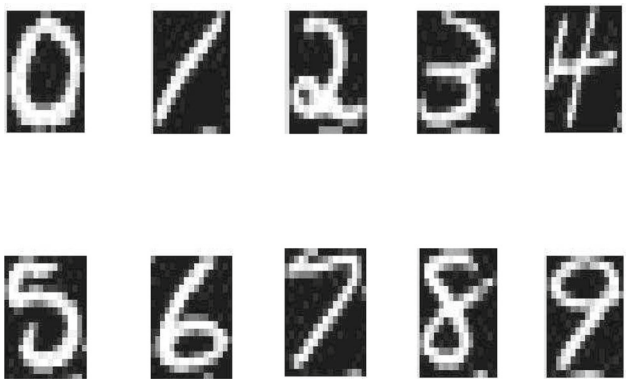

Fig. 3 Illustration of the handwritten digits

We can get $\chi_F^2 = 14.95$ and $F_F = 6.87$ according to (32) and (33), where F_F is distributed according to F(4,24). The critical value of $F(4,24)$ is 2.19 for the level of significance $\alpha = 0.1$. Similarly, it is 2.78 for the level of significance $\alpha = 0.05$. Since the critical value is smaller than F_F, there is significant difference among five classifiers. It can be observed from Table 3 that the average rank of LDTHSVM is lower than the other classifiers. It implies that the LDTHSVM has better accuracy than the other classifiers.

6.4 Analysis on the parameter *k*

In this subsection, we further analyze the effect of parameter k on classification performance. In Fig. 2, we show the classification accuracies for different parameter k on different datasets. From Fig. 2, we can observe that, for smaller k, it is hard to get the best results, the main reason is that many useful neighbors are lost, when calculating the local density degrees of training samples, for larger k, it cannot be guaranteed to get the best classification accuracy, the main reason is that many noise samples could be introduced. Through a large number of experiments and observation, we notice that $k = 7$ can obtain satisfactory performance in general.

6.5 Handwritten digits recognition

We use LDTHSVM to recognize handwritten digits in this subsection. The USPS dataset, which is a publicly available database of handwritten digits recognition, is used to evaluate our LDTHSVM. In USPS dataset, there are 11000 8-bit grayscale images of handwritten digits, and each handwritten digit has 1100 images, as shown in Fig. 3.

The classification results of four classifiers are shown in Table 4. From Table 4, we can learn that our LDTHSVM has better classification accuracy, compared with the SVM, TWSVM and THSVM.

7 Conclusions

In this paper, the improvements for THSVM, called LDTHSVM classifier, have been presented. The proposed LDTHSVM inherits good properties from THSVM. For instance, LDTHSVM solves a pair of smaller sized optimiza-

Table 4 Classification result of the LDTHSVM, THSVM, TWSVM and SVM on USPS dataset with linear kernel

Dataset	SVM accuracy(%) training time(s)	TWSVM accuracy(%) training time(s)	THSVM accuracy(%) training time(s)	LDTHSVM accuracy(%) training time(s)
2 versus 5	99.55 ± 0.59	99.62 ± 0.57	99.69 ± 0.49	$\mathbf{99.75 \pm 0.47}$
	44.5965	38.5449	41.8993	85.7952
1 versus 9	98.42 ± 1.18	98.05 ± 1.29	97.65 ± 1.45	$\mathbf{98.51 \pm 1.16}$
	54.7776	32.0918	39.4862	85.4703
3 versus 6	99.49 ± 0.64	99.21 ± 0.89	99.27 ± 0.80	$\mathbf{99.56 \pm 0.70}$
	47.4127	36.9042	46.4143	92.4035
5 versus 8	99.82 ± 0.38	99.52 ± 0.62	99.86 ± 0.34	$\mathbf{99.89 \pm 0.43}$
	65.4818	62.5315	64.3012	105.7868

The accuracy of bold values are the highest one in all algorithms

tion problems, avoids the inversion matrix in its dual QPPs and directly uses kernel trick to solve nonlinear problems as in the SVM. Further, unlike THSVM, LDTHSVM prunes training dataset according to local density degrees of training samples and introduce local density degrees into THSVM and reconstruct classification model with the pruned training dataset. The classification results on synthetic and publicly available benchmark datasets have shown that LDTHSVM can obtain better classification performance, compared with THSVM, WLTSVM, TWSVM and SVM, especially for the large-scale datasets which include many noise samples.

Acknowledgements The authors thank the editors and anonymous referees for helpful comments and suggestions that have led to improvement of the paper. This work is supported by the Natural Science Foundation of Liaoning province in China (No. 201601291), the Education Committee Project of Liaoning province in China (No. L2012089, No. 2016TSPY13).

