Pattern Recognition 74 (2018) 434–447

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Least squares twin bounded support vector machines based on L1-norm distance metric for classification

He Yan[a], Qiaolin Ye[a,d,e,*], Tian'an Zhang[a,b], Dong-Jun Yu[c], Xia Yuan[c], Yiqing Xu[a], Liyong Fu[f]

[a] College of Information Science and Technology, Nanjing Forestry University, No.159 Longpan Road, Nanjing, 210037, PR China
[b] Collaborative Innovation Center of Sustainable Forestry in Southern China of Jiangsu Province, Nanjing Forestry University, Nanjing, 210037, PR China
[c] School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing, 210094, PR China
[d] Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing, 210094, PR China
[e] The Laboratory for Internet of Things and Mobile Internet Technology of Jiangsu Province, Huaiyin Institute of Technology, Huai'an, 223003, PR China
[f] Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing, 100091, PR China

ARTICLE INFO

Article history:
Received 26 September 2016
Revised 19 September 2017
Accepted 23 September 2017
Available online 25 September 2017

Keywords:
L1-LSTBSVM
TBSVM
L1-norm distance
Outliers

ABSTRACT

In this paper, we construct a least squares version of the recently proposed twin bounded support vector machine (TBSVM) for binary classification. As a valid classification tool, TBSVM attempts to seek two non-parallel planes that can be produced by solving a pair of quadratic programming problems (QPPs), but this is time-consuming. Here, we solve two systems of linear equations rather than two QPPs to avoid this deficiency. Furthermore, the distance in least squares TBSVM (LSTBSVM) is measured by L2-norm, but L1-norm distance is usually regarded as an alternative to L2-norm to improve model robustness in the presence of outliers. Inspired by the advantages of least squares twin support vector machine (LSTWSVM), TBSVM and L1-norm distance, we propose a LSTBSVM based on L1-norm distance metric for binary classification, termed as L1-LSTBSVM, which is specially designed for suppressing the negative effect of outliers and improving computational efficiency in large datasets. Then, we design a powerful iterative algorithm to solve the L1-norm optimal problems, and it is easy to implement and its convergence to an optimum solution is theoretically ensured. Finally, the feasibility and effectiveness of L1-LSTBSVM are validated by extensive experimental results on both UCI datasets and artificial datasets.

1. Introduction

Support Vector Machines (SVMs) [1–3] have become powerful computing tools for supervised learning, which are widely used in data classification and regression analysis. SVM is based on the VC dimension and the structural risk minimization principle of statistical learning theory. The central idea of SVM is to search an optimal separating plane by maximizing the margin between two parallel support planes [4]. However, there are two main challenges in the original SVM: 1) seeking this plane needs to solve a constrained and complex quadratic programming problem (QPP), which leads to higher computational costs; 2) exclusive OR (XOR) problem cannot be handled smoothly [5], because when the linear independence condition is not satisfied, a single line classifier plane for the XOR problem may misclassify some points, which leads to bad performance of low correctness on XOR datasets, compared to nonparallel planes.

To address these, many improved algorithms have been proposed, e.g. SMO [6], SVMLight [7], Libsvm [8] and proximal support vector machine (PSVM) [9]. Specially, Mangasarian and Wild proposed a proximal support vector machine via generalized eigenvalues (GEPSVM) [5] for binary classification problem. The geometric meaning of GEPSVM shows that each plane is the closest to the samples of its own class and meanwhile the furthest from the samples of other classes [5]. Moreover, GEPSVM tries to find a pair of non-parallel planes by solving two generalized eigenvalue problems to replace complex QPPs, which lowers computational complexity and is superior in handling the XOR problem. The advantages of GEPSVM bring great impact to its multifarious improvement [10,11]. Jayadeva et al. proposed a twin support vector machine (TWSVM) [12] based on the idea of GEPSVM, which solves two QPPs (the scale is relatively small compared to that of SVM) instead of generalized eigenvalue problems [13], so the computational time of TWSVM is only 1/4 of the standard SVM [12]. However, it takes a long time to calculate two QPPs when the sample is large enough.

Currently, many researchers have improved TWSVM in various aspects. Shao [14] optimized TWSVM by introducing the regu-

* Corresponding author.
E-mail addresses: yqlcom@njfu.edu.cn, yeqiaolin65620868@163.com (Q. Ye).

https://doi.org/10.1016/j.patcog.2017.09.035
0031-3203/

larization terms, and proposed a twin bounded support vector machine (TBSVM). In TWSVM the empirical risk is minimized, while in TBSVM the structural risk minimization is achieved. This modification improves the classification performance. Kumar et al. [15] optimized TWSVM by taking advantage of constraints in the form of equalities instead of inequalities to modify the primal QPPs in least squares sense, and proposed a least squares version of TWSVM (LSTWSVM). The solutions of LSTWSVM follow directly from solving two linear equations as opposed to solving two QPPs. Thus, LSTWSVM effectively handles large samples without any external optimization, and meanwhile, the computational time is much less than that of TWSVM. Later, Tomar et al. [16] extended LSTWSVM to multiclass classification, and proposed multiclass least squares twin support vector machine (MLSTSVM). For multiple samples, MLSTSVM generates multiple planes, with one plane corresponding to one sample, that is, the *i*th sample being as close as possible to the *i*th plane, and simultaneously, away from other planes as far as possible. Ye [17] proposed a new classification method named weighted TWSVM with local information (WLTSVM), which makes use of the intra-class graph to show the compactness of intra-class samples and the inter-class graph to represent the dispersion of inter-class samples. The similarity between the samples taken into consideration, WLTSVM has a fine generalization ability. Zhang [18] reformulated the optimization problems of TWSVM by utilizing an effective fast algorithm (called Newton-Armijo algorithm) to solve a pair of unconstraint differentiable optimization problems rather than two dual QPPs, and proposed a smoothed projective twin support vector machine (SPTSVM).

However, it should be noted that TWSVM and its variants are sensitive to outliers, because the L2-norm distance easily exaggerates the effect of outliers by the square operation [19], which lowers the classification performance. To improve model robustness in the presence of the outliers, L1-norm distance is applied in the algorithms [20–25], and in fact the use of L1-norm distance is often considered as an efficient way to reduce the impact of outliers [22,26]. Li [20] proposed a non-parallel proximal support vector machine via L1-norm distance (L1-NPSVM). To solve the formulated objective, a gradient ascending (GA) iterative algorithm is proposed, and it is simple to perform but may not guarantee the optimality of the solution due to both the need of introduction of a non-convex surrogate function and the difficult selection of step-size. Gao [27] developed a new non-parallel plane classifier, which can automatically select the relevant features. The feature selection was performed by applying L1-norm to the solution. To obtain the strong suppression ability of features, the exterior penalty theorem is applied, where the zero or nonzero solution is obtained by a non-exact Newton algorithm. Furthermore, the exterior penalty theorem can be applied only when the modeling shares some special formulations, otherwise, the exact solution cannot be obtained theoretically. In addition, as TBSVM needs to solve two QPPs, it is hard to deal with large datasets availably.

In this study, we have enhanced TBSVM to least squares TBSVM (LSTBSVM) using the idea of proximal support vector machine (PSVM) [9]. We modify the primal QPPs of TBSVM in least squares sense and settle them with constraints in the form of equalities to replace inequalities of TBSVM [15]. We obtain two non-parallel planes by solving two systems of linear equations as opposed to solving two QPPs in TBSVM, so LSTBSVM can easily handle large datasets without any additional optimizers. In addition, in order to improve the robustness of LSTBSVM, L1-norm distance is applied in this algorithm. Thus, we construct a strong LSTBSVM model based on L1-norm distance metric for binary classification problem, termed as L1-LSTBSVM. In L1-LSTBSVM, the constraints in the form of inequalities of TBSVM [15] are replaced with constraints in the form of equalities, which costs comparably less computational time in large datasets. The goal of L1-LSTBSVM is to minimize the L1-norm intra-class distance dispersion, and maximize the L1-norm inter-class distance dispersion simultaneously. To sum up, L1-LSTBSVM owns the following four advantages.

1. The distance in L1-LSTBSVM is measured by L1-norm. To solve the L1-norm distance optimal problems, we implement a simple but efficient iterative algorithm, which is easy to actualize and its convergence to a reasonable optimum solution is theoretically ensured. Therefore, we can obtain two non-parallel optimal planes.
2. The application of L1-norm distance makes L1-LSTBSVM more robust in the presence of outliers than L2-norm distance, and it can availably control the impact of the outliers even if outliers take a great proportion.
3. Extensive experimental results on both UCI datasets and artificial datasets confirm that, compared with four algorithms (TBSVM [14], TWSVM [12], LSTWSVM [15] and L1-NPSVM [20]), L1-LSTBSVM surpasses them in classification accuracy and lowers computational time, which indicates L1-norm distance improves the generalization ability of L1-LSTBSVM.
4. Last but not the least, it is worth pointing out that the method which we proposed can be conveniently extended to solve other improved methods of TWSVM [14]. We are planning to research them in the future.

The paper is organized as follows. The basic principle of TWSVM and TBSVM is introduced in Section 2. L1-LSTBSVM is proposed with its feasibility and theoretical analysis in Section 3. All the experimental results are shown in Section 4 and conclusions are given in Section 5.

2. Related works

In this section, all vectors are column vectors unless transformed to row vectors by a prime superscript T. The vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ of appropriate dimension are represented by identity column vectors. Besides, $\mathbf{I}$ denotes an identity matrix of appropriate dimension. We consider a binary classification problem in the n dimensional real space R^n, and the set of training sample is indicated by $T = \{(\mathbf{x}_j^{(i)}, y_i) | i = 1, 2,\ j = 1, 2, \ldots, m_i\}$, where $\mathbf{x}_j^{(i)} \in R^n$ and $y_i \in \{-1,\ 1\}$, $\mathbf{x}_j^{(i)}$ denotes the ith class and jth sample. We suppose that matrix $\mathbf{A} = [\mathbf{A}_1{}^{(1)}, \mathbf{A}_2{}^{(1)}, \ldots, \mathbf{A}_{m_1}{}^{(1)}]^T$ with size of $m_1 \times n$ represents the data points of class 1 (class +1), while matrix $\mathbf{B} = [\mathbf{B}_1{}^{(2)}, \mathbf{B}_2{}^{(2)}, \ldots, \mathbf{B}_{m_2}{}^{(2)}]^T$ with size of $m_2 \times n$ represents the data points of class 2 (class -1). Matrices $\mathbf{A}$ and $\mathbf{B}$ represent all the training data points, where $m_1 + m_2 = m$. m_1 represents the number of positive class samples while m_2 represents the number of negative class samples. In the following, we review two well-known non-parallel proximal classifiers: TWSVM [12] and TBSVM [14].

2.1. TWSVM

TWSVM is an excellent classifier for binary classification problem, which solves two QPPs (the scale is relatively small compared to that of SVM) to obtain two non-parallel planes. The primary target of TWSVM is to seek a pair of non-parallel planes:

$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0, \quad \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{1}$$

where $\mathbf{w}_1,\ \mathbf{w}_2 \in R^n,\ b_1,\ b_2 \in R$. The optimization goal of TWSVM is that each plane is as close as possible to one of the two classes and away from the other class as far as possible [12].

This produces the following two objective problems of TWSVM:

$$\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \ \frac{1}{2}||\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1||_2^2+c_1\mathbf{e}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2\geq\mathbf{e}_2, \quad \mathbf{q}_2\geq 0 \tag{2}$$

$$\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \ \frac{1}{2}||\mathbf{B}\mathbf{w}_2+\mathbf{e}_2b_2||_2^2+c_2\mathbf{e}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}_2+\mathbf{e}_1b_2)+\mathbf{q}_1\geq\mathbf{e}_1, \quad \mathbf{q}_1\geq 0 \tag{3}$$

where $||\cdot||_2$ denotes the L2-norm, $\mathbf{q}_1$ and $\mathbf{q}_2$ are slack vectors, c_1 and c_2 are two nonnegative penalty coefficients, which are the balance factors of positive and negative samples respectively, and can overcome the problem of sample imbalance in TWSVM. It is worth noting that the distance in TWSVM is measured by L2-norm, which is easy to exaggerate the effect of outliers by the square operation [19].

