Neural Process Lett
DOI 10.1007/s11063-017-9714-3

L1-Norm GEPSVM Classifier Based on an Effective Iterative Algorithm for Classification

He Yan[1] · Qiaolin Ye[1] · Tianan Zhang[1,2] · Dong-Jun Yu[3] · Yiqing Xu[1]

Abstract The proximal support vector machine via generalized eigenvalues (GEPSVM) is an excellent classifier for binary classification problem. However, in conventional GEPSVM the distance is measured by L2-norm, which makes it prone to being affected by the presence of outliers by the square operation. To alleviate this, we propose a robust and effective GEPSVM classification algorithm based on L1-norm distance metric, termed as L1-GEPSVM. The optimization goal is to minimize the intra-class distance dispersion, and maximize the inter-class distance dispersion simultaneously. It is known that the application of L1-norm distance is often used as a simple and powerful way to reduce the impact of outliers, which improves the generalization ability and flexibility of the model. In addition, we develop an effective iterative algorithm to solve the L1-norm optimal problems, which is easy to implement and its convergence to a local optimum is theoretically ensured. Thus, the classification performance of L1-GEPSVM is more robust than GEPSVM. Finally, the feasibility and effectiveness of L1-GEPSVM are further verified by extensive experimental results on artificial datasets, UCI datasets and NDC datasets.

Keywords GEPSVM · L1-GEPSVM · L1-norm · L2-norm · Outliers

✉ Qiaolin Ye
yqlcom@njfu.edu.cn

He Yan
yanhecom@163.com

[1] College of Information Science and Technology, Nanjing Forestry University,
No. 159 Longpan Road, Nanjing 210037, China

[2] Collaborative Innovation Center of Sustainable Forestry in Southern China of Jiangsu Province,
Nanjing Forestry University, Nanjing 210037, China

[3] School of Computer Science and Engineering, Nanjing University of Science and Technology,
Xiaolingwei 200, Nanjing 210094, China

Published online: 27 September 2017

1 Introduction

Support Vector Machine [1–3] (SVM) plays a very important role in data classification and regression problems. Its main idea of SVM is to seek an optimal plane by maximizing the margin between two parallel support planes [4]. Thus, as an effective classification tool, SVM has been widely applied to many practical problems, such as image classification [5], scene classification [6], fault diagnosis [7], bioinformatics [8] and so on.

However, there are two main troubles in the original SVM: XOR problem and the complex Quadratic Programming Problems (QPP). To solve the two problems above, Mangasarian and Wild proposed a fast classifier for binary classification problem, termed as proximal SVM via generalized eigenvalues (GEPSVM) [9], which is an extension of proximal SVM (PSVM) [10]. GEPSVM relaxes the requirement of PSVM that the planes should be parallel, and tries to find two nonparallel planes by solving two generalized eigenvalue problems, thus it deals with the XOR problem smoothly and has better generalization ability than that of SVM. The geometric interpretation of GEPSVM 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 [11]. At present, the research about GEPSVM is still in the ascendant, and many improved methods of GEPSVM have been developed based on the idea of GEPSVM. The advantages of GEPSVM bring great impact to its multifarious improvement [11–17]. Jayadeva et al. put forward a twin support vector machine (TWSVM) [11] based on the idea of GEPSVM, TWSVM solves two QPPs to replace generalized eigenvalue problems. So the computational time of TWSVM is only 1/4 of the standard SVM. But it takes much time to compute two QPPs when the sample is large enough. Ye [12] reformulated the optimization problems of GEPSVM by solving a simple eigenvalue problem rather than generalized eigenvalues, and proposed a new algorithm via singular value decomposition (IGEPSVM) to deal with the singular problem. Besides, Ye [12] also proposed an algorithm of IDGEPSVM to overcome the bad performance when influenced by noise data and longer training time problem in GEPSVM. Later, Ye [18] put forward a feature selection method for TWSVM via a regularization technique (RTWSVM), which makes use of the regularization technique to overcome the possible singular problem and improve the generalization ability. Guarracino [13] reformulated the optimization problems of GEPSVM by using the regularization technique to solve the generalized eigenvalue problem, and proposed a regularized general eigenvalue classifier (ReGEC). Shao [14] raised an improved version of GEPSVM (IGEPSVM) by using the standard eigenvalue decomposition instead of the generalized eigenvalue decomposition, which can solve the singular problem. Also, an extra meaningful parameter is introduced, which improves the classification generalization ability. Marghny [16] proposed an improved version of GEPSVM by using Differential Search Algorithm to find near optimal values of the GEPSVM parameters and its kernel parameters, termed as DSA-GEPSVM, which overcomes the influence of error or noise in real world. Zhang [19] reformulated the optimization problems of twin Mahalanobis distance-based support vector machine (TMSVM) by using the least squares sense, and constructed a fast least squares version of TMSVM which solves two modified primal problems instead of two dual problems. By combining least squares TMSVM and directed acyclic graph (DAG), Zhang also proposed a new multiclass classification algorithm, named DAGLSTMSVM for multi-class classification. In order to facilitate the robustness and generalization of nonparallel proximal support vector machine (L1-NPSVM), Li [15] reformulated the optimization problems of PSVM by using L1-norm distance to replace L2-norm distance, and put forward a gradient ascending (GA) iterative algorithm to solve the objective function, which is simple to carry out but may not ensure 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 [20].

