Machine Learning
https://doi.org/10.1007/s10994-018-5771-8

Efficient and robust TWSVM classification via a minimum L1-norm distance metric criterion

He Yan[1] · **Qiao-Lin Ye**[2] · **Dong-Jun Yu**[1]

Received: 4 December 2017 / Accepted: 7 November 2018

Abstract
A twin support vector machine (TWSVM) is a classic distance metric learning method for classification problems. The TWSVM criterion is formulated based on the squared L2-norm distance, making it prone to being influenced by the presence of outliers. In this paper, to develop a robust distance metric learning method, we propose a new objective function, called L1-TWSVM, for the TWSVM classifier using the robust L1-norm distance metric. The optimization strategy is to maximize the ratio of the inter-class distance dispersion to the intra-class distance dispersion by using the robust L1-norm distance rather than the traditional L2-norm distance. The resulting objective function is much more challenging to optimize because it involves a non-smooth L1-norm term. As an important contribution of this paper, we design a simple but valid iterative algorithm for solving L1-norm optimal problems. This algorithm is easy to implement, and its convergence to an optimum is theoretically guaranteed. The efficiency and robustness of L1-TWSVM have been validated by extensive experiments on both UCI datasets as well as synthetic datasets. The promising experimental results indicate that our proposal approaches outperform relevant state-of-the-art methods in all kinds of experimental settings.

Keywords L1-norm distance · L1-TWSVM · L2-norm distance · Outliers · TWSVM

1 Introduction

A support vector machine (SVM) (Bradley and Mangasarian 2000; Cortes and Vapnik 1995; Liu et al. 2002; Tian and Huang 2000; Vapnik 1995) plays a critical role in data classification and regression analysis. It operates under the constraint that two support planes are parallel, and the maximum interval classification is implemented by solving quadratic programming

Editor: Ulf Brefeld.

✉ Dong-Jun Yu
njyudj@njust.edu.cn

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing 210094, People's Republic of China

[2] College of Information Science & Technology, Nanjing Forestry University, No. 159 Longpan Road, Nanjing 210037, People's Republic of China

Published online: 15 November 2018

problems (QPPs). Based on structural risk minimization and Vapnik–Chervonenkis dimension, SVM has good generalization performance. SVM has been extensively used in many practical problems, such as image classification (Song et al. 2002), scene classification (Yin et al. 2015), fault diagnosis (Muralidharan et al. 2014) and bioinformatics (Subasi 2013).

The advantages of SVM are remarkable, but in some cases, deep learning and shallow approaches such as random forests give competitive results compared to SVM on several application domains. In particular, the performance of deep learning may be superior to that of SVM when handling large samples, and SVM may cost more computational burden than random forests on the same samples, so SVM still has a lot of space for improvement. There are two main shortcomings: exclusive OR (XOR) problems cannot be handled smoothly (Mangasarian and Wild 2006), and QPPs suffer from high computational complexity (Chang and Lin 2011; Deng et al. 2012). To alleviate these problems, Mangasarian and Wild proposed a proximal support vector machine via generalized eigenvalues (GEPSVM) based on the concept of proximal support vector machine (PSVM) (Fung and Mangasarian 2001) for binary classification problems (Mangasarian and Wild 2006). According to the geometric interpretation of GEPSVM, the numerator should be as small as possible, while the denominator should be as large as possible to minimize the objective function value. GEPSVM relaxes the requirement of PSVM that the planes be parallel and can solve XOR problem smoothly. Moreover, GEPSVM attempts to find two nonparallel planes by solving a pair of generalized eigenvalue problems instead of complex QPPs, which can reduce the computation time and improve the generalization ability over that of PSVM (Mangasarian and Wild 2006). The advantages of GEPSVM play an important role in this improvement (Guarracino et al. 2007; Shao et al. 2013, 2014; Ye and Ye 2009). However, it should be noted that GEPSVM and its variants are sensitive to outliers because the L2-norm distance exaggerates the effect of outliers by the square operation (Kwak 2008), which reduces the classification performance. Outliers are defined as the data points that deviate significantly from the majority of the data points or those do not have a regular distribution over the data points (Wang et al. 2014b). In view of this limitation, many researches have been carried out to improve the robustness of machine learning models by using the L1-norm distance (Gao et al. 2011; Li et al. 2015a, b; Wang et al. 2014a, b; Ye et al. 2016, 2017). To promote the robustness, Li et al. (2015a) reformulated the optimization problems of a nonparallel proximal support vector machine using the L1-norm distance (L1-NPSVM). To solve the formulated objective, a gradient ascent (GA) iterative algorithm is proposed, which is simple to execute but may not guarantee the optimality of the solution due to both the need of introducing a non-convex surrogate function and the difficulty in selecting the step-size (Kwak 2014). Wang et al. (2014a) optimized Fisher linear discriminant analysis (LDA) by taking advantage of the L1-norm distance instead of the conventional L2-norm distance; this optimized LDA is denoted as LDA-L1. The utilization of the L1-norm distance makes LDA-L1 robust to outliers, and LDA-L1 does not suffer from the problems of small sample size and rank limit that existed in the traditional LDA. Nevertheless, in LDA-L1, a gradient ascent iterative algorithm is applied, which suffers from the difficulty in choosing the step-size.

As a successful improvement of GEPSVM, Jayadeva et al. proposed a twin support vector machine (Jayadeva and Chandra 2007)_(TWSVM) based on the concept of GEPSVM. TWSVM solves two QPPs (the scale is relatively small compared to that of standard SVM) to replace generalized eigenvalue problems (Mangasarian and Wild 2006). As TWSVM inherits the advantages of GEPSVM, it can handle the XOR problem smoothly. At present, the research on TWSVM is still in its infancy, and many improved methods have been developed based on the concept of TWSVM, such as smooth TWSVM (Kumar and Gopal 2008), localized TWSVM (LCTSVM) (Ye et al. 2011b), twin bounded SVM (TBSVM) (Shao et al.

2011), and robust TWSVM (R-TWSVM) (Qi et al. 2013a). Ye et al. (2011a) introduced a regularization technique for optimizing TWSVM and proposed a feature selection method for TWSVM via the regularization technique (RTWSVM), which is a convex programming problem, to overcome the possible singular problem and improve the generalization ability. Kumar and Gopal (2009) reformulated the optimization problems of TWSVM by using constraints in the form of equalities to replace inequalities to modify the primal QPPs in least-squares sense and proposed a least-squares version of TWSVM (LSTSVM). The solutions of LSTSVM follow directly from solving two linear equations, as opposed to solving two QPPs. Therefore, LSTSVM effectively addresses large samples without any external optimization. Moreover, its computational cost is much lower than that of TWSVM. Qi et al. (2013b) optimized TWSVM by applying the structural information of data, which may contain useful prior domain knowledge for training the classifier, and proposed a new structural TWSVM (S-TWSVM). S-TWSVM utilizes two hyperplanes to decide the category of new data, and each model only considers the structural information of one class. Each plane is closer to one of the two classes and as far away as possible from the other class. This allows S-TWSVM to fully exploit the prior knowledge to directly improve its generalization ability.

It is worth noting that TWSVM and its variants are also sensitive to outliers. L1-norm distance is more robust to outliers than the squared L2-norm distance in distance metric learning (Cayton and Dasgupta 2006; Ke and Kanade 2005; Li et al. 2015a; Lin et al. 2015; Pang et al. 2010; Wang et al. 2012, 2014a; Zhong and Zhang 2013). The utilization of the L1-norm distance is considered to be a simple and effective way to reduce the impact of outliers (Li et al. 2015b; Wang et al. 2014a) and can improve the generalization ability and flexibility of the model, as with L1-NPSVM and LDA-L1. Following the same motivations as these prior studies, we propose replacing the squared L2-norm distance in TWSVM with the robust L1-norm distance to improve the robustness; the resulting TWSVM is called L1-TWSVM. L1-TWSVM seeks two nonparallel optimal planes by solving two QPPs. The optimization goal of L1-TWSVM is to minimize the intra-class distance and maximize the inter-class distance simultaneously. Moreover, L1-TWSVM seamlessly integrates the merits of TWSVM with those of the robust L1-norm-based distance metric, which improves the classification performance and robustness. In summary, this paper makes the following contributions: (1) An iterative algorithm is presented to solve L1-norm distance optimization problems. The iterative optimization technique is simple and convenient to implement. We theoretically prove that the objective function value of L1-TWSVM is reduced at each step of iteration. This means that the convergence of the iterative algorithm to a local optimal solution is theoretically guaranteed. (2) In L1-TWSVM, the conventional L2-norm distance is replaced by more robust L1-norm distance to reduce the effect of outliers, which makes L1-TWSVM robust to outliers. L1-TWSVM can efficiently decrease the impact of the outliers, even if the ratio of outliers is large. (3) The proposed method is evaluated with relevant algorithms (SVM, GEPSVM, TWSVM, LSTSVM and L1-NPSVM) on both synthetic datasets and UCI datasets. Extensive experimental results confirm that L1-TWSVM and L1-NPSVM effectively reduce the effect of the outliers, which improves the generalization ability and flexibility of the model. (4) The proposed method can be conveniently extended to solve other improved methods based on TWSVM.

The remainder of this paper is organized as follows. Section 2 briefly introduces GEPSVM and TWSVM. Section 3 proposes L1-TWSVM, discusses its feasibility and presents the theoretical analysis. All the experimental results are shown in Sect. 4, and conclusions are presented in Sect. 5.

2 Related works

In this paper, all vectors are column vectors unless a superscript T is present, which denotes transposition. We use bold uppercase letters to represent matrices and bold lowercase ones to represent vectors. The vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ of appropriate lengths are represented by identity column vectors. Furthermore, $\mathbf{I}$ denotes an identity matrix of appropriate dimension. We consider a binary classification problem in the n-dimensional real space R^n, and the dataset is denoted by $\mathbf{T} = \left\{\left(\mathbf{x}_j^{(i)}, y_i\right) | i = 1, 2,\ j = 1, 2, \ldots, m_i\right\}$, where $\mathbf{x}_j^{(i)} \in R^n$ and $y_i \in \{-1,\ 1\}$, and $\mathbf{x}_j^{(i)}$ denotes the i-th class and j-th sample. We suppose that matrix $\mathbf{A} = \left(\mathbf{a}_1^{(1)}, \mathbf{a}_2^{(1)}, \ldots, \mathbf{a}_{m_1}^{(1)}\right)^T$ of size $m_1 \times n$ represents the data points of Class 1 (Class +1), while matrix $\mathbf{B} = \left(\mathbf{b}_1^{(2)}, \mathbf{b}_2^{(2)}, \ldots, \mathbf{b}_{m_2}^{(2)}\right)^T$ of size of $m_2 \times n$ represents the data points of Class 2 (Class -1), where matrices $\mathbf{A}$ and $\mathbf{B}$ represent all the data points, m_1 represents the number of positive class samples, m_2 represents the number of negative class samples, and $m_1 + m_2 = m$. In the following, we review two well-known nonparallel proximal classifiers: GEPSVM (Mangasarian and Wild 2006) and TWSVM (Jayadeva and Chandra 2007).