In order to acquire the corresponding Wolfe dual problems of (2) and (3), we suppose that matrices $\mathbf{H}^T\mathbf{H}$ and $\mathbf{G}^T\mathbf{G}$ are nonsingular, where $\mathbf{H}=[\mathbf{A}\ \ \mathbf{e}_1]$ and $\mathbf{G}=[\mathbf{B}\ \ \mathbf{e}_2]$. Thus, the dual problems are shown as following:

$$\max_{\boldsymbol{\alpha}} \ \mathbf{e}_2^T\boldsymbol{\alpha}-\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}$$
$$\text{s.t.} \quad 0\leq\boldsymbol{\alpha}\leq c_1\mathbf{e}_2 \tag{4}$$

$$\max_{\boldsymbol{\beta}} \ \mathbf{e}_1^T\boldsymbol{\beta}-\frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta}$$
$$\text{s.t.} \quad 0\leq\boldsymbol{\beta}\leq c_2\mathbf{e}_1 \tag{5}$$

where $\boldsymbol{\alpha}\in R^{m_2}, \boldsymbol{\beta}\in R^{m_2}$ are Lagrange multipliers. It should be pointed out that, inverse matrices $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}$ and $\left(\mathbf{G}^T\mathbf{G}\right)^{-1}$ are easy to encounter singular problem. Therefore, regularization term $\varepsilon\mathbf{I}$ is introduced to solve these [5,12], because $\left(\mathbf{H}^T\mathbf{H}+\varepsilon\mathbf{I}\right)^{-1}$ and $\left(\mathbf{G}^T\mathbf{G}+\varepsilon\mathbf{I}\right)^{-1}$ satisfy positive definiteness, which do not suffer from the singular problem. It is equivalent to adding scaled L2-norm of hyperplane weight vector to original objective function, where ε is a positive scalar and small enough to keep the structure of data. So the dual problems are amended artificially as

$$\max_{\boldsymbol{\alpha}} \ \mathbf{e}_2^T\boldsymbol{\alpha}-\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{H}+\varepsilon\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}$$
$$\text{s.t.} \quad 0\leq\boldsymbol{\alpha}\leq c_1\mathbf{e}_2 \tag{6}$$

$$\max_{\boldsymbol{\beta}} \ \mathbf{e}_1^T\boldsymbol{\beta}-\frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{G}+\varepsilon\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta}$$
$$\text{s.t.} \quad 0\leq\boldsymbol{\beta}\leq c_2\mathbf{e}_1 \tag{7}$$

Two non-parallel proximal planes can be obtained by using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to solve Eq. (8).

$$\mathbf{z}_1=[\mathbf{w}_1\ \ b_1]^T=-\left(\mathbf{H}^T\mathbf{H}+\varepsilon\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}$$
$$\mathbf{z}_2=[\mathbf{w}_2\ \ b_2]^T=\left(\mathbf{G}^T\mathbf{G}+\varepsilon\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \tag{8}$$

Further, weight vectors $\mathbf{w}_1$, $\mathbf{w}_2$ and deviations b_1, b_2 can be obtained. Hence, we can get two non-parallel proximal planes. A new data point $\mathbf{x}$ is assigned to class 1 or class 2 depending on its proximity to the two non-parallel planes.

2.2. TBSVM

Similar to TWSVM, TBSVM also needs to solve two smaller QPPs to gain two non-parallel planes. But, there are some obvious differences: 1) in TWSVM, the empirical risk is minimized, whereas in TBSVM, to minimize the structural risk, a regularization term is added by the idea of maximizing some margin; 2) the dual problems of TBSVM can be derived without any extra assumption, which is stricter and more complete than TWSVM. The primal problems of TBSVM are expressed as:

$$\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \ \frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)+\frac{1}{2}||\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1||_2^2+c_1\mathbf{e}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2\geq\mathbf{e}_2, \quad \mathbf{q}_2\geq 0 \tag{9}$$

$$\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \ \frac{1}{2}c_4\left(\|\mathbf{w}_2\|_2^2+b_2^2\right)+\frac{1}{2}||\mathbf{B}\mathbf{w}_2+\mathbf{e}_2b_2||_2^2+c_2\mathbf{e}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}_2+\mathbf{e}_1b_2)+\mathbf{q}_1\geq\mathbf{e}_1, \quad \mathbf{q}_1\geq 0 \tag{10}$$

where c_1, c_2, c_3 and c_4 are positive parameters, $||\cdot||_2$ denotes the L2-norm, $(1/2)c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)$ is an additional regularization term, which makes the structural risk minimized in formula (9) [14].

To obtain two non-parallel proximal planes, we need to derive the dual problems of formulas (9) and (10). The Lagrange of the formula (9) is built, as shown in the following:

$$L\left(\mathbf{w}_1,b_1,\mathbf{q}_2,\mathbf{a},\boldsymbol{\beta}\right)=\frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)+\frac{1}{2}||\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1||_2^2+c_1\mathbf{e}_2^T\mathbf{q}_2$$
$$-\boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2-\mathbf{e}_2)-\boldsymbol{\beta}^T\mathbf{q}_2 \tag{11}$$

where $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\alpha_3\ldots\alpha_{m_2})^T$, $\boldsymbol{\beta}=(\beta_1,\beta_2,\beta_3\ldots\beta_{m_2})^T$ are Lagrange multipliers. The derivatives of $\mathbf{w}_1$, b_1 and $\mathbf{q}_2$ are solved by Lagrange function L separately, and their derivatives are set to be zero. Thus, the Karush-Kuhn-Tucker (KKT) conditions can be obtained by these derivatives, as shown in the following:

$$\frac{\partial L}{\partial\mathbf{w}_1}=c_3\mathbf{w}_1+\mathbf{A}^T(\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1)+\mathbf{B}^T\boldsymbol{\alpha}=0, \tag{12}$$

$$\frac{\partial L}{\partial b_1}=c_3b_1+\mathbf{e}_1^T(\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1)+\mathbf{e}_2^T\boldsymbol{\alpha}=0, \tag{13}$$

$$\frac{\partial L}{\partial\mathbf{q}_2}=c_1\mathbf{e}_2-\boldsymbol{\alpha}-\boldsymbol{\beta}=0, \tag{14}$$

$$-(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2\geq\mathbf{e}_2, \quad \mathbf{q}_2\geq 0, \tag{15}$$

$$\boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2-\mathbf{e}_2)=0, \quad \boldsymbol{\beta}^T\mathbf{q}_2=0, \tag{16}$$

$$\boldsymbol{\alpha}\geq 0, \quad \boldsymbol{\beta}\geq 0 \tag{17}$$

As $\boldsymbol{\beta}\geq 0$, from (14) we get

$$0\leq\boldsymbol{\alpha}\leq c_1\mathbf{e}_2 \tag{18}$$

Next, Eqs. (12) and (13) are combined to be:

$$\left([\mathbf{A}\ \ \mathbf{e}_1]^T[\mathbf{A}\ \ \mathbf{e}_1]+c_3\mathbf{I}\right)[\mathbf{w}_1\ \ b_1]^T+[\mathbf{B}\ \ \mathbf{e}_2]^T\boldsymbol{\alpha}=0 \tag{19}$$

We have defined $\mathbf{H}=[\mathbf{A}\ \ \mathbf{e}_1]$, $\mathbf{G}=[\mathbf{B}\ \ \mathbf{e}_2]$ and $\mathbf{z}_1=(\mathbf{w}_1\ b_1)^T$, $\mathbf{z}_2=(\mathbf{w}_2\ b_2)^T$, with these notations, Eq. (19) can be rewritten as

$$\left(\mathbf{H}^T\mathbf{H}+c_3\mathbf{I}\right)\mathbf{z}_1+\mathbf{G}^T\boldsymbol{\alpha}=0 \tag{20}$$

Eq. (20) is equivalent to

$$\mathbf{z}_1=-\left(\mathbf{H}^T\mathbf{H}+c_3\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \tag{21}$$

Put Eq. (21) into the Lagrange and use the K.K.T conditions above, we get the dual problem of formula (9) as below:

$$\max_{\boldsymbol{\alpha}} \ \mathbf{e}_2^T\boldsymbol{\alpha}-\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{H}+c_3\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}$$
$$\text{s.t.} \quad 0\leq\boldsymbol{\alpha}\leq c_1\mathbf{e}_2 \tag{22}$$

It seems the same when the parameter c_3 in (22) is replaced by ε in (6), but, there is essential difference in the sense. The parameter ε is only a smaller scalar, while c_3 is a weighting factor which determines the weigh between the empirical risk and the regularization term, so it is very important to choose a proper parameter c_3. $(1/2)c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)$ embodies the structure risk minimization principle and improves the classification accuracy.

Similarly, we consider formula (10) and obtain its dual problem as

$$\max_{\boldsymbol{\beta}} \ \mathbf{e}_1^T\boldsymbol{\beta}-\frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{G}+c_4\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta}$$
$$\text{s.t.} \quad 0\le\boldsymbol{\beta}\le c_2\mathbf{e}_1 \tag{23}$$

The augmented vector $\mathbf{z}_2$ is given by

$$\mathbf{z}_2=\left(\mathbf{G}^T\mathbf{G}+c_4\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \tag{24}$$

Once the solutions $\mathbf{z}_1$ and $\mathbf{z}_2$ of the formulas (9) and (10) are obtained, we can obtain two non-parallel proximal planes.

$$\mathbf{x}^T\mathbf{w}_1+b_1=0, \quad \mathbf{x}^T\mathbf{w}_2+b_2=0 \tag{25}$$

The distance from the plane $(\mathbf{x}^T\mathbf{w}_1+b_1=0)$ is smaller to the positive sample than to the negative sample, while the distance from the plane $(\mathbf{x}^T\mathbf{w}_2+b_2=0)$ is smaller to the negative sample than to the positive sample.

3. Least squares TBSVM based on L1-norm distance

The advantages of TBSVM are remarkable, whereas we should not ignore the shortcomings: 1) formulas (9) and (10) show that L2-norm distance criterion exaggerates the effect of outliers by the square operation; 2) as TBSVM needs to solve two QPPs, it is hard to handle large datasets availably. To alleviate these problems, we alter the primal QPPs of TBSVM in least squares sense and solve them with constraints in the form of equalities to replace inequalities of TBSVM [15], which costs comparably less computational time. Besides, the application of L1-norm distance is often considered as a simple and effective way to reduce the impact of outliers, and can improve the generalization ability and flexibility of the model. Two primal problems of L1-LSTBSVM are shown as followed:

$$\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)+||\mathbf{A}\mathbf{w}_1+\mathbf{e}_1b_1||_1+\frac{1}{2}c_1\mathbf{q}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2=\mathbf{e}_2, \tag{26}$$

$$\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \frac{1}{2}c_4\left(\|\mathbf{w}_2\|_2^2+b_2^2\right)+||\mathbf{B}\mathbf{w}_2+\mathbf{e}_2b_2||_1+\frac{1}{2}c_2\mathbf{q}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}_2+\mathbf{e}_1b_2)+\mathbf{q}_1=\mathbf{e}_1, \tag{27}$$

where $||\cdot||_1$ denotes the L1-norm, note that formula (26) uses the slack variable $\mathbf{q}_2^T\mathbf{q}_2$ and penalty coefficient $c_1/2$ instead of $\mathbf{e}_2^T\mathbf{q}_2$ and c_1 used in formula (9) separately. That is, (26) utilizes the square of L2-norm distance of slack variable $\mathbf{q}_2$ with $c_1/2$ replacing $\mathbf{q}_2$ with c_1 as used in (9), which makes the constraint $\mathbf{q}_2\ge 0$ redundant [28]. $\mathbf{e}_2^T\mathbf{q}_2$ is used to measure the error wherever the optimal plane is closer than this minimum distance of 1, and it minimizes the sum of slack variables, hence attempts to minimize misclassification due to points belonging to class 2. A classic improvement of SVM is the least squares SVM, followed by the least squares version of TWSVM. In view of this, we extend TBSVM to least squares TBSVM. This simple modification allows us to rewrite the solution of (26) as a solution of linear equation. Under the minimum value of objective problems, each plane is required to be as close as possible to one of the two classes and as far as possible from the other class [12].