But it should be noted that GEPSVM is sensitive to outliers, because in GEPSVM the distance is measured by L2-norm, which may exaggerate the effect of outliers by the square operation [21]. To improve model robustness in the presence of the outliers, L1-norm distance is applied in the algorithms [15,22–25], and the utilization of L1-norm distance is often considered as a simple and powerful way to reduce the impact of noises. In this study, we propose a robust GEPSVM based on L1-norm distance for binary classification problem, termed as L1-GEPSVM. It is to seek two nonparallel optimal planes by solving a pair of QPPs instead of generalized eigenvalue problems, whose goal is to make each plane closest to the samples of its own class and at the same time furthest from the samples of other classes. In summary, our L1-GEPSVM owns the following several compelling properties: (1) L1-GEPSVM converts the generalized eigenvalue problem into a strong convex programming problem, besides, we implement a simple iterative algorithm to solve the L1-norm optimal problems, and its convergence to a local optimum is theoretically ensured; (2) in L1-GEPSVM the distance is measured by L1-norm, which is more robust to outliers, and it can efficiently decrease the impact of the outliers even if the ratio of outliers is large; (3) extensive experimental results confirm that, compared with GEPSVM [9], IGEPSVM [14], TWSVM [11] and L1-NPSVM [15], L1-GEPSVM and L1-NPSVM can effectively alleviate the effect of the outliers, which improves the generalization ability and flexibility of model; (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 GEPSVM. We are planning to study these in the future.

This paper is organized as follows. Section 2 briefly introduces the GEPSVM. Section 3 proposes L1-GEPSVM with its feasibility and theoretical analysis. All the experimental results are shown in Sect. 4 and conclusions are given in Sect. 5.

2 Related Works

In this paper, 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, denote $\mathbf{I}$ as an identity matrix of appropriate dimension. We consider a binary classification problem in the n dimensional real space R^n, the set of training sample is indicated by $T = \left\{\left(\mathbf{x}_j^{(i)}, y_i\right) | i = 1, 2,\ \ j = 1, 2, \ldots, m_i\right\}$, where $\mathbf{x}_j^{(i)} \in R^n$ and $y_i \in \{-1,\ \ 1\}$, $\mathbf{x}_j^{(i)}$ denotes the i-th class and j-th sample. We suppose that matrix $\mathbf{A} = \left[\mathbf{A}_1{}^{(1)}, \mathbf{A}_2{}^{(1)}, \ldots, \mathbf{A}_{m_1}{}^{(1)}\right]^T$ with size of $m_1 \times n$ represents the data points of class 1 , while matrix $\mathbf{B} = \left[\mathbf{B}_1{}^{(2)}, \mathbf{B}_2{}^{(2)}, \ldots, \mathbf{B}_{m_2}{}^{(2)}\right]^T$ with size of $m_2 \times n$ represents the data points of class -1. Matrices $\mathbf{A}$ and $\mathbf{B}$ represent all the training data points, where $m_1 + m_2 = m$. And 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 famous nonparallel proximal classifiers: GEPSVM [10] and TWSVM [11].

2.1 GEPSVM

The GEPSVM [10] classifier aims to seek two nonparallel proximal optimal planes:

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

where $\mathbf{w}_1, \mathbf{w}_2 \in R^n$, $b_1, b_2 \in R$. Its aim is to minimize the Euclidean distance of the planes from the data points of Class 1 and Class -1 respectively. This produces the following two objective problems of GEPSVM:

$$\min_{(\mathbf{w}_1,b_1)\neq 0} \frac{||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 / ||(\mathbf{w}_1 \; b_1)^T||_2^2}{||\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1||_2^2 / ||(\mathbf{w}_1 \; b_1)^T||_2^2} \tag{2}$$

$$\min_{(\mathbf{w}_2,b_2)\neq 0} \frac{||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2 / ||(\mathbf{w}_2 \; b_2)^T||_2^2}{||\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2||_2^2 / ||(\mathbf{w}_2 \; b_2)^T||_2^2} \tag{3}$$

where $||\cdot||_2$ denotes the L2-norm. This is implicitly assumed that $\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1 \neq 0$ and $\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2 \neq 0$. The original problems (2) and (3) can be optimized in the following form:

$$\min_{\mathbf{w}_1,b_1} \frac{||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2}{||\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1||_2^2} \tag{4}$$

$$\min_{\mathbf{w}_2,b_2} \frac{||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2}{||\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2||_2^2} \tag{5}$$