2.1 GEPSVM

GEPSVM is an excellent classifier for binary classification problems and is widely used for pattern classification problems. The primary aim of GEPSVM is to find two nonparallel proximal planes

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

where $\mathbf{w}_1,\ \mathbf{w}_2 \in R^n$ and $b_1,\ b_2 \in R$. The geometric interpretation of GEPSVM is that each plane is closer to one of the two classes and as far away as possible from the other class. This produces the following two optimization problems of GEPSVM:

$$\min_{\mathbf{w}_1, b_1} \frac{||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 + \delta||\left(\mathbf{w}_1^T\ b_1\right)^T||_2^2}{||\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1||_2^2} \tag{2}$$

$$\min_{\mathbf{w}_2, b_2} \frac{||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2 + \delta||\left(\mathbf{w}_2^T\ b_2\right)^T||_2^2}{||\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2||_2^2} \tag{3}$$

where $||\cdot||_2$ denotes the L2-norm, $\delta||\left(\mathbf{w}_1^T\ b_1\right)^T||_2^2$ is a Tikhonov regularization term, and δ is a regularization factor. The regularization terms are introduced to address the singular problem when solving the generalized eigenvalue problems, which can improve the stability of GEPSVM. Then, optimization problems (2) and (3) become

$$\min_{\mathbf{z}_1} \frac{\mathbf{z}_1^T\mathbf{E}\mathbf{z}_1}{\mathbf{z}_1^T\mathbf{F}\mathbf{z}_1} \tag{4}$$

$$\min_{\mathbf{z}_2} \frac{\mathbf{z}_2^T\mathbf{L}\mathbf{z}_2}{\mathbf{z}_2^T\mathbf{M}\mathbf{z}_2} \tag{5}$$

where $\mathbf{H} = (\mathbf{A}\ \mathbf{e}_1)$, $\mathbf{G} = (\mathbf{B}\ \mathbf{e}_2)$ are matrices and $\mathbf{z}_1 = \left(\mathbf{w}_1^T\ b_1\right)^T$, $\mathbf{z}_2 = \left(\mathbf{w}_2^T\ b_2\right)^T$ are augmented vectors, $\mathbf{E} = \mathbf{H}^T\mathbf{H} + \delta\mathbf{I}$, $\mathbf{L} = \mathbf{G}^T\mathbf{G} + \delta\mathbf{I}$, $\mathbf{F} = \mathbf{G}^T\mathbf{G}$, and $\mathbf{M} = \mathbf{H}^T\mathbf{H}$.

$\mathbf{E}$, $\mathbf{F}$ and $\mathbf{L}$, $\mathbf{M}$ are symmetric matrices in $R^{(n+1)\times(n+1)}$. The objective functions in (4) and (5) are Rayleigh quotient problems (Parlett 1998) and have some very useful properties,

as we now state. It is easy to derive the solutions of (4) and (5) by solving the generalized eigenvalue problems

$$\mathbf{E}\mathbf{z}_1 = \lambda_1 \mathbf{F}\mathbf{z}_1,\ \mathbf{z}_1 \neq 0 \tag{6}$$

$$\mathbf{L}\mathbf{z}_2 = \lambda_2 \mathbf{M}\mathbf{z}_2,\ \mathbf{z}_2 \neq 0 \tag{7}$$

where the minimum of (4) is attained at an eigenvector corresponding to the smallest eigenvalue λ_1 of (6). Consequently, if $\mathbf{z}_1$ denotes the eigenvector corresponding to λ_1, then the augmented vector $\mathbf{z}_1 = \left(\mathbf{w}_1^T\ \mathbf{b}_1\right)^T$ determines the plane $\mathbf{x}^T\mathbf{w}_1 + b_1 = 0$, which is close to data points of Class 1. Similarly, the augmented vector $\mathbf{z}_2 = \left(\mathbf{w}_2^T\ \mathbf{b}_2\right)^T$ determines the plane $\mathbf{x}^T\mathbf{w}_2 + b_2 = 0$, which is close to data points of Class 2.

2.2 TWSVM

In this section, after a brief review of GEPSVM, we introduce TWSVM, which is an improved version of GEPSVM. To obtain two planes, TWSVM solves two convex programming problems rather than solving a system of two linear equations as GEPSVM does (Mangasarian and Wild 2006). The two objective functions of TWSVM are expressed as follows:

$$\begin{aligned} &\min_{\mathbf{w}_1, b_1} \frac{1}{2}||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 + c_1 \mathbf{e}_2^T \mathbf{q}_1 \\ &s.t.\ -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 \geq \mathbf{e}_2,\ \mathbf{q}_1 \geq 0 \end{aligned} \tag{8}$$

$$\begin{aligned} &\min_{\mathbf{w}_2, b_2} \frac{1}{2}||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2 + c_2 \mathbf{e}_1^T \mathbf{q}_2 \\ &s.t.\ (\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2) + \mathbf{q}_2 \geq \mathbf{e}_1,\ \mathbf{q}_2 \geq 0 \end{aligned} \tag{9}$$

where $||\cdot||_2$ denotes the L2-norm, $\mathbf{q}_1$ and $\mathbf{q}_2$ are slack vectors, and c_1 and c_2 are nonnegative penalty coefficients, which are the balance factors of the positive and negative samples, respectively, and can overcome the problem of sample imbalance in TWSVM. It should be noted that in TWSVM, the distance is measured by the L2-norm, which is likely to exaggerate the effect of outliers by the square operation. The optimization strategy of TWSVM is that points of the same class are clustered as compactly as possible and are as far as possible from data in the other class, which guarantees the minimization of the objective function. By solving formulas (8) and (9), we can obtain two nonparallel planes:

$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0,\ \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{10}$$

A new data point $\mathbf{x}$ is assigned to Class 1 or Class 2 depending on its proximity to each of the two nonparallel planes. We can obtain the corresponding Wolfe dual problems of formulas (8) and (9):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \mathbf{e}_2^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ &s.t.\ 0 \leq \boldsymbol{\alpha} \leq c_1\mathbf{e}_2 \end{aligned} \tag{11}$$

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \mathbf{e}_1^T\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \\ &s.t.\ 0 \leq \boldsymbol{\beta} \leq c_2\mathbf{e}_1 \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha} \in R^{m_2}$ and $\boldsymbol{\beta} \in R^{m_1}$ are Lagrange multipliers, we can derive two nonparallel planes using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$\begin{aligned} \mathbf{z}_1 &= \left(\mathbf{w}_1^T \; b_1\right)^T = -\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ \mathbf{z}_2 &= \left(\mathbf{w}_2^T \; b_2\right)^T = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \end{aligned} \tag{13}$$

Note that the inverse matrices $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}$ and $\left(\mathbf{G}^T\mathbf{G}\right)^{-1}$ in Eq. (13) easily encounter singularity problems. To prevent matrix singularity, the regularization term $\varepsilon\mathbf{I}$, where ε is a positive scalar that is small enough to preserve the structure of the data, is introduced (Jayadeva and Chandra 2007; Mangasarian and Wild 2006). Because $\left(\mathbf{H}^T\mathbf{H} + \varepsilon\mathbf{I}\right)^{-1}$ and $\left(\mathbf{G}^T\mathbf{G} + \varepsilon\mathbf{I}\right)^{-1}$ are positive definite, they do not suffer from singularity problems.

3 Efficient and robust TWSVM based on L1-norm distance

TWSVM has become a hotspot in the research of data classification due to its good classification performance. However, in TWSVM, the distance is measured by the L2-norm. It is well known that the squared L2-norm distance is sensitive to outliers, which implies that abnormal observations may affect the solution obtained by TWSVM. In the literature (Ding et al. 2006; Gao 2008; Kwak 2008; Li et al. 2015a; Nie et al. 2015; Wright et al. 2009), the L1-norm distance is usually considered as a robust alternative to the L2-norm distance for improving the generalization ability and flexibility of the model. Motivated by the basic idea of L1-norm-based modeling, we propose a robust classifier based on the L1-norm distance metric, which replaces the squared L2-norm distance in the distance metric learning objective functions in formulas (8) and (9), thereby leading to the following optimization problems:

$$\begin{aligned} &\min_{\mathbf{w}_1, b_1} \frac{1}{2}||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_1 + c_1\mathbf{e}_2^T\mathbf{q}_1 \\ &s.t. \; -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 \geq \mathbf{e}_2, \; \mathbf{q}_1 \geq 0 \end{aligned} \tag{14}$$

$$\begin{aligned} &\min_{\mathbf{w}_2, b_2} \frac{1}{2}||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 \mathrm{b}_2||_1 + c_2\mathbf{e}_1^T\mathbf{q}_2 \\ &s.t. \; (\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2) + \mathbf{q}_2 \geq \mathbf{e}_1, \; \mathbf{q}_2 \geq 0 \end{aligned} \tag{15}$$

where $||\cdot||_1$ denotes the L1-norm. In a solution that minimizes the objective functions, each plane is as close as possible to one of the two classes and as far as possible from the other class. Because formulas (14) and (15) are convex optimization problems with non-convex constraints in the form of inequalities, they have the local optimal solutions, and we can obtain two nonparallel planes by solving them:

$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0, \; \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{16}$$

The original problems in formulas (14) and (15) can be optimized in the following forms:

$$\begin{aligned} &\min_{\mathbf{w}_1, b_1} \frac{1}{2}\left(\sum_{i=1}^{m_1} \frac{\left(\mathbf{a}_i^T\mathbf{w}_1 + e_1^i b_1\right)^2}{d_i}\right) + c_1\mathbf{e}_2^T\mathbf{q}_1 \\ &s.t. \; -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 \geq \mathbf{e}_2, \; \mathbf{q}_1 \geq 0 \end{aligned} \tag{17}$$

$$\min_{\mathbf{w}_2,b_2} \frac{1}{2}\left(\sum_{j=1}^{m_2} \frac{\left(\mathbf{b}_j^T\mathbf{w}_2 + e_2^j b_2\right)^2}{d_j}\right) + c_2\mathbf{e}_1^T\mathbf{q}_2$$
$$s.t.\ (\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2) + \mathbf{q}_2 \geq \mathbf{e}_1,\ \mathbf{q}_2 \geq 0 \tag{18}$$

where $d_i = \left|\mathbf{a}_i^T\mathbf{w}_1 + e_1^i b_1\right| \neq 0$ and $d_j = \left|\mathbf{b}_j^T\mathbf{w}_2 + e_2^j b_2\right| \neq 0$, e_1^i,e_2^j denote the i-th and j-th element of $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively. It is difficult to directly solve formulas (17) and (18) because they each contain an absolute value operation, which makes the optimization of objective function (17) intractable. To solve these problems, we propose an iterative convex optimization strategy. The basic idea of this method is to iteratively update the augmented vector $\mathbf{z}_1$ until its objective values in (17) of two successive iterations is less than a fixed value (0.001); then, $\mathbf{z}_1$ is the local minimum solution. Assume that $\mathbf{z}_1^p$ is the solution for iteration p. Then, the solution $\mathbf{z}_1^{(p+1)}$ for iteration $p+1$ is defined as the solution to the following problems:

$$\min_{\mathbf{z}_1} \frac{1}{2}\left(\sum_{i=1}^{m_1} \frac{\left(\mathbf{h}_i^T\mathbf{z}_1\right)^2}{d_{1i}}\right) + c_1\mathbf{e}_2^T\mathbf{q}_1$$
$$s.t.\ -\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 \geq \mathbf{e}_2,\ \mathbf{q}_1 \geq 0 \tag{19}$$
$$\min_{\mathbf{z}_2} \frac{1}{2}\left(\sum_{j=1}^{m_2} \frac{\left(\mathbf{g}_j^T\mathbf{z}_2\right)^2}{d_{2j}}\right) + c_2\mathbf{e}_1^T\mathbf{q}_2$$
$$s.t.\ \mathbf{H}\mathbf{z}_2 + \mathbf{q}_2 \geq \mathbf{e}_1,\ \mathbf{q}_2 \geq 0 \tag{20}$$

where $d_{1i} = \left|\mathbf{h}_i^T\mathbf{z}_1^p\right|$, $d_{2\,j} = \left|\mathbf{g}_j^T\mathbf{z}_2^p\right|$, $\mathbf{h}_i^T = \left(\mathbf{a}_i^T\ e_1^i\right)$, and $\mathbf{g}_j^T = \left(\mathbf{b}_j^T\ e_2^j\right)$. Then, formulas (19) and (20) are rewritten as

$$\min_{\mathbf{z}_1} \frac{1}{2}\mathbf{z}_1^T\mathbf{H}^T\mathbf{D}_1\mathbf{H}\mathbf{z}_1 + c_1\mathbf{e}_2^T\mathbf{q}_1$$
$$s.t.\ -\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 \geq \mathbf{e}_2,\ \mathbf{q}_1 \geq 0 \tag{21}$$
$$\min_{\mathbf{z}_2} \frac{1}{2}\mathbf{z}_2^T\mathbf{G}^T\mathbf{D}_2\mathbf{G}\mathbf{z}_2 + c_2\mathbf{e}_1^T\mathbf{q}_2$$
$$s.t.\ \mathbf{H}\mathbf{z}_2 + \mathbf{q}_2 \geq \mathbf{e}_1,\ \mathbf{q}_2 \geq 0 \tag{22}$$

where $\mathbf{D}_1 = diag\left(1/d_{11}, 1/d_{12}, \ldots, 1/d_{1m_1}\right)$ and $\mathbf{D}_2 = diag\left(1/d_{21}, 1/d_{22}, \ldots, 1/d_{2m_2}\right)$ are diagonal matrices.