The primal problems of formulas (26) and (27) can be optimized in the following form (which is further discussed in the last paragraph of this section):

$$\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)+\sum_{i=1}^{m_1}\frac{\left(\mathbf{A}_i\mathbf{w}_1+\mathbf{e}_1^ib_1\right)^2}{\mathbf{D}_i}+\frac{1}{2}c_1\mathbf{q}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2=\mathbf{e}_2, \tag{28}$$

$$\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \frac{1}{2}c_4\left(\|\mathbf{w}_2\|_2^2+b_2^2\right)+\sum_{j=1}^{m_2}\frac{\left(\mathbf{B}_j\mathbf{w}_2+\mathbf{e}_2^jb_2\right)^2}{\mathbf{D}_j}+\frac{1}{2}c_2\mathbf{q}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}_2+\mathbf{e}_1b_2)+\mathbf{q}_1=\mathbf{e}_1, \tag{29}$$

where $\mathbf{D}_i=2\left|\mathbf{A}_i\mathbf{w}_1+\mathbf{e}_1^ib_1\right|\ne 0$, $\mathbf{D}_j=2\left|\mathbf{B}_j\mathbf{w}_2+\mathbf{e}_2^jb_2\right|\ne 0$, and $\mathbf{z}_1=(\mathbf{w}_1\ b_1)^T$, $\mathbf{z}_2=(\mathbf{w}_2\ b_2)^T$. It is difficult to solve formulas (28) and (29), because they contain absolute value operation. To solve these, we propose an iterative convex optimization strategy. The basic idea of this method is to iteratively update the vector $\mathbf{z}_1$ until its objective function value converges to a fixed value. Assume $\mathbf{z}_1^{(p)}$ is the solution for the iteration of p. Then, the optimal solution of $\mathbf{z}_1^{(p+1)}$ for the iteration of $p+1$ is defined as the solution to the following problems:

$$\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2+b_1^2\right)+\sum_{i=1}^{m_1}\frac{(\mathbf{h}_i\mathbf{z}_1)^2}{\mathbf{D}_{1i}}+\frac{1}{2}c_1\mathbf{q}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -(\mathbf{B}\mathbf{w}_1+\mathbf{e}_2b_1)+\mathbf{q}_2=\mathbf{e}_2, \tag{30}$$

$$\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \frac{1}{2}c_4\left(\|\mathbf{w}_2\|_2^2+b_2^2\right)+\sum_{j=1}^{m_2}\frac{\left(\mathbf{g}_j\mathbf{z}_2\right)^2}{\mathbf{D}_{2j}}+\frac{1}{2}c_2\mathbf{q}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad (\mathbf{A}\mathbf{w}_2+\mathbf{e}_1b_2)+\mathbf{q}_1=\mathbf{e}_1, \tag{31}$$

where $\mathbf{D}_{1i}=2|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}|$, $\mathbf{D}_{2j}=2|\mathbf{g}_j{\mathbf{z}_2}^{(p)^T}|$, $h_i=[\mathbf{A}_i \quad \mathbf{e}_1^i]$, $g_j=[\mathbf{B}_j \quad \mathbf{e}_2^j]$. Hence, we rewrite formulas (30) and (31) as follows:

$$\min_{\mathbf{z}_1,\mathbf{q}_2} \frac{1}{2}c_3\mathbf{z}_1^T\mathbf{z}_1+\mathbf{z}_1^T\mathbf{H}^T\mathbf{D}_1\mathbf{H}\mathbf{z}_1+\frac{1}{2}c_1\mathbf{q}_2^T\mathbf{q}_2$$
$$\text{s.t.} \quad -\mathbf{G}\mathbf{z}_1+\mathbf{q}_2=\mathbf{e}_2, \tag{32}$$

$$\min_{\mathbf{z}_2,\mathbf{q}_1} \frac{1}{2}c_4\mathbf{z}_2^T\mathbf{z}_2+\mathbf{z}_2^T\mathbf{G}^T\mathbf{D}_2\mathbf{G}\mathbf{z}_2+\frac{1}{2}c_2\mathbf{q}_1^T\mathbf{q}_1$$
$$\text{s.t.} \quad \mathbf{H}\mathbf{z}_2+\mathbf{q}_1=\mathbf{e}_1, \tag{33}$$

where $\mathbf{D}_1=diag(\mathbf{D}_{11},\mathbf{D}_{12},\ldots,\mathbf{D}_{1m_1})$, $\mathbf{D}_2=diag(\mathbf{D}_{21},\mathbf{D}_{22},\ldots,\mathbf{D}_{2m_2})$, and $\mathbf{H}=[\mathbf{A} \quad \mathbf{e}_1]$, $\mathbf{G}=[\mathbf{B} \quad \mathbf{e}_2]$.

The algorithm which we design is depicted in Algorithm 1.

Algorithm 1 A valid iterative algorithm to solve the optimization problem of formula (26). To prove it, we first introduce Lemma 1.

Input: Matrices $\mathbf{H}\in R^{m_1\times(n+1)}$ and $\mathbf{G}\in R^{m_2\times(n+1)}$.

Result: $\mathbf{z}_1\in R^{(n+1)\times 1}$. Set $p=0$, Initialize diagonal matric $\mathbf{D}_1\in R^{m_1\times m_1}$, $\mathbf{D}_1=diag(1/|\mathbf{H}\mathbf{z}_1|)$, and initialize $\mathbf{z}_1$.

Repeat

1. Calculate

$$\mathbf{z}_1{}^{(P+1)}=-\left(\frac{1}{c_1}\left(\mathbf{H}^T\mathbf{D}_1{}^{(p)}\mathbf{H}+c_3\mathbf{I}\right)+\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{e}_2 \tag{34}$$

2. Update matrix $\mathbf{D}_1{}^{(p+1)}$ based on the current $\mathbf{z}_1^{(p+1)}$, the i-th element of $\mathbf{D}_1{}^{(p+1)}$ is $1/\left(2\left|\mathbf{h}_i\mathbf{z}_1^{(p+1)^T}\right|\right)$.

3. $p=p+1$.

Until Convergence.

Output: The optimal solution of $\mathbf{z}_1$.

In each iteration, $\mathbf{z}_1$ is calculated with the current $\mathbf{D}_1$, and then $\mathbf{D}_1$ is updated based on the current calculated $\mathbf{z}_1$. The iteration procedure is repeated until the algorithm converges.

Lemma 1. *For any nonzero vector* $\mathbf{z}$, $\mathbf{z}^p \in R$, *the following inequality is established:*

$$\|\mathbf{z}\|_1 - \frac{\|\mathbf{z}\|_1^2}{2\|\mathbf{z}^p\|_1} \le \|\mathbf{z}^p\|_1 - \frac{\|\mathbf{z}^p\|_1^2}{2\|\mathbf{z}^p\|_1} \tag{35}$$

Proof. Start with an explicit inequality $\left(\sqrt{u} - \sqrt{u^p}\right)^2 \ge 0$, we have

$$\left(\sqrt{u} - \sqrt{u^p}\right)^2 \ge 0 \Rightarrow u - 2\sqrt{uu^p} + u^p \ge 0 \Rightarrow \sqrt{u} - \frac{u}{2\sqrt{u^p}} \le \frac{\sqrt{u^p}}{2}$$
$$\Rightarrow \sqrt{u} - \frac{u}{2\sqrt{u^p}} \le \sqrt{u^p} - \frac{u^p}{2\sqrt{u^p}} \tag{36}$$

Replace u and u^p in (36) with $\|\mathbf{z}\|_1^2$ and $\|\mathbf{z}^p\|_1^2$ respectively, we reach (35). □

Note that the proof is motivated by [29], we cannot set $\mathbf{D}_{1i}^p = 0$ when $|\mathbf{h}_i\mathbf{z}_1{}^{(p)^T}| = 0$, otherwise the derived algorithm cannot be ensured to converge. We have two ways to solve this. First, we can slightly move $\mathbf{z}_1{}^{(p)}$ in the way of $\mathbf{z}_1{}^{(p)} = (\mathbf{z}_1{}^{(p)} + \mathbf{\Delta})/||\mathbf{z}_1{}^{(p)} + \mathbf{\Delta}||$, motivated by [22], where $\mathbf{\Delta}$ is a very small random vector. However, this method is not practical, since we may need to try a large number of times. Second, we can regularize $\mathbf{D}_1^p$ as $\mathbf{D}_{1i}^p = 1/(\sqrt{(\mathbf{h}_i\mathbf{z}_1^{(p)})^2 + \varsigma})$, where ς is a very small number. When $\varsigma \to 0$, $\mathbf{D}_{1i}^p = 1/(\sqrt{(\mathbf{h}_i\mathbf{z}_1^{(p)})^2 + \varsigma})$ approximates the original one. The second method is used in our problem, since it has been a popular method and widely used for solving the same problem existed in L1-related feature extractions.

The convergence of Algorithm 1 is summarized in the following Theorem 1.

Theorem 1. *Assume that* $\mathbf{z}_1^{(p)}$ *is a vector, then formula (32) is built,* $\mathbf{z}_1^{(p+1)}$ *is the solution obtained from (32), then* $\mathbf{z}_1^{(p+1)}$ *is deemed to be better than* $\mathbf{z}_1^{(p)}$. *Here, we solve formula (32) in closed form in first iteration and later use update from Algorithm 1 until we obtain the optimum solution of* $\mathbf{z}_1^{(p+1)}$.

Proof. In the problem in (34), it is the solution of the following problem

$$\mathbf{z}_1^{(p+1)} = \arg\min_{\mathbf{z}_1,\mathbf{q}_2 - \mathbf{G}\mathbf{z}_1 + \mathbf{q}_2 = \mathbf{e}_2} \frac{1}{2}c_3(\mathbf{z}_1)^T\mathbf{z}_1 + \frac{1}{2}(\mathbf{H}\mathbf{z}_1)^T\mathbf{D}_1^p(\mathbf{H}\mathbf{z}_1) + \frac{1}{2}c_1(\mathbf{q}_2)^T\mathbf{q}_2 \tag{37}$$

According to (34), we obtain

$$\frac{1}{2}c_3\left(\mathbf{z}_1{}^{(p+1)}\right)^T\mathbf{z}_1{}^{(p+1)} + \frac{1}{2}\left(\mathbf{H}\mathbf{z}_1{}^{(p+1)}\right)^T\mathbf{D}_1^p\left(\mathbf{H}\mathbf{z}_1{}^{(p+1)}\right) + \frac{1}{2}c_1\left(\mathbf{q}_2^{(p+1)}\right)^T\mathbf{q}_2^{(p+1)}$$
$$\le \frac{1}{2}c_3\left(\mathbf{z}_1^p\right)^T\mathbf{z}_1^p + \frac{1}{2}\left(\mathbf{H}\mathbf{z}_1^p\right)^T\mathbf{D}_1^p\left(\mathbf{H}\mathbf{z}_1^p\right) + \frac{1}{2}c_1\left(\mathbf{q}_2^p\right)^T\mathbf{q}_2^p \tag{38}$$