References

1. Ar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**(1), 1–30 (2006)
2. Chen, W.J., Shao, Y.H., Li, C.N., Deng, N.Y.: MLTSVM: a novel twin support vector machine to multi-label learning. Pattern Recogn. **52**, 61–74 (2015)
3. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. J. Mach. Learn. Res. **6**(4), 1889–1918 (2005)
4. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf. Sci. **180**(10), 2044–2064 (2010)
5. Khemchandani, R.J., Chandra, R.S.: Twin support vector machines for pattern classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 905–910 (2007)
6. Joachims, T.: Making large-scale support vector machine learning practical. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods-Support Vector Learning, pp. 169–184. MIT Press, Cambridge (1999)
7. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Trans. Neural Netw. **11**(1), 124–136 (2000)
8. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to platt's SMO algorithm for SVM classifier design. Neural Comput. **13**(3), 637–649 (2001)
9. Kumar, M.A., Gopal, M.: Application of smoothing technique on twin support vector machines. Pattern Recogn. Lett. **29**(13), 1842–1848 (2008)
10. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Syst. Appl. **36**(4), 7535–7543 (2009)
11. Lee, K.Y., Kim, D.W., Lee, K.H., Lee, D.: Density-induced support vector data description. IEEE Trans. Neural Netw. **18**(1), 284–289 (2007)
12. Li, Z., Tian, Y., Li, K., Zhou, F., Yang, W.: Reject inference in credit scoring using semi-supervised support vector machines. Expert Syst. Appl. **74**, 105–114 (2017)
13. Liang, X., Zhu, L., Huang, D.S.: Multi-task ranking svm for image cosegmentation. Neurocomputing **247**, 126–136 (2017)
14. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 69–74 (2006)
15. Mavroforakis, M.E., Theodoridis, S.: A geometric approach to support vector machine (SVM) classification. IEEE Trans. Neural Netw. **17**(3), 671–682 (2006)
16. Osuna, E., Freund, R., Girosi, F.: An improved training algorithm for support vector machines. Neural Netw. Signal Process. **17**(5), 276–285 (1998)
17. Peng, J., Rafferty, K., Ferguson, S.: A fast algorithm for sparse support vector machines for mobile computing applications. Neurocomputing **230**, 160–172 (2017)
18. Peng, X.: A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. Inf. Sci. **180**(20), 3863–3875 (2010)
19. Peng, X.: TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition. Pattern Recogn. **44**(10–11), 2678–2692 (2011)
20. Peng, X., Xu, D.: Twin Mahalanobis distance-based support vector machines for pattern recognition. Inf. Sci. **200**(1), 2237 (2012)
21. Peng, X., Xu, D.: Bi-density twin support vector machines for pattern recognition. Neurocomputing **99**(1), 134–143 (2013)
22. Peng, X., Xu, D.: A twin-hypersphere support vector machine classifier and the fast learning algorithm. Inf. Sci. **221**(1), 12–27 (2013)
23. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods-Support Vector Learning, pp. 185–208. MIT Press, Cambridge (1999)
24. Qi, Z., Tian, Y., Shi, Y.: Structural twin support vector machine for classification. Knowl.-Based Syst. **43**(2), 74–81 (2013)
25. Shao, Y.H., Zhang, C.H., Wang, X.B., Deng, N.Y.: Improvements on twin support vector machines. IEEE Trans. Neural Netw. **22**(6), 962–968 (2011)
26. Tian, Y., Qi, Z., Ju, X., Shi, Y., Liu, X.: Nonparallel support vector machines for pattern classification. IEEE Trans. Cybern. **44**(7), 1067–1079 (2014)
27. Vapnik, V.: Statistical Learning Theory. Cambridge University Press, New York (1998)
28. Wang, Z., Shao, Y.H., Bai, L., Deng, N.Y.: Twin support vector machine for clustering. IEEE Trans. Neural Netw. Learn. Syst. **26**(10), 2583–2588 (2015)
29. Xie, L., Li, G., Xiao, M., Peng, L., Chen, Q.: Hyperspectral image classification using discrete space model and support vector machines. IEEE Geosci. Remote Sens. Lett. **14**(03), 374–378 (2017)
30. Xu, Y., Guo, R.: A twin hyper-sphere multi-class classification support vector machine. J. Intell. Fuzzy Syst. Appl. Eng. Technol. **27**(4), 1783–1790 (2014)
31. Xu, Y., Yang, Z., Zhang, Y., Pan, X., Wang, L.: A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification. Knowl.-Based Syst. **95**, 75–85 (2016)
32. Ye, Q., Zhao, C., Gao, S., Zheng, H.: Weighted twin support vector machines with local information and its application. Neural Netw. **35**(11), 31–39 (2012)
33. Ye, Q., Zhao, C., Ye, N., Chen, X.: Localized twin SVM via convex minimization. Neurocomputing **74**(4), 580–587 (2011)
34. Zhang, X.G.: Introduction to statistical learning theory and support vector machines. Acta Autom. Sin. **26**(01), 32–42 (2000)
35. Zhu, F., Wei, J.: Localization algorithm in wireless sensor networks based on improved support vector machine. J. Nanoelectron. Optoelectron. **12**(05), 452–459 (2017)