We rewrite the problems (21) and (22) with the following equivalent formulation,

$$\min_{\mathbf{z}_1} \frac{1}{2}\|\mathbf{H}\mathbf{z}_1\|_1 + c_1\mathbf{e}_2^T\mathbf{q}_1$$
$$s.t.\ -\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 \geq \mathbf{e}_2,\ \mathbf{q}_1 \geq 0 \tag{23}$$
$$\min_{\mathbf{z}_2} \frac{1}{2}\|\mathbf{G}\mathbf{z}_2\|_1 + c_2\mathbf{e}_1^T\mathbf{q}_2$$
$$s.t.\ \mathbf{H}\mathbf{z}_2 + \mathbf{q}_2 \geq \mathbf{e}_1,\ \mathbf{q}_2 \geq 0 \tag{24}$$

Formula (14) is a convex optimization problem with inequality constraints (non-convex); therefore, it has a closed-form solution. The Lagrange function of (14) is constructed to solve this problem:

$$L_1(\mathbf{w}_1, b_1, \mathbf{q}_1, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1)^T \mathbf{D}_1 (\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1) + c_1 \mathbf{e}_2^T \mathbf{q}_1 - \boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 - \mathbf{e}_2) - \boldsymbol{\beta}^T \mathbf{q}_1 \quad (25)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{m_2})^T$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \ldots, \beta_{m_1})^T$ are Lagrange multipliers, and $\boldsymbol{\alpha} \geq 0$, $\boldsymbol{\beta} \geq 0$. The partial derivatives of $\mathbf{w}_1$, b_1 and $\mathbf{q}_1$ are obtained with Lagrange function L_1 separately, and their derivatives are set equal to zero. Then, the Karush–Kuhn—Tucker (KKT) conditions can be obtained:

$$\frac{\partial L}{\partial \mathbf{w}_1} = \mathbf{A}^T \mathbf{D}_1 (\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{B}^T \boldsymbol{\alpha} = 0 \quad (26)$$

$$\frac{\partial L}{\partial b_1} = \mathbf{e}_1^T \mathbf{D}_1 (\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{e}_2^T \boldsymbol{\alpha} = 0 \quad (27)$$

$$\frac{\partial L}{\partial \mathbf{q}_1} = c_1 \mathbf{e}_2 - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0 \quad (28)$$

$$-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 \geq \mathbf{e}_2, \ \mathbf{q}_1 \geq 0 \quad (29)$$

$$\boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_1 - \mathbf{e}_2) = 0, \ \boldsymbol{\beta}^T \mathbf{q}_1 = 0 \quad (30)$$

We can obtain $0 \leq \boldsymbol{\alpha} \leq c_1 \mathbf{e}_2$ from Eq. (28) since $\boldsymbol{\alpha} \geq 0$, $\boldsymbol{\beta} \geq 0$. Next, Eqs. (26) and (27) are combined:

$$\left(\mathbf{A}^T \ \mathbf{e}_1^T\right) \mathbf{D}_1 (\mathbf{A} \ \mathbf{e}_1)(\mathbf{w}_1 \ b_1)^T + \left(\mathbf{B}^T \ \mathbf{e}_2^T\right)\boldsymbol{\alpha} = 0 \quad (31)$$

We have previously defined matrices ($\mathbf{H}$, $\mathbf{G}$) and augmented vectors ($\mathbf{z}_1$, $\mathbf{z}_2$). With these notations, the solution of $\mathbf{z}_1^{(p+1)}$ can be obtained based on the conditions above:

$$\mathbf{H}^T \mathbf{D}_1^p \mathbf{H} \mathbf{z}_1^{(p+1)} + \mathbf{G}^T \boldsymbol{\alpha} = 0 \quad (32)$$

Equation (32) is equivalent to Eq. (33):

$$\mathbf{z}_1^{(P+1)} = -\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H}\right)^{-1} \mathbf{G}^T \boldsymbol{\alpha} \quad (33)$$

In Eq. (33), it is necessary to calculate inverse matrix $\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H}\right)^{-1}$ to obtain $\mathbf{z}_1^{(p+1)}$. $\mathbf{H}^T \mathbf{D}_1^p \mathbf{H}$ is a positive semi-definite matrix that may be ill-conditioned in some situations, so we may obtain an inaccurate or unstable solution. In real applications, we can use the methods described in Jayadeva and Chandra (2007), Mangasarian and Wild (2006). The regularization term is introduced to address this problem, where ε is a small perturbation. $\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H} + \varepsilon \mathbf{I}\right)$ is a positive definite matrix and does not suffer from the singularity problem. Moreover, inverse matrix $\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H}\right)^{-1}$ is approximately replaced by $\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H} + \varepsilon \mathbf{I}\right)^{-1}$. Therefore, we can derive the final solution of $\mathbf{z}_1^{(p+1)}$:

$$\mathbf{z}_1^{(p+1)} = -\left(\mathbf{H}^T \mathbf{D}_1^p \mathbf{H} + \varepsilon \mathbf{I}\right)^{-1} \mathbf{G}^T \boldsymbol{\alpha} \quad (34)$$

Similarly,

$$\mathbf{z}_2^{(p+1)} = \left(\mathbf{G}^T \mathbf{D}_2^p \mathbf{G} + \varepsilon \mathbf{I}\right)^{-1} \mathbf{H}^T \boldsymbol{\beta} \quad (35)$$

Augmented vectors $\mathbf{z}_1^{(p+1)}$ and $\mathbf{z}_2^{(p+1)}$ are substituted into Lagrange function (25) separately. Under KKT conditions, the original optimization problems (14) and (15) can be transformed into Wolfe dual problems:

$$\max_{\boldsymbol{\alpha}} \mathbf{e}_2^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{D}_1\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}$$
$$s.t.\ 0 \le \boldsymbol{\alpha} \le c_1\mathbf{e}_2 \tag{36}$$

$$\max_{\boldsymbol{\beta}} \mathbf{e}_1^T\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{D}_2\mathbf{G}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta}$$
$$s.t.\ 0 \le \boldsymbol{\beta} \le c_2\mathbf{e}_1 \tag{37}$$

We can obtain the Lagrange multipliers $\boldsymbol{\alpha} \in R^{m_2\times 1}$ and $\boldsymbol{\beta} \in R^{m_1\times 1}$ by solving the dual problems and substitute $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ into Eqs. (34) and (35), respectively. In addition, weight vectors $\mathbf{w}_1$, $\mathbf{w}_2$ and deviations b_1, b_2 can be obtained. That is, we acquire two nonparallel planes (16).

A new point $\mathbf{x} \in \mathrm{R}^{\mathrm{n}}$ is assigned to Class 1 or Class 2, according to which of the two nonparallel planes given by (16) lies closest to the decision function

$$f(\mathbf{x}) = \arg\min_{i=1,2}\left(\left|\mathbf{x}^T\mathbf{w}_i + b_i\right| \Big/ \|\mathbf{w}_i\|\right) \tag{38}$$

Here, $|\cdot|$ is the absolute value operation.

The new objective function in (14) is a convex problem with non-convex constraint, so $\mathbf{z}_1^{(p+1)}$ is the local optimal solution to the problem. Note that in Eq. (33), $\mathbf{D}_1^p$ is dependent on $\mathbf{z}_1^{(p+1)}$; thus, it is an unknown variable and can be viewed as the potential variable of the objective in (14), which can be solved using the same iterative algorithm by alternating optimization. We calculate $\mathbf{D}_1^p$ based on the solution $\mathbf{z}_1^{(p+1)}$ that was obtained in the previous iteration and iteratively update $\mathbf{D}_1^p$ to change $\mathbf{z}_1^{(p+1)}$, increase p until the objective values of two successive iterations is less than a fixed value. Besides, proper initialization can effectually expedite the convergence of the algorithm. In practice, we solve formulas (8) and (9) to obtain initial solutions, which empirically works very well in our experiments. The iterative procedure of L1-TWSVM is summarized in Algorithm 1.

Algorithm 1: An efficient iterative algorithm to solve problem (23)

Input: Input matrices $\mathbf{A} \in R^{m_1\times n}$ and $\mathbf{B} \in R^{m_2\times n}$.
Construct the matrices $\mathbf{H} = (\mathbf{A}\ \ \mathbf{e}_1) \in R^{m_1\times(n+1)}$ and $\mathbf{G} = (\mathbf{B}\ \ \mathbf{e}_2) \in R^{m_2\times(n+1)}$.
Set $p = 0$, an iteration number. Initialize $\mathbf{z}_1^p$, a standard solution of TWSVM.
While not converge do
1. Compute the diagonal matrix $\mathbf{D}_1^p \in R^{m_1\times m_1}$, where $\mathbf{D}_1^p = diag\left(1/\left|\mathbf{H}\mathbf{z}_1^p\right|\right)$ and the i-th diagonal element of $\mathbf{D}_1^{(p+1)}$ is $1/\left(\left|\mathbf{h}_i^T\mathbf{z}_1^{(p+1)}\right|\right)$, $\mathbf{h}_i$ denotes the i-th column of $\mathbf{H}$.
2. Compute $\mathbf{z}_1^{(p+1)}$ by solving

$$\mathbf{z}_1^{(p+1)} = \arg\min_{\mathbf{z}_1} \frac{1}{2}\mathbf{z}_1^T\mathbf{H}^T\mathbf{D}_1^p\mathbf{H}\mathbf{z}_1 + c_1\mathbf{e}_2^T\mathbf{q}_1,\ s.t.\ -\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 \ge \mathbf{e}_2,\ \mathbf{q}_1 \ge 0 \tag{39}$$

3. $\mathbf{p} = \mathbf{p} + \mathbf{1}$.
End while
Output: The learned solution of $\mathbf{z}_1$.

Algorithm 1 is an efficient iterative algorithm for solving the optimization problem defined by formula (14), which implies that each updating step decreases the value of the objective function, whose convergence is guaranteed by Theorem 1. To prove this, we first introduce Lemma 1.