Which is simplified as

$$\frac{1}{2}c_3\left(\mathbf{z}_1{}^{(p+1)}\right)^T\mathbf{z}_1{}^{(p+1)} + \sum_{i=1}^{m_1}\frac{\left(\mathbf{h}_i\mathbf{z}_1{}^{(p+1)}\right)^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)^T}|} + \frac{1}{2}c_1\left(\mathbf{q}_2^{(p+1)}\right)^T\mathbf{q}_2^{(p+1)}$$
$$\le \frac{1}{2}c_3\left(\mathbf{z}_1^p\right)^T\mathbf{z}_1^p + \sum_{i=1}^{m_1}\frac{(\mathbf{h}_i\mathbf{z}_1{}^{(p)})^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)}|} + \frac{1}{2}c_1\left(\mathbf{q}_2^p\right)^T\mathbf{q}_2^p \tag{39}$$

According to Lemma 1, we can get

$$|\mathbf{h}_i\mathbf{z}_1{}^{(p+1)}| - \frac{(\mathbf{h}_i\mathbf{z}_1{}^{(p+1)})^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)}|} \le |\mathbf{h}_i\mathbf{z}_1{}^{(p)}| - \frac{(\mathbf{h}_i\mathbf{z}_1{}^{(p)})^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)}|} \tag{40}$$

Which leads to

$$\sum_{i=1}^{m_1}\left(|\mathbf{h}_i\mathbf{z}_1{}^{(p+1)}| - \frac{(\mathbf{h}_i\mathbf{z}_1{}^{(p+1)})^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)}|}\right) \le \sum_{i=1}^{m_1}\left(|\mathbf{h}_i\mathbf{z}_1{}^{(p)}| - \frac{(\mathbf{h}_i\mathbf{z}_1{}^{(p)})^2}{2|\mathbf{h}_i\mathbf{z}_1{}^{(p)}|}\right) \tag{41}$$

Combining Eqs. (39) and (41), we have

$$\frac{1}{2}c_3\left(\mathbf{z}_1{}^{(p+1)}\right)^T\mathbf{z}_1{}^{(p+1)} + \sum_{i=1}^{m_1}|\mathbf{h}_i\mathbf{z}_1{}^{(p+1)}| + \frac{1}{2}c_1\left(\mathbf{q}_2^{(p+1)}\right)^T\mathbf{q}_2^{(p+1)}$$
$$\le \frac{1}{2}c_3\left(\mathbf{z}_1^p\right)^T\mathbf{z}_1^p + \sum_{i=1}^{m_1}|\mathbf{h}_i\mathbf{z}_1{}^{(p)}| + \frac{1}{2}c_1\left(\mathbf{q}_2^p\right)^T\mathbf{q}_2^p \tag{42}$$

which shows the objective function value of formula (26) decreases monotonically in each iteration in Algorithm 1. In the convergence, $\mathbf{z}_1^{(p+1)}$ is an optimal solution to the convex problem in formula (26). □

Formula (26) is a convex optimization problem with constraints in the form of equalities, and has a close-form solution. On substituting the equality constraints into the objective function, and converting L1-norm problem to L2-norm, as shown in formulas (28), (30) and (32), we rewrite (26) as the following form:

$$\min_{\mathbf{w}_1,b_1}\frac{1}{2}c_3\left(\|\mathbf{w}_1\|_2^2 + b_1^2\right) + (\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1b_1)\mathbf{D}_1(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1b_1)^{\mathrm{T}} + \frac{1}{2}c_1\|\mathbf{e}_2 + \mathbf{B}\mathbf{w}_1 + \mathbf{e}_2b_1\|_2^2$$
$$s.t. -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2b_1) + \mathbf{q}_2 = \mathbf{e}_2, \tag{43}$$

Setting the gradient of (43) with respect to $\mathbf{w}_1$ and b_1 to zero, we can obtain,

$$c_3\mathbf{w}_1 + 2\mathbf{A}^T\mathbf{D}_1(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1b_1) + c_1\mathbf{B}^T(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2b_1 + \mathbf{e}_2) = 0 \tag{44}$$

$$c_3b_1 + 2\mathbf{e}_1^T\mathbf{D}_1(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1b_1) + c_1\mathbf{e}_2^T(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2b_1 + \mathbf{e}_2) = 0 \tag{45}$$

Combining Eqs. (44) and (45), we have that,

$$\frac{c_3\mathbf{I}}{c_1}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} + \frac{1}{c_1}\begin{bmatrix}2\mathbf{A}^T\mathbf{D}_1\mathbf{A} & 2\mathbf{A}^T\mathbf{D}_1\mathbf{e}_1\\ 2\mathbf{e}_1^T\mathbf{D}_1\mathbf{A} & 2m_1\mathbf{D}_1\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} + \begin{bmatrix}\mathbf{B}^T\mathbf{B} & \mathbf{B}^T\mathbf{e}_2\\ \mathbf{e}_2^T\mathbf{B} & m_2\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} + \begin{bmatrix}\mathbf{B}^T\mathbf{e}_2\\ m_2\end{bmatrix} = 0 \tag{46}$$

Eq. (46) is equivalent to

$$\left(\frac{c_3\mathbf{I}}{c_1} + \frac{1}{c_1}\begin{bmatrix}2\mathbf{A}^T\mathbf{D}_1\mathbf{A} & 2\mathbf{A}^T\mathbf{D}_1\mathbf{e}_1\\ 2\mathbf{e}_1^T\mathbf{D}_1\mathbf{A} & 2m_1\mathbf{D}_1\end{bmatrix} + \begin{bmatrix}\mathbf{B}^T\mathbf{B} & \mathbf{B}^T\mathbf{e}_2\\ \mathbf{e}_2^T\mathbf{B} & m_2\end{bmatrix}\right)\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} = -\begin{bmatrix}\mathbf{B}^T\mathbf{e}_2\\ m_2\end{bmatrix} \tag{47}$$

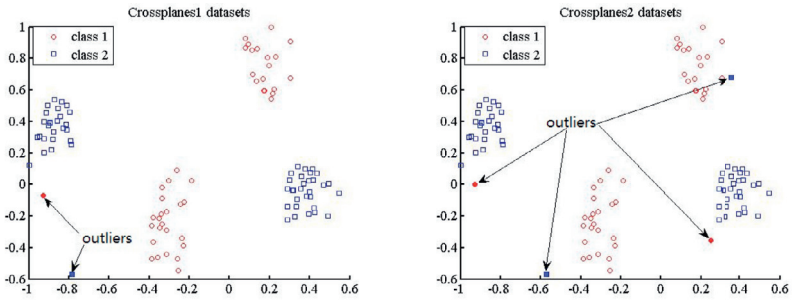
Thus, we can obtain

$$\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} = \left(\begin{bmatrix}(2/c_1)\mathbf{A}^T\mathbf{D}_1\mathbf{A}+\mathbf{B}^T\mathbf{B} & (2/c_1)\mathbf{A}^T\mathbf{D}_1\mathbf{e}_1+\mathbf{B}^T\mathbf{e}_2\\ (2/c_1)\mathbf{e}_1^T\mathbf{D}_1\mathbf{A}+\mathbf{e}_2^T\mathbf{B} & (2/c_1)m_1\mathbf{D}_1+m_2\end{bmatrix} + \frac{c_3\mathbf{I}}{c_1}\right)^{-1} \times \begin{bmatrix}-\mathbf{B}^T\mathbf{e}_2\\ -m_2\end{bmatrix} \tag{48}$$

That is,

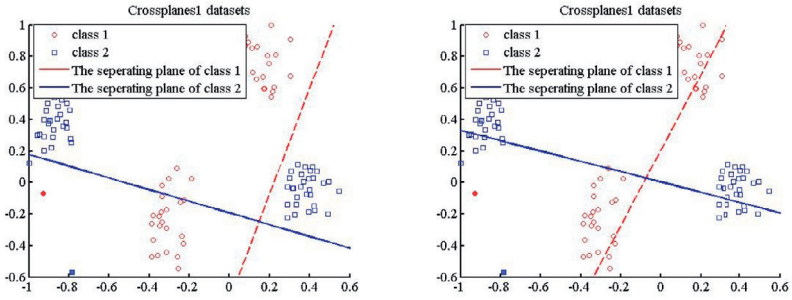
$$\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix} = \left[\frac{2}{c_1}\begin{bmatrix}\mathbf{A}^T\\ \mathbf{e}_1^T\end{bmatrix}\mathbf{D}_1[\mathbf{A}\ \ \mathbf{e}_1] + \begin{bmatrix}\mathbf{B}^T\\ \mathbf{e}_2^T\end{bmatrix}[\mathbf{B}\ \ \mathbf{e}_2] + \frac{c_3\mathbf{I}}{c_1}\right]^{-1}\begin{bmatrix}-\mathbf{B}^T\mathbf{e}_2\\ -m_2\end{bmatrix} \tag{49}$$

So we can get the solution of $\mathbf{z}_1^{(p+1)}$ as shown in the following: $\mathbf{z}_1{}^{(p+1)} = [\mathbf{w}_1\ \ b_1]^T = -\left(\frac{2}{c_1}\left(\mathbf{H}^T\mathbf{D}_1\mathbf{H} + \frac{1}{2}c_3\mathbf{I}\right) + \mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{e}_2$.



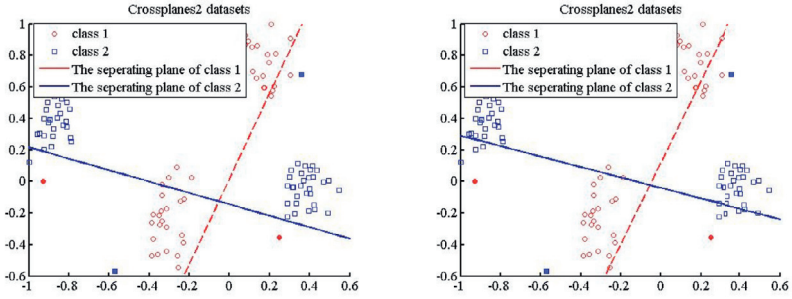
(a) With two outliers (b) With four outliers

Fig. 1. XOR datasets.



(a) By TBSVM (b) By L1-LSTBSVM

Fig. 2. Two classification planes on Crossplanes1.



(a) By TBSVM (b) By L1-LSTBSVM

Fig. 3. Two classification planes on Crossplanes2.

Since c_1 and c_3 are two tuning parameters, $\mathbf{z}_1^{(p+1)}$ can expressed equivalently as

$$\mathbf{z}_1{}^{(p+1)} = [\mathbf{w}_1 \;\; b_1]^T = -\left(\frac{1}{c_1}\left(\mathbf{H}^T\mathbf{D}_1\mathbf{H} + c_3\mathbf{I}\right) + \mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{e}_2 \tag{50}$$

Note that $\mathbf{D}_1$ is dependent on $\mathbf{z}_1^{(p+1)}$ and it is also an unknown variable, we can iteratively update the value of $\mathbf{D}_1$ to change $\mathbf{z}_1^{(p+1)}$. In other words, increase p until the objective function value converges to a fixed value, then, $\mathbf{z}_1^{(p+1)}$ is the optimal solution that we seek. In this study, we propose an iterative algorithm to obtain the optimum solution $\mathbf{z}_1^{(p+1)}$ to satisfy Eq. (50), and have demonstrated that the proposed iterative algorithm converges to an optimum solution.

Similar to $\mathbf{z}_2^{(p+1)}$.

$$\mathbf{z}_2{}^{(p+1)} = [\mathbf{w}_2 \;\; b_2]^T = \left(\frac{1}{c_2}\left(\mathbf{G}^T\mathbf{D}_2\mathbf{G} + c_4\mathbf{I}\right) + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{e}_1 \tag{51}$$

where $\mathbf{D}_2 = diag\left(1/\left|\mathbf{G}\mathbf{z}_2{}^{(p)^T}\right|\right)$. So weight vectors $\mathbf{w}_1$, $\mathbf{w}_2$ and deviations b_1, b_2 can be obtained, two non-parallel optimal planes are given by

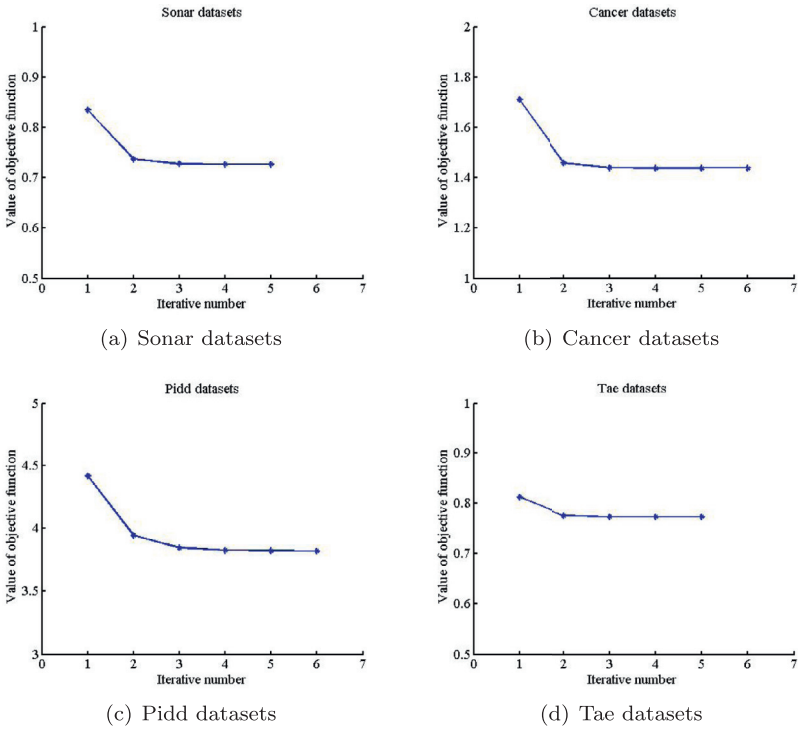
$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0, \;\; \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{52}$$

A new point $\mathbf{x} \in \mathrm{R}^n$ is assigned to class i (i=1, 2), according to which of the two non-parallel planes in (52) is closer to the decision function

$$f(\mathbf{x}) = \arg\min_{i=1,2}\left(\left|\mathbf{x}^T\mathbf{w}_{1,2} + b_{1,2}\right| / \|\mathbf{w}_{1,2}\|\right) \tag{53}$$

Here, $|\cdot|$ is the absolute value. Next, we illustrate the effectiveness and robustness of L1-LSTBSVM by experiments, and the classification accuracy is demonstrated by the experimental results on artificial datasets and UCI datasets [30,31].

Last, begin with (26), we explain why the primal problems of formulas (26) and (27) can be optimized by (28) and (29). Define the function $f(\mathbf{w}_1, b_1) = ||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_1$ in (26). The main difference between (26) and (28) is that $f(\mathbf{w}_1, b_1)$ is replaced with $\sum_{i=1}^{m_1} \frac{(\mathbf{A}_i\mathbf{w}_1 + \mathbf{e}_1^i b_1)}{\mathbf{D}_i}^2$. We know the gradient of $f(\mathbf{w}_1, b_1)$ w.r.t. $\mathbf{w}_1$, b_1 is $\sum_{i=1}^{m_1} \frac{2\mathbf{h}_i^T\mathbf{h}_i\mathbf{z}_1}{2|\mathbf{h}_i\mathbf{z}_1|} = \sum_{i=1}^{m_1} \frac{2\mathbf{h}_i^T\mathbf{h}_i\mathbf{z}_1}{\mathbf{D}_i}$, which is equivalent to the gradient



(a) Sonar datasets (b) Cancer datasets

(c) Pidd datasets (d) Tae datasets

Fig. 4. The objective function values of L1-LSTBSVM along with the iterative number on four datasets.

of $\sum_{i=1}^{m_1} \frac{\left(\mathbf{A}_i\mathbf{w}_1+\mathbf{e}_1^i b_1\right)^2}{\mathbf{D}_i}$ in (28) when $\mathbf{D}_i$ is given. This implies that applying the iterative algorithm in Algorithm 1 to (26) generates the same solution as applying Algorithm 1 to (28).

4. Experimental results

To confirm the classification performance of L1-LSTBSVM, it is compared with four algorithms (TWSVM [12], LSTWSVM [15], TBSVM [14] and L1-NPSVM [20]) on UCI datasets, which reflects the performance of the algorithm [32,33]. L1-NPSVM and our L1-LSTBSVM are two iterative algorithms, which need to set the initial solutions. Consider that these two algorithms aim to correct the planes of GEPSVM and TBSVM which may be unsatisfied because of the effect of outliers, we set their initial solutions as the solutions of GEPSVM and LSTBSVM, respectively. L1-NPSVM and our L1-LSTBSVM are inspired by L1-norm distance LDA methods [22]. Therefore, this setting also follows [22]. For them, we stop the iterative procedures when the difference of objective values of two successive iterations is less than 0.001 and the iterative number is greater than 50. Experimental environment: Windows10 operating system, PC with an Intel(R) Core(TM) i5-5200u, quad core processor (2.2GHz), 4GB of RAM. Five kinds of classification algorithms are implemented in MATLAB 7.1. The experimental parameters are selected by 10-fold cross validation method [17,34], and testing accuracy is the average value of results for 10 times for each datasets. In addition, the experimental data only contain two types (class 1 and class 2), and all the sample data are normalized by the interval $[-1, 1]$ to reduce the difference between the characteristics of different samples. As is known, experimental parameters may influence the classification accuracy. Thus, to obtain the best generalization performance, all experimental parameters are chosen as below. Parameters c_1, c_2, c_3, c_4 are in the range of $\left\{2^i \,|\, i = -16, -15, -14 \ldots, 15, 16\right\}$, while parameter ε is in the range of $\left\{10^i \,|\, i = -10, -9, -8 \ldots, 10\right\}$.

4.1. Experiments on artificial datasets

To verify the performance of our L1-LSTBSVM, we did the experiments on a simple 2-D XOR datasets (called Crossplanes (100×2), the number of positive samples is 44, while negative samples have 56 data points), which is generated by perturbing points originally lying on two intersecting planes (lines), and each plane corresponds to one class. As is known, outliers tend to have a certain effect on the classification performance, which is also the standard for measuring the stability of the algorithms. Thus, we introduce outliers into Crossplanes datasets to evaluate the robustness of TBSVM and L1-LSTBSVM. Here, two or four extra outliers are added on the Crossplanes datasets, called Crossplanes1 (102×2) and Crossplanes2 (104×2) respectively, as shown in Fig. 1 (a) and (b), two classes of samples are distinguished by "$\circ$" and "$\square$", respectively. The classification results of each classifier on this dirty XOR datasets are given in Figs. 2 and 3 respectively.

Figs. 2 and 3 describe the results of two methods on the Crossplanes datasets where two and four outliers were introduced respectively, from which we can observe that the learnt two non-parallel optimal separating planes characterize the Crossplanes1 and Crossplanes2 datasets well by robust L1-norm distance. Compared with L1-LSTBSVM, TBSVM misclassifies more points (class 1 has more points closer to the blue separating plane of class 2). That is, the proposed L1-LSTBSVM approach can deal with the outliers better than TBSVM due to the robust property of L1-norm distance against the negative effect of outliers, which is the main motivation for us to demonstrate these figures. The classification accuracy of TBSVM and L1-LSTBSVM on Crossplanes1 are 42.14% and 66.38% respectively, and on Crossplanes2 49.76% and 61.38% respectively, which reveals that the classification ability of L1-LSTBSVM is better after introducing outliers. Presented results explain that classifiers based on L2-norm distance are sensitive to outliers, and this may lead to the large distance dominating the sum in TBSVM, but the L1-norm distance can powerfully

Table 1
Test results of TBSVM, TWSVM, LSTWSVM, L1-NPSVM and L1-LSTBSVM.

	TBSVM	TWSVM	LSTWSVM	L1-NPSVM	L1-LSTBSVM
Datasets (N × n)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s) Iterative number
Cancer	96.20 ± 1.81	92.54 ± 0.95	96.05 ± 1.99	86.23 ± 10.20	**96.49 ± 1.75***
(683 × 9)	1.143	1.148	0.109	0.438	0.825 6
Ticdata	67.33 ± 2.35	67.54 ± 2.97	68.48 ± 3.45	66.81 ± 2.81	**68.79 ± 4.19***
(958 × 9)	3.709	3.821	0.122	1.197	1.419 4
Ktest	54.60 ± 5.77	54.60 ± 5.77	54.51 ± 5.66	53.19 ± 4.36	**55.13 ± 1.42***
(1130 × 5)	10.139	9.850	0.106	0.191	0.932 4
Pidd	76.69 ± 3.35	75.39 ± 2.31	**77.21 ± 3.66**	74.87 ± 5.01	76.17 ± 3.99
(768 × 8)	3.910	2.582	0.134	0.258	0.981 6
ClaveVectors	**74.35 ± 2.48**	73.52 ± 2.10	74.04 ± 2.16	73.31 ± 3.33	73.52 ± 2.10
(963 × 19)	4.311	4.401	0.262	0.476	0.293 4
Sonar	**79.83 ± 5.30**	68.70 ± 5.55	72.08 ± 4.94	74.14 ± 9.97	75.99 ± 4.86
(208 × 60)	1.248	1.365	0.546	1.231	0.755 5
Pimadata	76.17 ± 2.16	74.39 ± 2.31	**76.56 ± 2.91**	73.70 ± 4.02	74.48 ± 2.00
(768 × 8)	1.978	2.911	0.104	0.262	0.140 3
Ionodata	86.33 ± 3.76	85.75 ± 5.66	90.60 ± 1.93	80.06 ± 6.36	**90.88 ± 3.80***
(351 × 34)	1.145	1.030	0.774	1.161	0.828 4
Clevedata	85.21 ± 4.22	83.85 ± 3.71	81.49 ± 3.33	84.53 ± 5.17	**85.55 ± 6.65***
(297 × 13)	0.464	0.330	0.117	0.342	0.278 3
Housingdata	84.39 ± 3.50	78.66 ± 5.48	**86.18 ± 3.32**	72.32 ± 5.28	77.67 ± 3.33
(506 × 13)	3.290	1.009	0.121	0.278	0.339 4
Tae	**83.42 ± 2.25**	82.75 ± 3.37	83.40 ± 4.32	79.38 ± 10.99	82.75 ± 3.37
(151 × 5)	0.313	0.209	0.083	0.140	0.151 5
Mushdata	81.66 ± 0.95	80.75 ± 0.97	81.60 ± 0.88	78.90 ± 9.26	**84.11 ± 1.08***
(8124 × 22)	67.093	63.269	1.005	6.724	13.480 4
Spect	80.17 ± 5.03	79.43 ± 5.61	78.69 ± 6.23	76.45 ± 5.95	**80.89 ± 3.26***
(267 × 44)	5.494	1.376	1.105	0.948	1.297 6
Brightdata	98.62 ± 0.43	98.58 ± 0.57	98.74 ± 0.35	95.66 ± 2.62	**99.03 ± 0.30***
(2462 × 14)	11.733	11.685	0.393	2.083	1.523 4
Dimdata	93.92 ± 0.57	**95.13 ± 0.65**	94.68 ± 0.47	89.74 ± 1.30	93.87 ± 0.91
(4192 × 14)	26.741	27.288	0.310	7.809	5.603 4

Table 2
Test results of five algorithms when 10% Gaussian noise is introduced.