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

$$\|\mathbf{u}\|_1 - \frac{\|\mathbf{u}\|_1^2}{2\|\mathbf{u}^p\|_1} \leq \|\mathbf{u}^p\|_1 - \frac{\|\mathbf{u}^p\|_1^2}{2\|\mathbf{u}^p\|_1} \tag{40}$$

Proof Starting with the inequality $\left(\sqrt{\mathbf{v}} - \sqrt{\mathbf{v}^p}\right)^2 \geq 0$, we have

$$\begin{aligned} &\left(\sqrt{\mathbf{v}} - \sqrt{\mathbf{v}^p}\right)^2 \geq 0 \Rightarrow \mathbf{v} - 2\sqrt{\mathbf{v}\mathbf{v}^p} + \mathbf{v}^p \geq 0 \\ &\Rightarrow \sqrt{\mathbf{v}} - \frac{\mathbf{v}}{2\sqrt{\mathbf{v}^p}} \leq \frac{\sqrt{\mathbf{v}^p}}{2} \Rightarrow \sqrt{\mathbf{v}} - \frac{\mathbf{v}}{2\sqrt{\mathbf{v}^p}} \leq \sqrt{\mathbf{v}^p} - \frac{\mathbf{v}^p}{2\sqrt{\mathbf{v}^p}} \end{aligned} \tag{41}$$

By replacing $\mathbf{v}$ and $\mathbf{v}^p$ in (41) with $\|\mathbf{u}\|_1^2$ and $\|\mathbf{u}^p\|_1^2$, respectively, we obtain (40). □

Theorem 1 *Algorithm* 1 *monotonously decreases the objective of problem* (23) *in each iteration*.

Proof First, we rewrite the problem in (39) with the following equivalent formulation:

$$\mathbf{z}_1^{(p+1)} = \arg\min_{\mathbf{z}_1} \frac{1}{2}\mathbf{z}_1^T \mathbf{H}^T \mathbf{D}_1^p \mathbf{H}\mathbf{z}_1 + c_1 \mathbf{e}_2^T \max(0, \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1) \tag{42}$$

That is,

$$\mathbf{z}_1^{(p+1)} = \arg\min_{\mathbf{z}_1} \frac{1}{2}(\mathbf{H}\mathbf{z}_1)^T \mathbf{D}_1^p \mathbf{H}\mathbf{z}_1 + c_1 \mathbf{e}_2^T \max(0, \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1) \tag{43}$$

Thus, in the $(p+1)$-th iteration, according to (39) in Algorithm 1, we have

$$\begin{aligned} &\frac{1}{2}\left(\mathbf{H}\mathbf{z}_1^{(p+1)}\right)^T \mathbf{D}_1^p \left(\mathbf{H}\mathbf{z}_1^{(p+1)}\right) + c_1 \mathbf{e}_2^T \max\left(0,\ \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1^{(p+1)}\right) \\ &\leq \frac{1}{2}\left(\mathbf{H}\mathbf{z}_1^p\right)^T \mathbf{D}_1^p \left(\mathbf{H}\mathbf{z}_1^p\right) + c_1 \mathbf{e}_2^T \max\left(0,\ \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1^p\right) \end{aligned} \tag{44}$$

Substituting $\mathbf{u}$ and $\mathbf{u}^p$ in (40) by $\left\|\mathbf{H}\mathbf{z}_1^{(p+1)}\right\|_1^2$ and $\left\|\mathbf{H}\mathbf{z}_1^p\right\|_1^2$, respectively, leads to

$$\left\|\mathbf{H}\mathbf{z}_1^{(p+1)}\right\|_1 - \frac{\left\|\mathbf{H}\mathbf{z}_1^{(p+1)}\right\|_1^2}{2\left\|\mathbf{H}\mathbf{z}_1^p\right\|_1} \leq \left\|\mathbf{H}\mathbf{z}_1^p\right\|_1 - \frac{\left\|\mathbf{H}\mathbf{z}_1^p\right\|_1^2}{2\left\|\mathbf{H}\mathbf{z}_1^p\right\|_1} \tag{45}$$

Therefore, the following inequality holds:

$$\sum_{i=1}^{m_1} \left(\left|\mathbf{h}_i^T \mathbf{z}_1^{(p+1)}\right| - \frac{\left(\mathbf{h}_i^T \mathbf{z}_1^{(p+1)}\right)^2}{2\left|\mathbf{h}_i^T \mathbf{z}_1^p\right|} \right) \leq \sum_{i=1}^{m_1} \left(\left|\mathbf{h}_i^T \mathbf{z}_1^p\right| - \frac{\left(\mathbf{h}_i^T \mathbf{z}_1^p\right)^2}{2\left|\mathbf{h}_i^T \mathbf{z}_1^p\right|} \right) \tag{46}$$

(46) can be simplified to (47)

$$\begin{aligned} &\left\|\mathbf{H}\mathbf{z}_1^{(p+1)}\right\|_1 - \frac{1}{2}\left(\mathbf{H}\mathbf{z}_1^{(p+1)}\right)^T \mathbf{D}_1^P \left(\mathbf{H}\mathbf{z}_1^{(p+1)}\right) \\ &\leq \left\|\mathbf{H}\mathbf{z}_1^p\right\|_1 - \frac{1}{2}\left(\mathbf{H}\mathbf{z}_1^p\right)^T \mathbf{D}_1^P \left(\mathbf{H}\mathbf{z}_1^p\right) \end{aligned} \tag{47}$$

Combining inequalities (44) and (47), we obtain

$$\begin{aligned} &\left\|\mathbf{H}\mathbf{z}_1^{(p+1)}\right\|_1 + c_1 \mathbf{e}_2^T \max\left(0,\ \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1^{(p+1)}\right) \\ &\leq \left\|\mathbf{H}\mathbf{z}_1^p\right\|_1 + c_1 \mathbf{e}_2^T \max\left(0,\ \mathbf{e}_2 + \mathbf{G}\mathbf{z}_1^p\right) \end{aligned} \tag{48}$$

As the problem in (23) is bounded below 0, Algorithm 1 converges. The inequality in (48) holds when the algorithm converges. This indicates that the objective value of (23) decreases in each iteration till the algorithm converges. □

Theorem 2 *Algorithm* 1 *converges to a local minimal solution to problem* (23).

Proof The Lagrange function of problem (23) is as follows,

$$L_2(\mathbf{z}_1, \mathbf{q}_1) = \frac{1}{2}\|\mathbf{H}\mathbf{z}_1\|_1 + c_1\mathbf{e}_2^T\mathbf{q}_1 - \boldsymbol{\alpha}^T(-\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 - \mathbf{e}_2) - \boldsymbol{\beta}^T\mathbf{q}_1 \tag{49}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the vectors of Lagrange multipliers. Taking the derivative of $L_2(\mathbf{z}_1, \mathbf{q}_1)$ w.r.t. $\mathbf{z}_1$ and $\mathbf{q}_1$ respectively and setting them to zero, we obtain the KKT condition of problem (23) in the following,

$$\mathbf{H}^T\mathbf{D}_1\mathbf{H}\mathbf{z}_1 + \mathbf{G}\boldsymbol{\alpha}{=}0,\ c_1\mathbf{e}_2 - \boldsymbol{\alpha} - \boldsymbol{\beta}{=}0 \tag{50}$$

In each iteration of Algorithm 1, we find the optimal $\mathbf{z}_1^{(p+1)}$ to the problem (39). Hence, the converged solution of Algorithm 1 satisfies the KKT condition of the problem. Next, we define the Lagrange function of problem (39) of Algorithm 1, shown as follows,

$$L_3(\mathbf{z}_1, \mathbf{q}_1) = \frac{1}{2}\mathbf{z}_1^T\mathbf{H}^T\mathbf{D}_1\mathbf{H}\mathbf{z}_1 + c_1\mathbf{e}_2^T\mathbf{q}_1 - \boldsymbol{\alpha}^T(-\mathbf{G}\mathbf{z}_1 + \mathbf{q}_1 - \mathbf{e}_2) - \boldsymbol{\beta}^T\mathbf{q}_1 \tag{51}$$

Taking the derivative of $L_3(\mathbf{z}_1, \mathbf{q}_1)$ w.r.t. $\mathbf{z}_1$ and $\mathbf{q}_1$ respectively and setting them to zero.

$$\mathbf{H}^T\mathbf{D}_1\mathbf{H}\mathbf{z}_1 + \mathbf{G}\boldsymbol{\alpha}{=}0,\ c_1\mathbf{e}_2 - \boldsymbol{\alpha} - \boldsymbol{\beta}{=}0 \tag{52}$$

According to the definition of $\mathbf{D}_1$ in Algorithm 1, the equivalence between (50) and (52) holds when Algorithm 1 converges. This implies that the converged solution $\mathbf{z}_1^{(p+1)}$ of Algorithm 1 satisfies (50) (the KKT condition of the problem in (23)) and is a local minimum solution to problem (23). □

Next, we evaluate the validity and robustness of L1-TWSVM by experiments, and the classification performance is demonstrated by the experimental results on synthetic datasets and UCI datasets (Bache and Lichman 2013; Chen et al. 2011).

4 Experimental results

To evaluate the classification performance and robustness of L1-TWSVM, it is compared with five related algorithms [SVM (Vapnik 1995), GEPSVM (Mangasarian and Wild 2006), TWSVM (Jayadeva and Chandra 2007), LSTSVM (Kumar and Gopal 2009) and L1-NPSVM (Li et al. 2015a)], further, and demonstrate the L1-norm distance can alleviate the effect of outliers and noise in most cases, so fifteen commonly used datasets are selected from the UCI datasets. L1-TWSVM and L1-NPSVM are two iterative algorithms, which require initial solutions to be specified. Good initialization in L1-TWSVM is critical for success but is non-trivial. Considering these two algorithms are designed to correct the planes of GEPSVM and TWSVM that may be non-optimal due to the effect of outliers, we set their initial solutions as the solutions of GEPSVM and TWSVM, respectively. Moreover, for L1-TWSVM and L1-NPSVM, we terminate the iterative procedures when the difference in the objective values of two successive iterations is less than 0.001. The experimental environment consists of a Windows 10 operating system, an Intel(R) Core(TM) i5-5200u quad-core processor (2.2 GHz) and 4 GB of RAM. Six classification algorithms are implemented in MATLAB

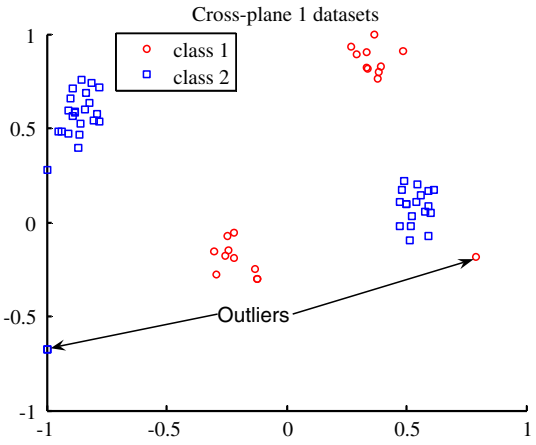

Fig. 1 XOR datasets with outliers

7.1. The experimental parameters are selected by 10-fold cross-validation (Ding et al. 2013; Ye et al. 2012). That is, each dataset is divided into ten subsets, one of which being testing data in turn, with the remaining nine subsets being training data. The testing accuracy is the average value of the results of N runs for each dataset (in this experiment, N = 10). In addition, the experimental datasets only contain two types of data (Class 1 & Class 2), and all sample data are normalized to the interval $(-1, 1)$ to reduce the differences between the characteristics of different samples. It is known that experimental parameters may influence the classification performance. Thus, to obtain the best generalization performance, all experimental parameters are selected by 10-fold cross-validation, which is described below. Parameters c_1 and c_2 are in the range of $\left\{2^i \,|i = -7, -6, -5, \ldots, 7\right\}$, while parameter ε is in the range of $\left\{10^i \,|i = -10, -9, -8, \ldots, 10\right\}$.