	TBSVM	TWSVM	LSTWSVM	L1-NPSVM	L1-LSTBSVM
Datasets (N × n)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s) Iterative number
Cancer	**96.78 ± 1.18**	92.10 ± 0.70	93.42 ± 1.65	85.94 ± 8.89	96.05 ± 1.76
(683 × 9)	2.119	1.123	0.306	0.450	0.574 6
Ticdata	67.12 ± 3.42	64.83 ± 2.24	67.64 ± 2.89	66.60 ± 2.89	**68.89 ± 2.61***
(958 × 9)	3.734	3.677	0.322	1.194	1.483 4
Ktest	55.04 ± 5.43	55.04 ± 5.43	**55.13 ± 2.69**	53.36 ± 4.41	55.04 ± 5.43
(1130 × 5)	9.647	9.876	0.264	0.340	0.880 4
Pidd	75.25 ± 5.17	73.57 ± 1.60	75.12 ± 3.88	75.39 ± 3.70	**75.91 ± 2.48***
(768 × 8)	2.797	2.722	0.288	0.238	0.735 6
ClaveVectors	73.42 ± 2.50	73.73 ± 1.81	**73.83 ± 2.15**	73.73 ± 2.73	**73.83 ± 2.03***
(963 × 19)	5.386	4.693	0.683	0.583	0.302 3
Sonar	75.97 ± 3.34	68.75 ± 4.88	70.13 ± 5.76	74.64 ± 10.56	**77.42 ± 4.09***
(208 × 60)	1.278	1.503	0.542	1.332	0.367 5
Pimadata	74.99 ± 5.20	74.86 ± 4.64	74.86 ± 3.84	75.39 ± 3.70	**75.78 ± 3.56***
(768 × 8)	2.683	2.585	0.287	0.269	0.301 4
Ionodata	85.76 ± 4.32	86.32 ± 4.12	89.74 ± 2.46	81.17 ± 8.23	**92.01 ± 4.30***
(351 × 34)	3.065	1.302	0.756	1.253	1.263 4
Clevedata	84.54 ± 4.22	75.76 ± 7.79	77.45 ± 6.44	85.53 ± 5.17	**85.89 ± 6.40***
(297 × 13)	0.377	0.516	0.115	0.320	1.470 4
Housingdata	**82.21 ± 5.28**	71.15 ± 6.28	81.03 ± 4.30	72.32 ± 5.28	77.09 ± 5.56
(506 × 13)	3.209	1.575	0.328	0.307	0.816 4
Tae	**84.73 ± 2.80**	83.40 ± 3.77	84.06 ± 3.99	80.71 ± 10.98	83.40 ± 3.77
(151 × 5)	0.390	0.580	0.226	0.110	0.690 5
Mushdata	81.65 ± 0.95	78.52 ± 1.37	81.52 ± 0.94	70.47 ± 7.68	**84.06 ± 1.09***
(8124 × 22)	64.517	61.656	1.036	5.407	18.851 4
Spect	79.43 ± 5.61	79.06 ± 5.88	79.43 ± 6.51	75.70 ± 8.23	**80.52 ± 2.86***
(267 × 44)	5.367	1.340	1.074	1.194	0.903 4
Brightdata	96.75 ± 0.49	97.08 ± 0.33	96.10 ± 0.24	95.16 ± 3.70	**97.85 ± 0.37***
(2462 × 14)	11.565	12.057	0.845	2.247	4.183 4
Dimdata	92.75 ± 0.85	**94.27 ± 0.74**	92.27 ± 0.85	89.67 ± 1.66	92.87 ± 1.15
(4192 × 14)	22.748	26.060	1.121	2.812	5.771 4

Table 3
Introduce 20% Gaussian noise, test results of five algorithms

	TBSVM	TWSVM	LSTWSVM	L1-NPSVM	L1-LSTBSVM
Datasets (N × n)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s)	Accuracy(%) Training time(s) Iterative number
Cancer	96.34 ± 1.72	96.19 ± 0.85	96.05 ± 1.99	86.97 ± 6.83	**96.35 ± 1.84***
(683 × 9)	1.193	1.121	0.402	0.569	0.858 6
Ticdata	66.81 ± 2.35	67.64 ± 3.09	**68.79 ± 3.31**	66.91 ± 1.83	68.69 ± 4.40
(958 × 9)	3.723	3.703	0.447	1.229	1.466 4
Ktest	54.78 ± 4.51	52.57 ± 1.57	55.02 ± 5.00	53.72 ± 4.13	**55.13 ± 4.48***
(1130 × 5)	10.736	10.409	0.384	0.232	0.904 4
Pidd	76.04 ± 2.12	**76.18 ± 3.43**	75.91 ± 2.35	74.74 ± 5.69	75.79 ± 4.79
(768 × 8)	2.719	2.779	0.421	0.248	0.989 6
ClaveVectors	74.04 ± 1.66	72.38 ± 2.63	72.69 ± 2.38	73.21 ± 3.17	**74.25 ± 2.18***
(963 × 19)	4.697	4.774	0.787	0.683	1.798 4
Sonar	74.03 ± 5.74	73.01 ± 7.65	73.53 ± 4.21	74.60 ± 9.91	**76.48 ± 5.38***
(208 × 60)	1.031	0.970	0.522	1.352	0.415 5
Pimadata	**76.04 ± 2.12**	75.13 ± 4.13	75.77 ± 3.41	74.62 ± 3.65	75.39 ± 3.66
(768 × 8)	2.780	2.723	0.410	0.723	1.355 5
Ionodata	87.18 ± 4.32	87.75 ± 4.67	90.03 ± 1.57	83.19 ± 7.50	**92.31 ± 4.10***
(351 × 34)	1.148	1.027	0.748	0.661	1.325 4
Clevedata	84.88 ± 4.29	84.19 ± 3.70	81.49 ± 4.24	85.18 ± 5.02	**86.22 ± 3.81***
(297 × 13)	0.389	0.332	0.122	0.351	1.513 4
Housingdata	**81.62 ± 3.32**	80.44 ± 3.73	79.84 ± 4.21	72.32 ± 5.28	76.68 ± 3.31
(506 × 13)	3.154	1.050	0.148	0.367	1.493 3
Tae	82.06 ± 5.09	81.40 ± 5.10	82.06 ± 5.51	80.71 ± 10.98	**82.09 ± 3.52***
(151 × 5)	0.819	0.602	0.238	0.125	0.383 5
Mushdata	80.66 ± 0.90	80.64 ± 0.91	81.25 ± 0.99	73.20 ± 2.64	**83.87 ± 1.34***
(8124 × 22)	73.373	74.136	1.228	5.354	21.283 4
Spect	79.80 ± 5.42	79.01 ± 5.56	78.30 ± 5.68	76.05 ± 9.65	**80.90 ± 3.64***
(267 × 44)	5.668	1.708	1.121	1.294	1.835 4
Brightdata	96.10 ± 1.15	97.68 ± 0.27	93.46 ± 1.29	96.79 ± 1.73	**97.77 ± 0.38***
(2462 × 14)	12.364	11.641	0.406	2.048	4.358 4
Dimdata	92.22 ± 0.77	**93.77 ± 0.37**	91.56 ± 1.06	89.79 ± 1.33	92.56 ± 0.75
(4192 × 14)	23.583	23.716	1.181	2.807	5.522 4

suppress the influence of outliers, which has effectively testified the practicability of L1-LSTBSVM.

4.2. Experiments on UCI datasets

The algorithm that we design iteratively updates the objective function value until it converges to a fixed value. Fig. 4 (a)–(d) show the objective function values of L1-LSTBSVM monotonically decrease with the iteration and the algorithm fast converges within about 6 iterations, which is completely consistent with our former theoretical analysis. Horizontal axis represents the number of iterations, and vertical axis represents the value of objective function.

To further validate the usefulness and practicality of L1-LSTBSVM, it is compared with the related algorithms (TBSVM [14], TWSVM [12], LSTWSVM [15] and L1-NPSVM [20]) on fifteen commonly used datasets that are selected from the UCI datasets, under the consideration that noise is one of the standards to measure the robustness of the algorithm. The classification accuracy changes smoothly with the increase of noise, which indicates the algorithm has better anti-noise ability. Next, we verify this by experiments.

Table 1 is the comparison of classification accuracy within five algorithms, while Table 2 shows the comparison of the five algorithms on the fifteen commonly used data where 10% Gaussian noise was introduced respectively. Table 3 is the comparison of the classification accuracy within the five algorithms on the data where 20% Gaussian noise was introduced respectively. Moreover, to further verify the convergence of L1-LSTBSVM, the average training iterative numbers of this algorithm are listed in the three tables corresponding to each experiment. The best classification accuracy is marked black bold, and * represents that the classification accuracy of L1-LSTBSVM is the best. Detailed information is exhibited in the following tables:

From the data in Table 1, it is observed that the classification accuracy of L1-LSTBSVM is much higher than other algorithms in most cases, with the training time less than that of TBSVM and TWSVM, but higher than that of LSTWSVM and L1-NPSVM. As seen in Table 2 and Table 3, 10% and 20% Gaussian noise were introduced to the UCI datasets separately. Results show that our L1-LSTBSVM obtains the best result compared to other methods in most cases, and the training time is also less than that of TBSVM and TWSVM. Although LSTWSVM and L1-NPSVM take less training time, they cannot catch up with L1-LSTBSVM in classification accuracy. From the columns of L1-LSTBSVM in the three tables above, it can be found that L1-LSTBSVM can fast converge within about 6 iterations.

From the experimental results in the three tables above, it is observed that L1-LSTBSVM is a winner when comparing it with four other methods in most cases. This is attributed to the iterative method we proposed, which is helpful for the resistance of the negative influence of outliers. This demonstrates that our proposed L1-LSTBSVM is advantageous to the correct data classification. A common point of L1-NPSVM and L1-LSTBSVM is the embedding of L1-norm distance. As two effective L1-norm classification methods, the accuracy of L1-NPSVM and L1-LSTBSVM only has a little change compared to other three methods, especially when Gaussian noise is introduced. These indicate L1-norm distance is stronger than L2-norm distance in the presence of outliers. Therefore, the experimental results have effectively confirmed our claims that the robustness and classification performance of L1-LSTBSVM are the best, particularly when Gaussian noise is introduced.

Furthermore, from the tables we can see, TBSVM and TWSVM cost much training time in datasets of Mushdata, Brightdata and Dimdata, because they need to solve a pair of QPPs, when used to handle large datasets, their computing speed is lower. Neverthe-

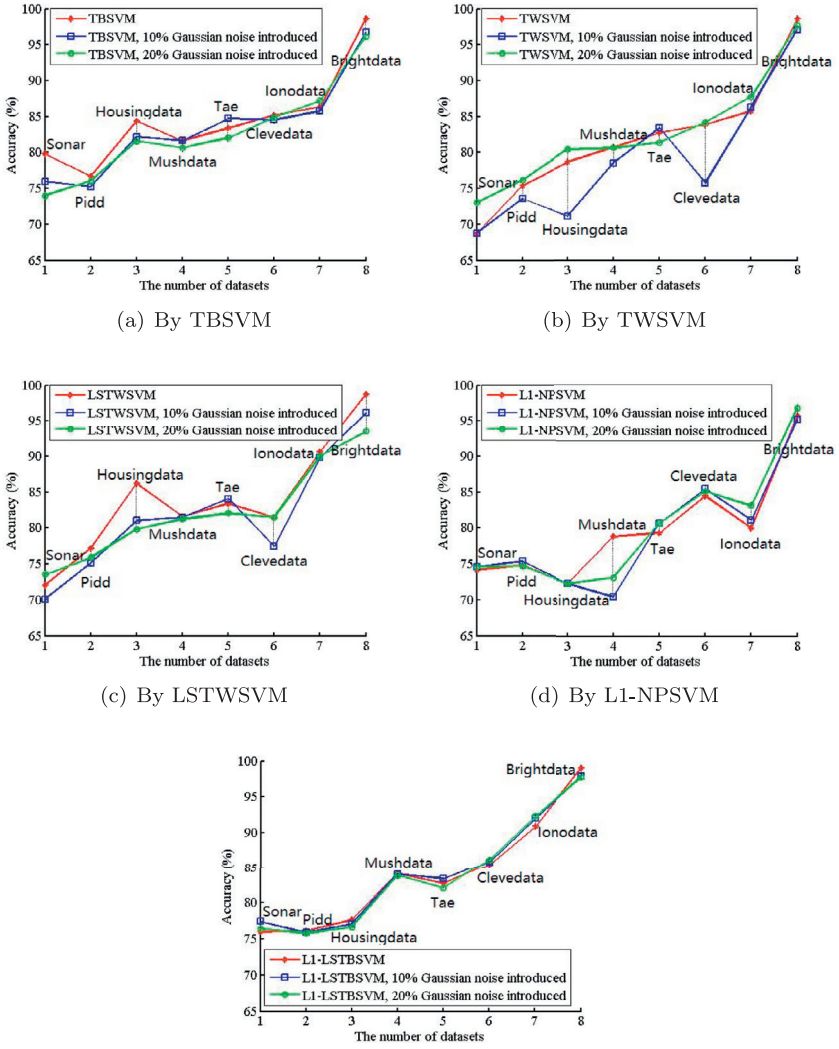


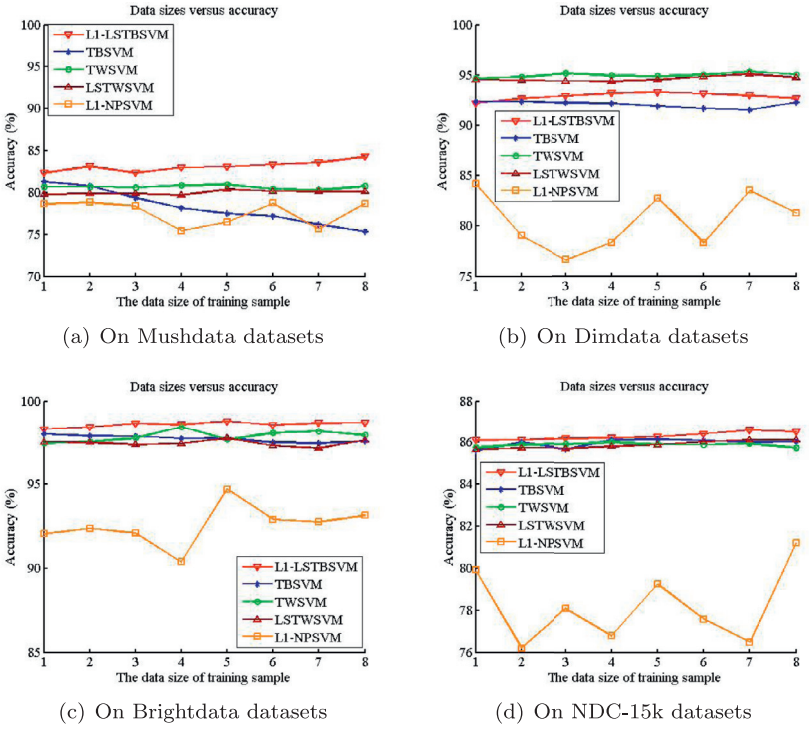
Fig. 5. Comparison between L1-LSTBSVM and other algorithms on eight datasets w.r.t classification performance.

less, our L1-LSTBSVM outperforms TBSVM and TWSVM in training time and classification accuracy. Moreover, it costs more time than LSTWSVM and L1-NPSVM, but surpasses them in accuracy, so the iterative algorithm is useful to dispose large datasets.

Note that, when the data size is increased, the training time of four algorithms (TWSVM, TBSVM, L1-NPSVM and L1-TBSVM) increase significantly, especially TWSVM and TBSVM, while the training time of LSTWSVM does not increase a lot with the change of data size. The reason is that TWSVM and TBSVM need to solve two QPPs respectively, but LSTWSVM just needs to solve one system of two linear equations, so does L1-LSTBSVM. With the increase in data size, the computational costs of TWSVM and TBSVM also increase markedly, and the computational costs of LSTWSVM increase correspondingly. As L1-LSTBSVM is an iterative algorithm (the same to L1-NPSVM) which requires to iteratively search the optimal solutions. Furthermore, in each iteration, it needs to compute the diagonal matrix $\mathbf{D}_1$, whose time complexity is $m_1 \times (d+1)$. Although the time cost is very low, it is also affected by the number of samples in class 1. Therefore, when the data size is increased, the training time of the proposed method increased. Despite this, our algorithm performs faster than TWSVM and TBSVM. The reasons for this are that: 1) in each iteration, as in LSTWSVM, only a system of linear equations is solved; 2) the iterative algorithm can fast converge; and 3) the time complexity of computing $\mathbf{D}_1$ is very low, although it is somewhat increasing with the increase of sample size.

More importantly, in performance, our algorithm has absolutely advantages over other methods. Comparing the experimental results (three tables above), we have the following interesting observations. First, no matter whether the Gaussian noise is introduced or not, L1-LSTBSVM obtains better accuracy compared to other algorithms in most cases. Second, we have found that after introducing Gaussian noise on the UCI datasets (10%, 20% respectively), with the increase of noise, the accuracy of TWSVM, TBSVM and LSTWSVM have larger changes compared with L1-LSTBSVM and L1-NPSVM in most cases. For example, on Tae datasets, the accuracy of TBSVM changes 1.31% and 2.67% when introducing 0%–10% and 10%–20% Gaussian noise, while the accuracy of L1-LSTBSVM changes 0.65% and 1.31% under the same scenario. Similar pattern can be observed for the other algorithms as well. This important observation clearly indicates the robustness of L1-LSTBSVM against outliers and effectively confirms our motivation to use the L1-norm distance to improve the distance metric learning.

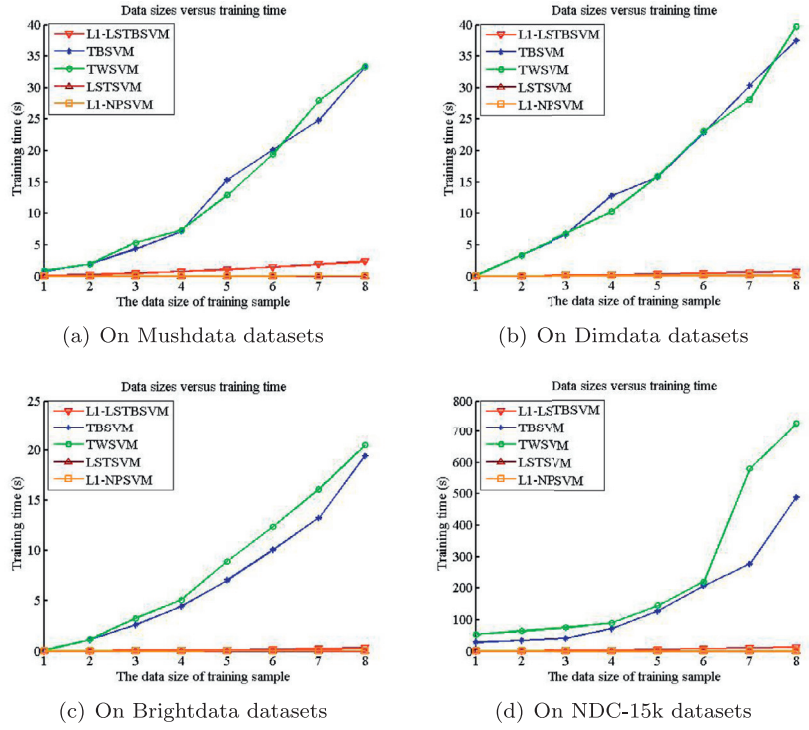
Fig. 5 illustrates the comparison of accuracy of five different algorithms on eight datasets (Sonar, Pidd, Housingdata, Mushdata,



(a) On Mushdata datasets (b) On Dimdata datasets

(c) On Brightdata datasets (d) On NDC-15k datasets

Fig. 6. Performance comparison of five classification algorithms w.r.t different data sizes N versus accuracy.



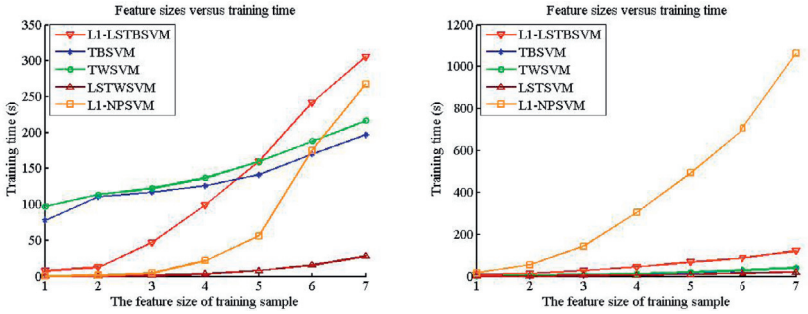
(a) On Mushdata datasets (b) On Dimdata datasets

(c) On Brightdata datasets (d) On NDC-15k datasets

Fig. 7. Performance comparison of five classification algorithms w.r.t different data sizes N versus training time.

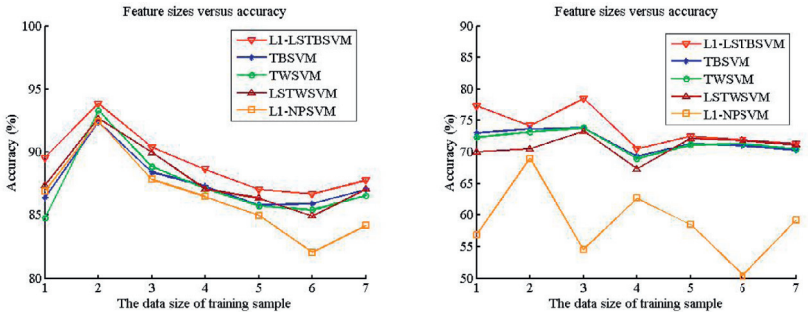
Tae, Clevedata, Ionodata and Brightdata) when introducing 0%, 10% and 20% Gaussian noise respectively. We see that the accuracy of L1-LSTBSVM and L1-NPSVM have a little change compared with that of TBSVM, TWSVM and LSTWSVM in most cases, which indicates L1-norm distance may ease the negative influence of outliers. Further, the utilization of L1-norm distance makes the model of L1-LSTBSVM stronger. Compared to other algorithms, the anti-noise performance of L1-LSTBSVM is best, which provides one more concrete evidence to support the validity of the L1-norm distance in metric learning and attest the correctness of the proposed method.

To compare the performance of the five different algorithms on the datasets with different data sizes N and feature sizes n versus accuracy or training time. We did the experiments on three relative large datasets (Mushdata, Dimdata and Brightdata) and NDC datasets, which were generated using David Musicant's NDC



(a) On the datasets in Table 4 (b) On the datasets in Table 5

Fig. 8. Performance comparison of five classification algorithms w.r.t different feature sizes n versus training time.



(a) On the datasets in Table 4 (b) On the datasets in Table 5

Fig. 9. Performance comparison of five classification algorithms w.r.t different feature sizes n versus accuracy.

Data Generator [35]. The detailed description of NDC datasets are exhibited in Tables 4 and 5.

To compare the accuracy and training time of the five different algorithms w.r.t different data sizes. We did the experiments on four datasets (Mushdata, Dimdata, Brightdata and NDC-15k). We randomly obtain the component of 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90% of each datasets as the training sample respectively, which corresponds to the coordinates on the horizontal axis in Figs. 6 and 7 separately, and the remaining as the test sample. We report the average value (10 times) of classification accuracy and training time as the final result, as shown in Figs. 6 and 7 separately.

Figs. 6 and 7 show the performance comparison (accuracy and training time) of L1-LSTBSVM and relevant algorithms on four datasets about different data sizes. From Fig. 6 we find the accuracy of our method is comparable even the best among all the algorithms, particularly in Fig. 6(a), the accuracy of L1-LSTBSVM is much higher than other algorithms. Besides, in Fig. 6(b), the accuracy of L1-LSTBSVM is not the best among five algorithms, but it is better than that of TBSVM. This may be attributed to the L1-norm distance embedded in LSTBSVM. In short, the proposed method is consistently better than other compared methods in Fig. 6, which demonstrates that L1-LSTBSVM is able to improve the classification performance and is useful for data classification.