4.1 Experiments on synthetic datasets

To examine the performance of L1-TWSVM, we performed the same experiment on XOR datasets called Cross-plane (60×2), in which the number of positive samples is 20 and the number of negative samples is 40. This datasets is generated by perturbing points that originally lie on two intersecting planes (lines), where each plane corresponds to one class. The two-dimensional datasets contain two classes (positive class and negative class) with their covariance matrices are $(1, 0.9576; 0.9576, 1)$ and $(1, -0.9067; -0.9067, 1)$ respectively, while the mean vectors are $(4.39, 11.6062)$ and $(8.15, 11.4137)$ respectively. Outliers tend to have a certain influence on the classification performance; this influence is measured for evaluating the stability of the algorithms. Here, two extra outliers (data points that deviate significantly from the remainder data points) are added to the Cross-plane datasets (called Cross-plane 1 (62×2)) to assess the robustness of the six algorithms, among which one outlier with coordinate $(17, 5)^T$ is generated in the positive class, and another with coordinate $(-5, -1)^T$ is generated in the negative class, as shown in Fig. 1. The classification results of each classifier on the Cross-plane 1 datasets are given in Fig. 2a–f.

The traditional distance metric learning methods (such as SVM, GEPSVM, TWSVM and LSTSVM) often formulate the objectives using the squared L2-norm distance, but they could be highly influenced by outlying data points. We know that each point has the same contribution, especially the large distance point, if the squared L2-norm distance of them

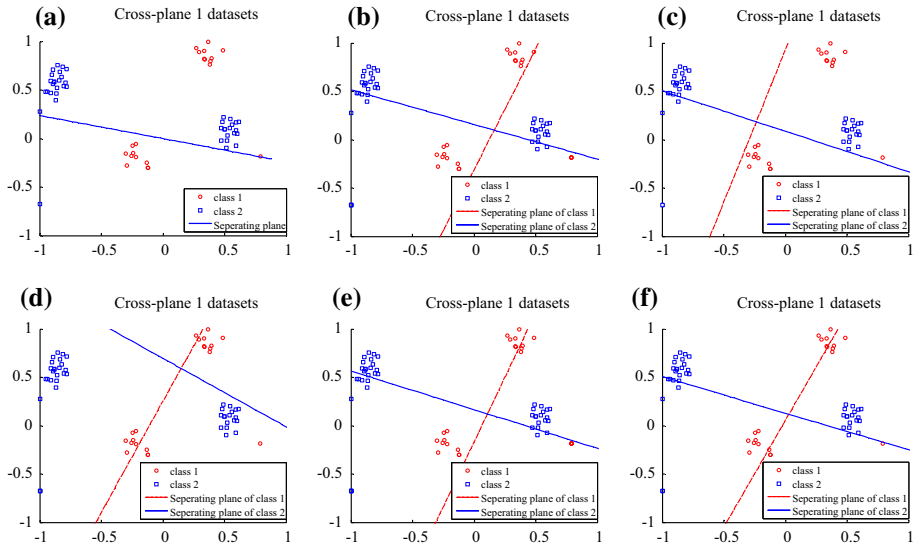

Fig. 2 The classification results on the Cross-plane 1 datasets. Annotation: The red line is the optimal plane of the "Circle" sample, while the blue line is the optimal plane of the "Square" sample. **a** By SVM, **b** By GEPSVM, **c** By TWSVM, **d** By LSTSVM, **e** By L1-NPSVM, **f** By L1-TWSVM (Color figure online)

dominates the sum (the squared L2-norm distance of remaining data points), it means that these measurements become inappropriate on the datasets. That is, these outlying data points are defined as outliers, which deviate significantly from the rest of the data points. According to Fig. 2, compared with L1-TWSVM, the other competing algorithms misclassify more points (Class 1 has more points closer to the blue separating plane of Class 2 or Class 2 has more points closer to the red separating plane of Class 1). The accuracies of the six algorithms (SVM, GEPSVM, TWSVM, LSTSVM, L1-NPSVM and L1-TWSVM) are 34.05, 74.34, 73.08, 67.86, 75.84 and 77.56%, respectively. According to the experimental results above, L1-TWSVM achieves the highest classification accuracy after the introduction of outliers. This may be attributed to the use of the robust L1-norm distance in TWSVM. The squared L2-norm distance may result in large distances dominating the sum in classifiers GEPSVM, TWSVM and LSTSVM when outliers appear in the datasets, which can easily lead to biased results; however, the L1-norm distance can greatly reduce the influence of outliers. The performance of SVM is the worst among six relative algorithms, which indicates SVM cannot deal with the XOR datasets effectively. These results validate the practicability and feasibility of L1-TWSVM.

4.2 Experiments on UCI datasets

To solve the L1-norm optimization problem, we developed an iterative method that is simple and convenient to implement. We also theoretically showed that the objective function value of L1-TWSVM is reduced in each step of the iteration. The objective function values of L1-TWSVM monotonically decrease as the iteration number increases until converging to fixed values (Fig. 3a–f); the algorithm can quickly converge within approximately six iterations. The horizontal axis represents the number of iterations, and the vertical axis represents the value of the objective function.

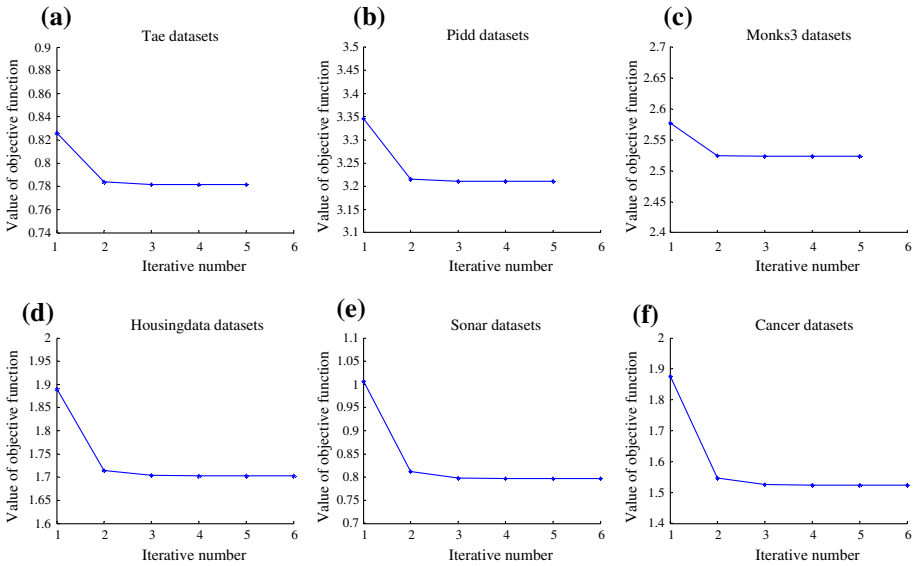

Fig. 3 The objective function values of L1-TWSVM monotonically decrease as the iteration number increases on six datasets. **a** On Tae datasets, **b** On Pidd datasets, **c** On Monks3 datasets, **d** On Housingdata datasets, **e** On Sonar datasets, **f** On Cancer datasets

To further evaluate the effectiveness and practicality of L1-TWSVM, it is compared with the relevant algorithms (SVM, GEPSVM, TWSVM, LSTSVM and L1-NPSVM) on fifteen commonly used datasets that are selected from the UCI datasets. Noise is one of the criteria used for evaluating the robustness of the algorithm. The accuracy changes smoothly with the increase of noise, which indicates that the algorithm has good robustness to noise.

To imitate the outlier data samples, we corrupt the training samples using a noise matrix $\mathbf{N}_0$ (the mean is 0 and the standard deviation is 1) whose element are *i.i.d.* (independent and identically distributed) standard Gaussian variables (Wang et al. 2015). Then we execute the training procedures on the corrupted training set $\mathbf{T} + \sigma\mathbf{N}_0$, where $\sigma = k\|\mathbf{T}\|_{\mathrm{F}} / \|\mathbf{N}_0\|_{\mathrm{F}}$ and k is a given noise factor. In this paper, we set $k = 0.1$. Table 1 lists the accuracies of the six algorithms on the original datasets, while Table 2 lists the accuracies on fifteen datasets where 10% Gaussian noise was introduced. Table 3 list the accuracies of the six algorithms on datasets where 20% Gaussian noise was introduced. To further test the convergence of L1-TWSVM, the average numbers of iterations used for training are listed in the three tables for each experiment. In addition, the P values are obtained from paired t tests comparing each algorithm to L1-TWSVM. An asterisk (*) indicates a significant difference from L1-TWSVM, which corresponds to a P value of less than 0.05. The highest accuracy is shown in bold. Standard deviation is a metric that is used to quantify the amount of variation or dispersion of a set of data values, called Std for short. Detailed results are given in the following tables:

We performed paired t tests comparing L1-TWSVM to the related algorithms. The P value for each test is the probability of the observed or a greater difference occurring between the correctness values of the two datasets, under the assumption of the null hypothesis that there is no difference between the correctness distributions of the datasets. Hence, the smaller the P-Value, the less likely it is that the observed difference resulted from datasets with the same correctness distribution. In this study, we set the threshold for the P value to 0.05.

Table 1 Experimental results of six algorithms on the original datasets

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Heart (270 × 13)	82.59 ± 5.25 0.2755 0.9994	72.59 ± 8.15* 0.0064 0.0108	**82.96 ± 4.74** 0.0209 0.5924	**82.96 ± 6.02** 0.0173 0.7271	68.15 ± 7.07* 0.0175 1.1796e−004	82.59 ± 5.98 0.1343 5
Monks1 (561 × 6)	56.65 ± 8.80* 0.0387 0.0018	78.79 ± 3.09 0.0061 0.5013	75.39 ± 6.46 0.0429 0.3679	74.14 ± 6.96 0.0196 0.6025	**79.33 ± 2.95** 0.0161 0.9199	74.86 ± 7.01 0.0822 5
Monks2 (601 × 6)	65.72 ± 5.68 0.0269 0.8813	**68.41 ± 7.73** 0.0065 0.1524	65.04 ± 6.12 0.1013 0.1936	65.54 ± 5.93 0.0202 0.4431	66.40 ± 7.17 0.0155 0.0841	66.05 ± 4.55 0.3178 7
Monks3 (554 × 6)	65.93 ± 12.82* 0.0405 9.0238e−004	78.51 ± 4.20* 0.0062 3.8023e−004	81.62 ± 6.93* 0.0675 0.0097	86.29 ± 3.11 0.0207 0.1699	84.28 ± 5.42 0.0145 0.1118	**87.91 ± 3.24** 0.3642 5
Wpbc (194 × 33)	73.03 ± 9.33* 0.4149 0.0479	76.29 ± 8.71* 0.0069 0.0105	**81.00 ± 6.34** 0.0262 0.2639	79.42 ± 8.18 0.0209 0.9993	75.74 ± 7.00* 0.0155 0.0014	79.42 ± 7.83 0.1188 5
Germ (1000 × 24)	71.50 ± 3.22* 1.3561 0.0094	70.70 ± 3.61* 0.0074 0.0056	76.90 ± 3.65 1.3657 0.0764	**77.60 ± 4.10** 0.0239 0.1934	70.50 ± 4.63* 0.0201 0.0207	74.40 ± 4.54 17.5300 5
Tae (151 × 5)	70.88 ± 11.18* 0.0494 0.0239	80.04 ± 11.21 0.0065 0.8883	82.75 ± 8.05 0.0171 0.0811	**83.42 ± 8.60** 0.0154 0.1039	79.38 ± 10.99 0.0147 0.7826	80.75 ± 8.20 0.0480 5
Cancer (683 × 9)	**97.36 ± 1.95** 0.0593 0.3024	95.76 ± 2.66 0.0059 0.4171	96.79 ± 1.92 0.0715 0.8528	96.93 ± 1.89 0.0190 0.5863	86.23 ± 10.20* 0.0153 0.0138	96.63 ± 1.86 0.4533 6

Table 1 continued

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Ticdata (958 × 9)	58.76 ± 5.14* 0.4193 1.5459e−004	65.87 ± 3.18* 0.0061 0.0106	65.97 ± 3.68* 0.5432 2.1681e−004	68.06 ± 4.10* 0.0155 0.0074	66.81 ± 2.81* 0.0177 0.0173	**71.30 ± 3.42** 10.8275 13
Pidd (768 × 8)	75.92 ± 6.51 0.5714 0.2664	74.74 ± 4.06* 0.0060 0.0311	76.30 ± 3.15* 0.0959 0.0491	77.08 ± 3.64* 0.0263 0.0230	73.70 ± 4.02* 0.0150 0.0381	**77.35 ± 2.01** 0.4255 5
Sonar (208 × 60)	73.57 ± 13.81 0.5526 0.7377	74.12 ± 8.13 0.0096 0.5002	74.02 ± 7.48 0.0217 0.5015	**76.43 ± 6.92** 0.0202 0.1227	74.14 ± 9.97 0.0217 0.4816	71.71 ± 8.44 0.1033 6
Pimadata (768 × 8)	75.92 ± 6.51* 0.6212 0.0254	74.74 ± 4.06* 0.0068 0.0311	76.30 ± 3.15* 0.0995 0.0491	77.08 ± 3.64* 0.0286 0.0230	73.70 ± 4.02* 0.0155 0.0381	**77.35 ± 2.00** 0.4256 5
Ionodata (351 × 34)	88.60 ± 7.01 0.3871 0.7523	78.07 ± 5.53* 0.0091 0.0123	**91.74 ± 4.32** 0.0315 0.0576	91.44 ± 4.44 0.0212 0.0724	80.06 ± 6.36* 0.0228 0.0365	87.47 ± 6.14 0.2796 5
Clevedata (297 × 13)	84.83 ± 5.53 0.2685 0.8561	85.89 ± 4.11 0.0065 0.5289	85.28 ± 7.29 0.0223 0.1909	**86.94 ± 7.03*** 0.0200 0.0362	84.53 ± 5.17 0.0183 0.9262	84.26 ± 7.54 0.1510 5
Housingdata (506 × 13)	85.58 ± 3.72 0.1323 0.1808	73.14 ± 5.42* 0.0060 9.8126e−004	**86.37 ± 6.03** 0.1011 0.0558	84.78 ± 5.13 0.0156 0.1233	72.32 ± 5.28* 0.0145 0.0054	83.02 ± 5.42 0.4578 6

Table 2 Experimental results of six algorithms on datasets into which 10% Gaussian noise has been introduced

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Heart (270 × 13)	78.52 ± 6.15 0.1157 0.6656	70.74 ± 8.02* 0.0064 0.0405	79.63 ± 6.47 0.0840 0.8717	77.41 ± 6.07* 0.0157 0.0443	70.74 ± 6.51* 0.0397 0.0199	**80.00 ± 7.07** 0.4399 5
Monks1 (561 × 6)	54.71 ± 10.12* 0.0347 5.7622e−004	79.15 ± 3.15 0.0064 0.1100	72.89 ± 6.26 0.1229 0.0664	75.04 ± 3.87 0.0160 0.9531	**80.04 ± 3.49*** 0.0518 0.0401	74.86 ± 7.01 0.1240 9
Monks2 (601 × 6)	65.73 ± 4.37 0.0331 0.8439	**69.41 ± 8.16** 0.0061 0.4317	65.54 ± 6.21 0.1286 0.5562	62.39 ± 5.13* 0.0158 0.0268	67.41 ± 7.57 0.0408 0.7601	66.21 ± 5.18 0.2298 7
Monks3 (554 × 6)	64.26 ± 15.04* 0.0488 0.0014	77.97 ± 4.76* 0.0062 0.0031	82.70 ± 6.81* 0.0761 0.0246	85.94 ± 5.96 0.0152 0.3923	83.74 ± 5.60 0.0409 0.0749	**87.02 ± 4.22** 0.3400 5
Wpbc (194 × 33)	70.53 ± 9.51* 0.0690 0.0380	75.21 ± 6.94 0.0070 0.7707	**78.92 ± 6.81** 0.0452 0.1959	76.29 ± 8.32 0.0172 0.9962	75.74 ± 7.00 0.0396 0.8860	76.26 ± 9.13 0.0656 5
Germ (1000 × 24)	70.40 ± 5.46* 1.2418 0.0356	70.30 ± 3.82* 0.0070 0.0101	73.20 ± 3.60 0.9523 0.8534	**74.30 ± 3.32** 0.0204 0.1809	70.60 ± 5.12* 0.0500 0.0262	73.10 ± 3.33 21.3388 5
Tae (151 × 5)	76.21 ± 13.27 0.0311 0.3810	81.38 ± 10.29 0.0059 0.8953	**83.42 ± 6.20** 0.0186 0.1039	82.75 ± 8.05 0.0147 0.1938	80.71 ± 10.98 0.0147 0.9931	80.75 ± 8.20 0.0735 5
Cancer (683 × 9)	97.07 ± 1.03 0.0761 0.2869	95.46 ± 2.96 0.0063 0.4653	95.91 ± 2.32 0.0901 0.5915	**97.08 ± 1.84** 0.0184 0.2443	83.89 ± 6.98* 0.0423 9.9683e−005	96.05 ± 2.16 0.4597 6

Table 2 continued

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Ticdata (958 × 9)	59.49 ± 4.85* 0.5711 1.9028e−004	65.14 ± 2.97* 0.0062 0.0027	67.02 ± 3.67* 0.1787 4.3492e−004	67.85 ± 3.97* 0.0183 0.0036	65.76 ± 2.64* 0.0399 0.0037	**71.09 ± 3.16** 11.8102 13
Pidd (768 × 8)	73.83 ± 4.82* 0.6475 0.0458	75.13 ± 4.42 0.0060 0.1596	77.08 ± 4.52 0.3794 0.6139	74.86 ± 4.54* 0.0178 0.0087	74.48 ± 3.68* 0.0442 0.0362	**77.28 ± 8.25** 1.3796 5
Sonar (208 × 60)	72.12 ± 14.34 0.4123 0.5900	75.57 ± 9.75 0.0097 0.3240	**76.40 ± 8.21** 0.0246 0.0520	74.48 ± 8.69 0.0213 0.1591	74.64 ± 10.56 0.0578 0.3742	69.17 ± 11.42 0.1014 6
Pimadata (768 × 8)	73.83 ± 4.82* 0.7190 0.0458	75.13 ± 4.42 0.0060 0.1596	77.08 ± 4.52 0.4043 0.6139	74.86 ± 4.54* 0.0153 0.0087	74.48 ± 3.68* 0.0410 0.0362	**77.31 ± 8.04** 1.1946 5
Ionodata (351 × 34)	88.03 ± 6.37 0.3845 0.6104	77.79 ± 5.93* 0.0090 0.0048	90.60 ± 4.06 0.0215 0.1648	**90.88 ± 5.08** 0.0237 0.1420	81.17 ± 8.23* 0.0563 0.0399	86.03 ± 8.99 0.1119 5
Clevedata (297 × 13)	84.49 ± 6.08 0.3131 0.7797	85.54 ± 4.21 0.0060 0.5220	84.60 ± 6.95 0.0342 0.3479	**85.60 ± 6.80** 0.0153 0.0528	84.53 ± 5.17 0.0480 0.7587	83.60 ± 7.91 0.1752 6
Housingdata (506 × 13)	81.82 ± 3.89 0.2815 0.7530	71.76 ± 5.15* 0.0062 0.0066	**84.02 ± 5.97** 0.1874 0.1489	81.83 ± 5.32 0.0162 0.4174	72.32 ± 5.28* 0.0435 0.0189	80.85 ± 7.02 0.5979 6

Table 3 Experimental results of six algorithms on datasets into which 20% Gaussian noise has been introduced

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Heart (270 × 13)	71.48 ± 7.23* 0.2728 0.0136	66.30 ± 7.85* 0.0066 0.0073	75.93 ± 6.25 0.0373 0.6274	72.22 ± 6.68* 0.0153 0.0456	69.26 ± 9.52* 0.0378 0.0406	**79.55 ± 6.63** 0.1053 7
Monks1 (561 × 6)	56.10 ± 12.92* 0.0513 0.0048	78.43 ± 3.95 0.0064 0.1969	71.65 ± 7.15* 0.1729 0.0039	74.32 ± 6.86 0.0160 0.4959	**79.69 ± 3.05** 0.0174 0.0663	74.86 ± 7.01 0.1202 9
Monks2 (601 × 6)	65.05 ± 6.79 0.0434 0.7916	**68.07 ± 6.81** 0.0061 0.5386	63.88 ± 7.87 0.1567 0.2173	65.22 ± 5.54 0.0193 0.2719	67.74 ± 6.35 0.0398 0.5914	65.88 ± 5.80 0.2435 7
Monks3 (554 × 6)	70.18 ± 12.54* 0.0615 0.0040	78.52 ± 3.94* 0.0061 0.0015	81.08 ± 7.19* 0.0961 0.0214	86.30 ± 6.14 0.0155 0.7767	84.28 ± 5.03* 0.0460 0.0455	**87.02 ± 4.22** 0.4715 5
Wpbc (194 × 33)	71.53 ± 10.83* 0.0790 0.0447	76.26 ± 8.46 0.0069 0.9031	77.34 ± 7.67 0.0489 0.7970	**78.37 ± 8.98** 0.0169 0.3376	75.74 ± 7.00 0.0394 0.7904	76.79 ± 9.70 0.0765 5
Germ (1000 × 24)	70.67 ± 2.86 1.2705 0.6893	70.60 ± 4.74 0.0080 0.7301	71.40 ± 2.65 1.2207 0.8723	**73.10 ± 4.76** 0.0284 0.2768	70.70 ± 4.52 0.0488 0.7723	71.30 ± 2.57 20.8417 5
Tae (151 × 5)	75.58 ± 13.74 0.0337 0.5147	79.38 ± 10.15 0.0061 0.9928	**81.42 ± 7.84** 0.0280 0.3437	**81.42 ± 9.85** 0.0152 0.1934	80.71 ± 10.98 0.0143 0.8143	79.42 ± 10.58 0.1061 5
Cancer (683 × 9)	96.93 ± 1.43 0.0777 0.9962	95.61 ± 2.78 0.0061 0.2054	**97.37 ± 1.42** 0.0881 0.3419	96.79 ± 2.22 0.0169 0.7315	83.18 ± 7.53* 0.0456 1.2907e−004	96.93 ± 1.90 0.5112 6