Fig. 7(a)–(d) describe different data sizes N versus training time, where N represents the number of samples. From Fig. 7 we can see that L1-LSTBSVM requires significantly less training time than that of TBSVM and TWSVM, which firmly demonstrates the computational advantage of L1-LSTBSVM, but the training time of LSTWSVM is far less than that of L1-LSTBSVM, and L1-NPSVM has a slightly less training time than L1-LSTBSVM. This may be attributed to the least squares sense embedded in TBSVM model and indicates the validity of proposed method, especially in dealing with large datasets.

Table 4
The NDC datasets with data sizes N much larger than feature sizes n.

Datasets	Training data	Testing data	Features
NDC-15k	15,000	1500	32
NDC-10k-50	10,000	2000	50
NDC-10k-100	10,000	2000	100
NDC-10k-400	10,000	2000	400
NDC-10k-700	10,000	2000	700
NDC-10k-1000	10,000	2000	1000
NDC-10k-1500	10,000	2000	1500
NDC-10k-2000	10,000	2000	2000

Table 5
The NDC datasets with feature sizes n larger than data sizes N.

Datasets	Training data	Testing data	Features
NDC-500-1000	500	100	1000
NDC-500-1500	500	100	1500
NDC-500-2000	500	100	2000
NDC-500-2500	500	100	2500
NDC-500-3000	500	100	3000
NDC-500-3500	500	100	3500
NDC-500-4000	500	100	4000

To further assess the performance of L1-LSTBSVM, we also did experiments on the datasets in Tables 4 and 5 respectively. Fig. 8 depicts different feature sizes n versus training time on seven datasets, where their feature size increases in turn and n represents the number of features. From Fig. 8(a) we discover the training time of our method is superior to the competing algo-

rithms (TBSVM and TWSVM) earlier, but later the training time of L1-LSTBSVM is the highest, this may ascribe to the iterative algorithm which we develop. It needs to iteratively compute the optimal solution, which involves the transpose operation of the matrices. When the feature sizes n of the training sample is relatively large, it takes a long time to do these transpose operations. From Fig. 8(b) we can see that the training time of L1-NPSVM and L1-LSTBSVN is the highest, but the training time of L1-NPSVM is much higher than that of L1-LSTBSVN, and TWSVM has almost the same training time as TBSVM, whose training time is higher than that of LSTWSVM. Combining Fig. 8(a) and (b), we find they have similar experimental results, so Fig. 8(b) further indicates L1-LSTBSVM may be detrimental to handling the samples with large feature sizes.

Fig. 9 shows the different feature sizes n versus accuracy on seven datasets. The accuracy of L1-LSTBSVM is the highest in Fig. 9(a), while the accuracy of L1-LSTBSVM is the highest earlier, and later is comparable to that of TWSVM, TBSVM and LSTWSVM, but is higher than that of L1-NPSVM in Fig. 9(b). Combining Figs. 6 and 9, we find L1-LSTBSVM has comparable and even better accuracy than other competing algorithms in most cases, which further demonstrates the classification performance of L1-LSTBSVM is better.

5. Conclusions

In this paper, we have enhanced TBSVM to LSTBSVM in least squares sense, while the distance in LSTBSVM is measured by L1-norm, termed as L1-LSTBSVM, which not only is more robust in the presence of outliers but can also lower the computational time costs and improve the classification performance. The optimization goal of L1-LSTBSVM is to minimize the intra-class distance dispersion, and maximize the inter-class distance dispersion simultaneously. In L1-LSTBSVM, we solve a pair of primal problems of TBSVM by utilizing the idea of LSTWSVM. Compared to TBSVM, which demands solving two QPPs, L1-LSTBSVM requires the solution of two systems of linear equations for linear cases. Further, we develop a simple and valid iterative algorithm to solve the L1-norm optimal problems, which is easy to implement and its convergence to a reasonable optimum solution is theoretically ensured. Thus we can obtain the non-parallel optimal planes. To sum up, the classification performance of L1-LSTBSVM is stronger, especially when Gaussian noise is introduced. Finally, the effectiveness and robustness of L1-LSTBSVM is confirmed by extensive experiments.

There are several interesting directions to research in the future. First, how to further reduce the computational time of L1-LSTBSVM, which makes it effectively deal with large samples is under our consideration. Second, L1-LSTBSVM is only effective for binary classification problem to date, so we would like to extend L1-LSTBSVM to multi-category classification, which is a promising area to study the application of multi-class L1-LSTBSVM in real world. Finally, it is possible to extend this work to a kernel L1-LSTBSVM version to handle the nonlinear tasks.

Acknowledgements

The work is supported in part by the National Science Foundation of China under Grants 61401214, 61773210, 61603184, 61603190 and 61772273, the Natural Science Foundation of Jiangsu Province under Grants BK20171453, BK20140794, the Jiangsu Key Laboratory for Internet of Things and Mobile Internet Technology, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety.

References

[1] X.J. Liu, S.C. Chen, H.J. Peng, Computer keystroker verification based on support vector machines, J. Comput. Res. Dev. 39 (9) (2002) 1082–1086.
[2] S.F. Tian, H.K. Huang, Database learning algorithms based on support vector machine, J. Comput. Res. Dev. 37 (2000) 17–22.
[3] N. Deng, Y. Tian, C. Zhang, Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions, Chapman and Hall/CRC, 2012.
[4] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of Annual Acm Workshop on Computational Learning Theory, 5, 1996, pp. 144–152.
[5] O.L. Mangasarian, E.W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 69–74.
[6] Platt, C. John, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods, 1999, pp. 185–208.
[7] T. Joachims, Making large-scale support vector machine learning practical, in: Advances in kernel methods, 1999, pp. 169–184.
[8] C.C. Chang, C.J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 389–396.
[9] G. Fung, O.L. Mangasarian, Proximal support vector machine classifiers, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 77–86.
[10] Y.H. Shao, W.J. Chen, N.Y. Deng, Nonparallel hyperplane support vector machine for binary classification problems, Inf. Sci. Int. J. 263 (3) (2014) 22–35.
[11] Q. Ye, N. Ye, Improved proximal support vector machine via generalized eigenvalues, in: International Joint Conference on Computational Sciences and Optimization, 2009, pp. 705–709.
[12] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[13] Q. Ye, C. Zhao, X. Chen, A feature selection method for twsvm via a regularization technique, J. Comput. Res. Dev. 48 (6) (2011) 1029–1037.
[14] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.
[15] M. Arun Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (4) (2009) 7535–7543.
[16] D. Tomar, S. Agarwal, Multiclass least squares twin support vector machine for pattern classification, Int. J. Database Theory Appl. 8 (6) (2015) 285–302.
[17] Q. Ye, C. Zhao, S. Gao, H. Zheng, Weighted twin support vector machines with local information and its application, Neural Netw. Official J. Int. Neural Netw. Soc. 35 (11) (2012) 31–39.
[18] X. Zhang, L. Fan, Application of smoothing technique on projective tsvm, Int. J. Appl. Math. Mach. Learn. 2 (2015) 27–45.
[19] N. Kwak, Principal component analysis based on l1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680.
[20] C.N. Li, Y.H. Shao, N.Y. Deng, Robust l1-norm non-parallel proximal support vector machine, Optimization 65 (1) (2015) 1–15.
[21] G. Lin, N. Tang, H. Wang, Locally principal component analysis based on l1-norm maximisation, Image Process. Iet 9 (2) (2015) 91–96.
[22] H. Wang, X. Lu, Z. Hu, W. Zheng, Fisher discriminant analysis with l1-norm, IEEE Trans. Cybern. 44 (6) (2013) 828–842.
[23] F. Zhong, J. Zhang, Linear discriminant analysis based on l1-norm maximization, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 22 (8) (2013) 3018–3027.
[24] Q. Ye, J. Yang, F. Liu, C. Zhao, N. Ye, T. Yin, L1-norm distance linear discriminant analysis based on an effective iterative algorithm, IEEE Trans. Circuits Syst. Video Technol. PP (99) (2016) 1–14.
[25] Q. Ye, H. Zhao, X. Yang, N. Ye, L1-norm distance minimization based fast robusttwin support vector k-plane clustering, IEEE Trans. Neural Netw. Learn. Syst. (2017) 1–10.
[26] C.N. Li, Y.H. Shao, N.Y. Deng, Robust l1-norm two-dimensional linear discriminant analysis, Neural Netw. Official J. Int. Neural Netw. Soc. 65 (C) (2015) 92–104.
[27] S. Gao, Q. Ye, N. Ye, 1-norm least squares twin support vector machines, Neurocomputing 74 (17) (2011) 3590–3597.
[28] O.L. Mangasarian, D.R. Musicant, Lagrangian support vector machines, J. Mach. Learn. Res. 1 (3) (2001) 161–177.
[29] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint l2,1-norms minimization, in: Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 1813–1821.
[30] X. Chen, J. Yang, Q. Ye, J. Liang, Recursive projection twin support vector machine via within-class variance minimization, Pattern Recognit. 44 (10) (2011) 2643–2655.
[31] K. Bache, M. Lichman, Uci machine learning repository (2013).
[32] X. Yang, S. Chen, B. Chen, Z. Pan, Proximal support vector machine using local information, Neurocomputing 73 (1–3) (2009) 357–365.
[33] H. Xue, S. Chen, Glocalization pursuit support vector machine, Neural Comput. Appl. 20 (7) (2011) 1043–1053.
[34] S. Ding, X. Hua, J. Yu, An overview on nonparallel hyperplane support vector machine algorithms, Neural Comput. Appl. 25 (5) (2014) 975–982.
[35] D.R. Musicant, Ndc:normally distributed clustered datasets (1998).

He Yan born in 1988. Master student in College of Information Science & Technology, Nanjing Forestry University, Jiangsu, China. His main research interests include pattern recognition, machine learning and data mining.

Qiaolin Ye received the BS degree in Computer Science from Nanjing Institute of Technology, Nanjing, China, in 2007, the MS degree in Computer Science and Technology from Nanjing Forestry University, Jiangsu, China, in 2009, and the Ph.D. degree in pattern recognition and intelligence system from Nanjing University of Science and Technology, Jiangsu, China, in 2013. He is currently an associate professor with the computer science department at the Nanjing Forestry University, Nanjing, China. He has authored more than 50 scientific papers. Some of them are published in IEEE TNNLS, IEEE TIFS, and IEEE TCSVT. His research interests include machine learning, data mining, and pattern recognition.

Tian'an Zhang born in 1992. Ph.D. student in College of Ecology and Environment, Nanjing Forestry University. Her main research interests include ecology, machine learning, image processing and data mining.

Dong–jun Yu received the B.S. degree in computer science and the MS degree in artificial intelligence from Jiangsu University of Science and Technology in 1997and 2000, respectively, and the Ph.D. degree in pattern analysis and machine intelligence from Nanjing University of Science and Technology in 2003. In 2008, he acted as an academic visitor at University of York in UK. He is currently an associate professor in the School of Computer Science and Engineering of Nanjing University of Science and Technology. His current interests include pattern recognition, data mining and bioinformatics.

Xia Yuan Professor of Nanjing University of Science and Technology. His main research interests include pattern recognition, image processing, machine learning and data mining.

Yiqing Xu Ph.D. of Southeast University. Teaching in College of Information Science & Technology, Nanjing Forestry University, Jiangsu, China. His main research interests include bioinformatics, pattern recognition, machine learning and data mining.

Liyong Fu Professor and master supervisor in Forest Resource Information Techniques in Chinese Academy of Forestry. His main research interests include statistical diagnosis, data mining, and forestry statistical analysis.