Table 3 continued

Datasets (N × n)	SVM	GEPSVM	TWSVM	LSTSVM	L1-NPSVM	L1-TWSVM
	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) *P* value	Test ± Std (%) Training time (s) Iteration number
Ticdata (958 × 9)	59.28 ± 5.40* 0.6013 4.4125e−004	65.66 ± 2.85* 0.0065 0.0075	66.29 ± 4.13* 0.5911 0.0031	68.37 ± 3.59* 0.0158 0.0066	65.76 ± 2.84* 0.0452 0.0105	**71.41 ± 3.63** 10.0191 13
Pidd (768 × 8)	75.40 ± 4.06 0.5456 0.3092	74.74 ± 3.81* 0.0063 0.0483	73.95 ± 3.64* 0.0296 0.0417	75.64 ± 4.15 0.0152 0.1003	74.48 ± 3.76* 0.0445 0.0498	**77.08 ± 3.66** 0.1222 5
Sonar (208 × 60)	76.50 ± 9.76 0.5054 0.1813	**76.02 ± 7.89** 0.0096 0.2259	71.57 ± 10.37 0.0325 0.2228	73.07 ± 8.58 0.0206 0.1967	71.69 ± 8.54 0.0572 0.5676	68.17 ± 13.50 0.1483 6
Pimadata (768 × 8)	75.40 ± 4.06 0.5488 0.0899	74.74 ± 3.81* 0.0065 0.0483	74.85 ± 4.82* 0.0317 0.0306	75.64 ± 4.15 0.0160 0.1003	74.48 ± 3.76* 0.0401 0.0498	**77.08 ± 3.66** 0.1237 4
Ionodata (351 × 34)	87.46 ± 6.55 0.3386 0.5585	77.50 ± 6.15* 0.0091 5.7687e−004	89.46 ± 6.64 0.0212 0.8910	**92.02 ± 4.57** 0.0210 0.1760	81.75 ± 6.94* 0.0539 0.0064	89.16 ± 5.56 0.1321 5
Clevedata (297 × 13)	**85.86 ± 4.90** 0.3387 0.2603	85.55 ± 4.43 0.0063 0.2577	82.29 ± 12.07 0.0417 0.9158	83.94 ± 8.22 0.0160 0.1032	85.53 ± 4.72 0.0403 0.3609	82.60 ± 8.21 0.1458 6
Housingdata (506 × 13)	79.05 ± 3.87 0.4195 0.6866	72.55 ± 4.84* 0.0079 0.0095	**83.81 ± 5.47*** 0.0758 0.0293	80.05 ± 6.12 0.0165 0.8951	72.32 ± 5.28* 0.0428 0.0260	80.24 ± 6.45 0.2795 6

For instance, the P value of the test comparing L1-TWSVM and TWSVM on the Ticdata datasets is 0.00021681, and that on the Monks3 datasets is 0.0097, both of which are less than 0.05; therefore, we can conclude that L1-TWSVM and TWSVM have different accuracies on these datasets and L1-TWSVM significantly outperforms TWSVM. In addition, on the Monks1, Monks2, Tae and Sonar datasets, we find that the performance differences between L1-TWSVM and the other algorithms (except for SVM) are statistically insignificant. Finally, by more carefully examining the experimental results, we also notice that although TWSVM and LSTSVM outperform L1-TWSVM in terms of accuracy on some datasets (such as Heart, Ionodata, and Housingdata), the P values are higher than 0.05; that is to say, the accuracies of TWSVM and LSTSVM are not significantly different from that of L1-TWSVM. This important observation clearly indicates that the classification performance of our method is superior to those of all other competing methods.

Based on the data in Table 1, we find the following interesting patterns. First, the accuracy of L1-TWSVM is comparable to those of other competing algorithms in most cases and is higher than those of others in some scenarios. This indicates that the classification performance of L1-TWSVM is better. Second, according to the columns corresponding to L1-TWSVM in Table 1, L1-TWSVM can rapidly converge within approximately seven iterations, except on the Ticdata datasets (the iteration number is 13); as guaranteed by Theorem 1, L1-TWSVM gradually converges to a local optimal solution.

According to the experimental results in the three tables, regardless of whether Gaussian noise is introduced or not, the accuracy of our method is comparable to or better than the other methods. The performance degradation of our method is very small when 10% and 20% Gaussian noise is introduced; even the accuracy of the proposed algorithm is superior to TWSVM. In addition, the accuracies of L1-NPSVM and L1-TWSVM show little change compared to other competing methods, especially when Gaussian noise is introduced. This may be attributed to the embedding of the L1-norm distance, which makes them more robust to outliers than the other methods. This further demonstrates that the L1-norm distance is useful for data classification, especially for samples with outliers.

According to the three tables, the iteration numbers of L1-TWSVM increase slightly after Gaussian noise is introduced; however, L1-TWSVM can converge within a limited number of iterations. Furthermore, the experimental results expose high computational cost. The training time of L1-TWSVM is the longest. The reasons for this are as follows: (1) L1-TWSVM, similar to TWSVM, requires the calculation of two QPPs; the time complexity of this calculation is no more than $m^3/4$ when $\mathbf{A}$ is equivalent to $\mathbf{B}$ in terms of the number of samples. (2) The time complexity of computing the two inverse matrices $\left(\mathbf{H}^T\mathbf{D}_1\mathbf{H}+\varepsilon\mathbf{I}\right)^{-1}$ and $\left(\mathbf{G}^T\mathbf{D}_2\mathbf{G}+\varepsilon\mathbf{I}\right)^{-1}$ is approximately $2n^2$. (3) As L1-TWSVM is an iterative algorithm that need to iteratively compute the solutions, in each iteration, it needs to calculate two QPPs, two inverse matrices and two diagonal matrices $\mathbf{D}_1$, $\mathbf{D}_2$, where the time complexities of calculating $\mathbf{D}_1$ and $\mathbf{D}_2$ during the learning process are $m_1\times(d+1)$ and $m_2\times(d+1)$, respectively. Therefore, the total time complexity of solving problem (14) is $l\left(m^3/4+2n^2+m(d+1)\right)$, where l is the iteration number. The iteration procedure results in L1-TWSVM with higher computational cost than the other five methods in most cases; fortunately, it surpasses them in accuracy and has good robustness.

Although the accuracy improvements of our method over the other compared methods on the original datasets are mediocre, the performance degradation of the proposed method is very small (less than 3.2%) when Gaussian noise is introduced. According to the experimental results in the three tables, the performances of all the methods are degraded due to the introduction of Gaussian noise; however, the degradation of our new method is much less

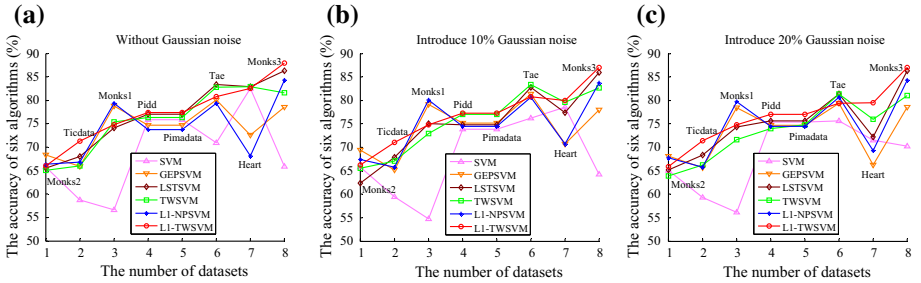

Fig. 4 Comparison of six algorithms on eight datasets with respect to classification accuracy. **a** Without Gaussian noise, **b** Introduce 10% Gaussian noise, **c** Introduce 20% Gaussian noise

than those of the other competing methods. This clearly indicates the robustness of our improved method to outliers and empirically confirms the validity of our strategy of using the robust L1-norm distance to improve the distance metric learning.

From Table 2, although the accuracy of L1-TWSVM is higher than that of TWSVM on the Heart datasets, the P value of the test comparing L1-TWSVM and TWSVM is 0.8717, which is higher than 0.05. This means that the accuracy of TWSVM is not significantly different from that of L1-TWSVM. The same scenario can be seen on the Monks1, Monks2, Pidd and other datasets. However, the P value of L1-TWSVM and TWSVM is 0.0246 on Monks 3 datasets, which is less than 0.05, we get the different accuracies of L1-TWSVM and TWSVM, the same is true on Ticdata datasets. Besides, the performance degradation of TWSVM is larger than that of L1-TWSVM after the Gaussian noise is introduced. This indicates that the robustness of L1-TWSVM is superior to TWSVM. In an extremely similar way, we can get the results of comparing the other algorithms (SVM, GEPSVM and LSTSVM) with L1-TWSVM respectively. Note that, the accuracy of L1-NPSVM has a little change after the Gaussian noise is introduced, but L1-TWSVM outperforms L1-NPSVM in accuracy.

To more clearly test the classification performance of the improved method (L1-TWSVM), and show the validity of L1-norm distance is useful for data classification, especially for datasets with outliers. Eight original datasets (Monks2, Ticdata, Monks1, Pidd, Pimadata, Tae, Heart and Monks3) are chosen from fifteen commonly used datasets. This is the main motivation for us to do it. Figure 4a shows the classification accuracies of the six algorithms on eight original datasets. Figure 4b, c shows the accuracies of the six algorithms when 10 and 20% Gaussian noise, respectively, is introduced. According to Fig. 4, the accuracy of our improved method is comparable to those of the other compared methods on the original datasets. However, the improvements achieved by our method on the contaminated datasets (with 10 and 20% Gaussian noise) are large in most cases. This indicates that the classification performance of L1-TWSVM is superior, further validates the practicability of utilizing the L1-norm distance in TWSVM, and shows that the proposed methods are more effective and robust against outlier samples than traditional squared L2-norm distance metric learning approaches.

To evaluate the robustness of L1-TWSVM, we designed three test schemes (called Test 1, Test 2, and Test 3) on the Heart and Pidd datasets separately. We introduce 0, 10 and 20% Gaussian noise into the three tests, respectively. As illustrated in Fig. 5, L1-TWSVM is superior to the other competing algorithms in most cases; in particular, on Test 3 (Fig. 5a), the accuracy of L1-TWSVM is the highest. In addition, on Test 1 and Test 2, the accuracies of L1-TWSVM are comparable to those of TWSVM. In brief, our proposed method consistently outperforms the other compared methods on the three tests, which demonstrates that L1-TWSVM can improve the classification performance and is useful for data classification.

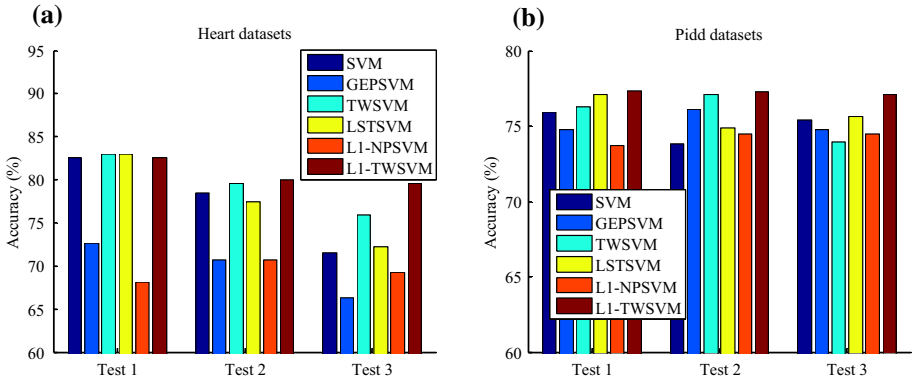

Fig. 5 The classification performances of six algorithms. **a** On Heart datasets, **b** On Pidd datasets

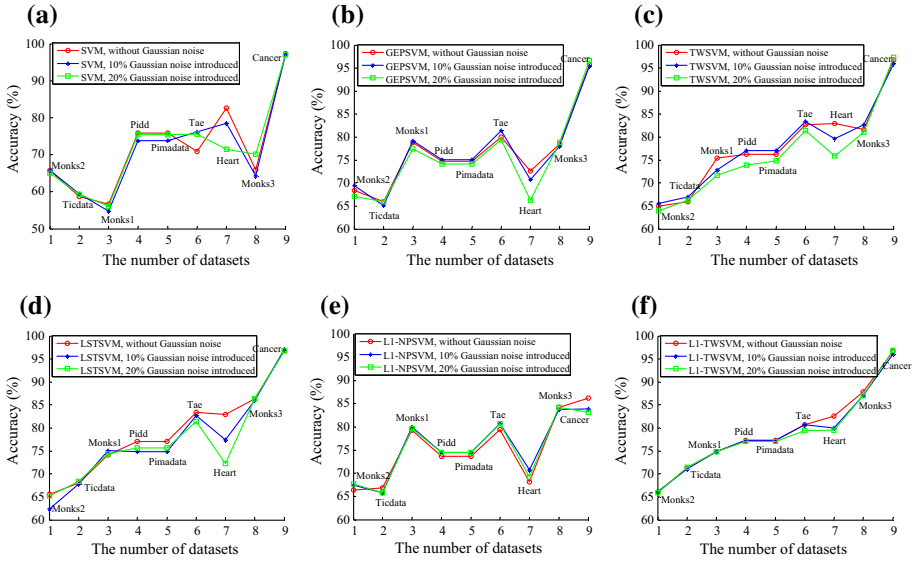

Fig. 6 The performance comparison of six algorithms on nine datasets. **a** By SVM, **b** By GEPSVM, **c** By TWSVM, **d** By LSTSVM, **e** By L1-NPSVM, **f** By L1-TWSVM

Combining Fig. 5a, b, we find they show similar experimental results. Therefore, Fig. 5b further verifies the practicability of L1-TWSVM in alleviating the effect of noise. These observations are consistent with those made in the previous experiments.

To further confirm the robustness to noise of L1-TWSVM, Fig. 6 vividly illustrates the performance comparison of the six algorithms on nine datasets, where 0, 10 and 20% Gaussian noise has been introduced. In terms of robustness, L1-TWSVM and L1-NPSVM obtain the best results among all competing methods, which indicates that the utilization of the L1-norm distance can alleviate the negative influence of outliers and make the model stronger. However, the accuracy of L1-TWSVM is higher than that of L1-NPSVM, which firmly demonstrates that our method is more effective and robust against outlier samples. Further-

more, this provides more evidence of the effectiveness of the robust L1-norm distance in metric learning and verifies the correctness of our improved method.

5 Conclusions and future work

We propose an efficient and robust TWSVM classifier based on the L1-norm distance metric for binary classification, which is denoted as L1-TWSVM. It makes full use of the robustness of the L1-norm distance to noise and outliers. As the new objective function contains the non-smooth L1-norm term, it is challenging to directly solve the optimization problem. In view of this, we develop a simple and valid iterative algorithm for solving L1-norm optimization problems that is easy to implement and prove that the objective function of the proposed method can obtain the local optimal solutions in theory. Moreover, extensive experimental results indicate that L1-TWSVM can effectively suppress the negative effects of outliers to some extent and improves the generalization ability and flexibility of the model. Nevertheless, L1-TWSVM still needs to iteratively compute two QPPs to obtain the solutions. According to the experimental results above, the computational cost of L1-TWSVM is the highest compared with other related algorithms under the same scenario. This makes it difficult to effectively address large data samples. In summary, L1-TWSVM has better classification performance and robustness than the other algorithms, especially when Gaussian noise is introduced.

There are three future directions for this research. First, we would like to find a better way to decrease the computational cost of L1-TWSVM so that it will be able to handle larger samples. Second, we would like to extend L1-TWSVM to a kernel version to deal with nonlinear tasks. Third, L1-TWSVM is only effective for binary classification problems at present; it is promising to extend L1-TWSVM to multi-category classification and study the application of multi-class L1-TWSVM to real-world problems.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. 61772273, 61373062 and 61603190), and the Fundamental Research Funds for the Central Universities (No. 30918011104). Postgraduate Research & Practice Innovation Program of Jiangsu Province 2018 (No. KYCX18_0424).

References

Bache, K., & Lichman, M. (2013). UCI machine learning repository. Available at http://archive.ics.uci.edu/ml/datasets.html.

Bradley, P. S., & Mangasarian, O. L. (2000). Massive data discrimination via linear support vector machines. *Optimization Methods & Software, 13,* 1–10.

Cayton, L., & Dasgupta, S. (2006). Robust euclidean embedding. In *International conference* (pp. 169–176).

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems & Technology, 2,* 389–396.

Chen, X., Yang, J., Ye, Q., & Liang, J. (2011). Recursive projection twin support vector machine via within-class variance minimization. *Pattern Recognition, 44,* 2643–2655.

Cortes, C., & Vapnik, V. (1995). Support vector network. *Machine Learning, 20,* 273–297.

Deng, N., Tian, Y., & Zhang, C. (2012). *Support vector machines. Optimization based theory, algorithms, and extensions*. Boca Raton: CRC Press.

Ding, S., Hua, X., & Yu, J. (2013). An overview on nonparallel hyperplane support vector machine algorithms. *Neural Computing and Applications, 25,* 975–982.

Ding, C., Zhou, D., He, X., & Zha, H. (2006). R1-PCA:rotational invariant L1-norm principal component analysis for robust subspace factorization. In *International conference on machine learning* (pp. 281–288).

Fung, G., & Mangasarian, O.L. (2001). Proximal support vector machine classifiers. In *ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 77–86).

Gao, J. (2008). Robust L1 principal component analysis and its bayesian variational inference. *Neural Computation, 20,* 555.

Gao, S., Ye, Q., & Ye, N. (2011). 1-norm least squares twin support vector machines. *Neurocomputing, 74,* 3590–3597.

Guarracino, M. R., Cifarelli, C., Seref, O., & Pardalos, P. M. (2007). A classification method based on generalized eigenvalue problems. *Optimization Methods & Software, 22,* 73–81.

Jayadeva, K. R., & Chandra, S. (2007). Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 29,* 905–910.

Ke, Q., & Kanade, T. (2005). Robust L1-norm factorization in the presence of outliers and missing data by alternative convex programming. In *IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005* (Vol. 731, pp. 739–746).

Kumar, M. A., & Gopal, M. (2008). Application of smoothing technique on twin support vector machines. *Pattern Recognition Letters, 29,* 1842–1848.

Kumar, M. A., & Gopal, M. (2009). Least squares twin support vector machines for pattern classification. *Expert Systems with Applications, 36,* 7535–7543.

Kwak, N. (2008). Principal component analysis based on L1-norm maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30,* 1672–1680.

Kwak, N. (2014). Principal component analysis by Lp-norm maximization. *IEEE Transactions on Cybernetics, 44,* 594–609.

Li, C. N., Shao, Y. H., & Deng, N. Y. (2015a). Robust L1-norm non-parallel proximal support vector machine. *Optimization, 65,* 1–15.

Li, C. N., Shao, Y. H., & Deng, N. Y. (2015b). Robust L1-norm two-dimensional linear discriminant analysis. *Neural Networks the Official Journal of the International Neural Network Society, 65C,* 92–104.

Lin, G., Tang, N., & Wang, H. (2015). Locally principal component analysis based on L1-norm maximisation. *IET Image Processing, 9,* 91–96.

Liu, X. J., Chen, S. C., & Peng, H. J. (2002). Computer keystroker verification based on support vector machines. *Journal of Computer Research & Development, 39,* 1082–1086.

Mangasarian, O. L., & Wild, E. W. (2006). Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28,* 69–74.

Muralidharan, V., Sugumaran, V., & Indira, V. (2014). Fault diagnosis of monoblock centrifugal pump using SVM. *Engineering Science & Technology An International Journal, 17,* 152–157.

Nie, F., Huang, H., Ding, C., Luo, D., & Wang, H. (2015). Robust principal component analysis with non-greedy L1-norm maximization. In *International joint conference on artificial intelligence* (pp. 1433–1438).

Pang, Y., Li, X., & Yuan, Y. (2010). Robust tensor analysis with L1-norm. *IEEE Transactions on Circuits and Systems for Video Technology, 20,* 172–178.

Parlett, B. N. (1998). *The symmetric eigenvalue problem*. Philadelphia: SIAM.

Qi, Z., Tian, Y., & Shi, Y. (2013a). Robust twin support vector machine for pattern classification. *Pattern Recognition, 46,* 305–316.

Qi, Z., Tian, Y., & Shi, Y. (2013b). Structural twin support vector machine for classification. *Knowledge-Based Systems, 43,* 74–81.

Shao, Y. H., Chen, W. J., & Deng, N. Y. (2014). Nonparallel hyperplane support vector machine for binary classification problems. *Information Sciences, 263,* 22–35.

Shao, Y. H., Deng, N. Y., Chen, W. J., & Wang, Z. (2013). Improved generalized eigenvalue proximal support vector machine. *IEEE Signal Processing Letters, 20,* 213–216.

Shao, Y. H., Zhang, C. H., Wang, X. B., & Deng, N. Y. (2011). Improvements on twin support vector machines. *IEEE Transactions on Neural Networks, 22,* 962–968.

Song, Q., Hu, W., & Xie, W. (2002). Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems Man & Cybernetics Part C, 32,* 440–448.

Subasi, A. (2013). Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Computers in Biology and Medicine, 43,* 576–586.

Tian, S. F., & Huang, H. K. (2000). Database learning algorithms based on support vector machine. *Journal of Computer Research & Development, 37,* 17–22.

Vapnik, V. (1995). *The nature of statistical learning theory* (pp. 988–999). Berlin: Springer.

Wang, H., Lu, X., Hu, Z., & Zheng, W. (2014a). Fisher discriminant analysis with L1-norm. *IEEE Transactions on Cybernetics, 44,* 828–842.

Wang, H., Nie, F.P., & Huang, H. (2014b). Robust distance metric learning via simultaneous L1-norm minimization and maximization. In *International conference on machine learning* (pp. 1836–1844).

Wang, H., Nie, F., & Huang, H. (2015). Learning robust locality preserving projection via p-order minimization. In *Proceedings of the 29 AAAI conference on artificial intelligence* (pp. 3059–3065).

Wang, H., Tang, Q., & Zheng, W. (2012). L1-norm-based common spatial patterns. *IEEE Transactions on Bio-Medical Engineering, 59,* 653–662.

Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems* (pp. 2080–2088).

Ye, Q., Yang, X., Gao, S., & Ye, N. (2017). L1-norm distance minimization based fast robust twin support vector k-plane clustering. *IEEE Transactions on Neural Networks and Learning Systems, 29,* 1–10.

Ye, Q., Yang, J., Liu, F., Zhao, C., Ye, N., & Yin, T. (2016). L1-norm distance linear discriminant analysis based on an effective iterative algorithm. *IEEE Transactions on Circuits and Systems for Video Technology, 99,* 1–14.

Ye, Q., & Ye, N. (2009). Improved proximal support vector machine via generalized eigenvalues. In *International joint conference on computational sciences and optimization, Cso 2009*, Sanya, Hainan, China, 24–26 April (pp. 705–709).

Ye, Q., Zhao, C., Gao, S., & Zheng, H. (2012). Weighted twin support vector machines with local information and its application. *Neural Networks the Official Journal of the International Neural Network Society, 35,* 31–39.

Ye, Q., Zhao, C., & Xiaobo, C. (2011a). A feature selection method for TWSVM via a regularization technique. *Journal of Computer Research & Development, 48,* 1029–1037.

Ye, Q., Zhao, C., Ye, N., & Chen, X. (2011b). Localized twin SVM via convex minimization. *Neurocomputing, 74,* 580–587.

Yin, H., Jiao, X., Chai, Y., & Fang, B. (2015). Scene classification based on single-layer SAE and SVM. *Expert Systems with Applications, 42,* 3368–3380.

Zhong, F., & Zhang, J. (2013). Linear discriminant analysis based on L1-norm maximization. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 22,* 3018–3027.