The positive semi-definite matrix may be involved in the computation when solving the generalized eigenvalue equations, which may cause singularity problem. Therefore, formulas (4) and (5) can be regularized by introducing Tikhonov regularization terms, shown as followed:

$$\min_{\mathbf{w}_1,b_1} \frac{||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 + \delta||(\mathbf{w}_1 \; b_1)^T||_2^2}{||\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1||_2^2} \tag{6}$$

$$\min_{\mathbf{w}_2,b_2} \frac{||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2 + \delta||(\mathbf{w}_2 \; b_2)^T||_2^2}{||\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2||_2^2} \tag{7}$$

where $\delta||(\mathbf{w}_1 \;\; b_1)^T||_2^2$ and $\delta||(\mathbf{w}_2 \;\; b_2)^T||_2^2$ are regularization terms, δ is a regularization factor, and the regularization term can improve the stability and classification accuracy of GEPSVM. The optimization goal is to make each plane as close as possible to one of the two classes and as far as possible from the other class. Formulas (6) and (7) are equivalent to:

$$\min \frac{\mathbf{z}_1^T \mathbf{E} \mathbf{z}_1}{\mathbf{z}_1^T \mathbf{F} \mathbf{z}_1} \tag{8}$$

$$\min \frac{\mathbf{z}_2^T \mathbf{L} \mathbf{z}_2}{\mathbf{z}_2^T \mathbf{M} \mathbf{z}_2} \tag{9}$$

where $\mathbf{H} = [\mathbf{A} \quad \mathbf{e}_1]$, $\mathbf{G} = [\mathbf{B} \;\; \mathbf{e}_2]$, $\mathbf{E} = \mathbf{H}^T\mathbf{H} + \delta\mathbf{I}$, $\mathbf{F} = \mathbf{G}^T\mathbf{G}$, $\mathbf{L} = \mathbf{G}^T\mathbf{G} + \delta\mathbf{I}$, $\mathbf{M} = \mathbf{H}^T\mathbf{H}$, $\mathbf{z}_1 = (\mathbf{w}_1 \;\; b_1)^T$, $\mathbf{z}_2 = (\mathbf{w}_2 \;\; b_2)^T$.

Both $\mathbf{H}$ and $\mathbf{G}$ are symmetric, while formulas (8) and (9) are Rayleigh quotient problems. It is easy to obtain the solutions of (8) and (9) by solving the generalized eigenvalue problem.

$$\mathbf{G}\mathbf{z}_1 = \lambda_1 \mathbf{H}\mathbf{z}_1, \quad \mathbf{z}_1 \neq 0 \tag{10}$$

$$\mathbf{H}\mathbf{z}_2 = \lambda_2 \mathbf{G}\mathbf{z}_2, \quad \mathbf{z}_2 \neq 0 \tag{11}$$

The minimum of (8) is achieved at an eigenvector corresponding to the smallest eigenvalue λ_1 of (10). Thus, if $\mathbf{z}_1$ denotes the eigenvector corresponding to λ_1, then $\mathbf{z}_1 = (\mathbf{w}_1 \;\; b_1)^T$, the first d components make up the weight vector $\mathbf{w}_1$ of the first planes, and the last component is the deviation b_1. $\mathbf{z}_1$ determines the plane $\mathbf{x}^T\mathbf{w}_1 + b_1 = 0$, which is close to data points of

Class 1. And $\mathbf{z}_2 = (\mathbf{w}_2 \ b_2)^T$ determines the plane $\mathbf{x}^T\mathbf{w}_2 + b_2 = 0$, which is close to data points of Class -1.

GEPSVM is designed for binary classification problems, when the regularization term is introduced, the objective problems need to optimize the regularization factor, and optimization strategy refers to [10]. After **G** and **H** are regularized, they may turn into positive definite matrices, which guaranteed that Eqs. (10) and (11) have no singularity problem. However, after bringing in the regularization term, it no longer has the original geometric meaning.

2.2 TWSVM

TWSVM is an outstanding classifier for binary classification problem, which solves two QPPs (the scale is relatively small compared to that of SVM) to obtain two nonparallel planes. The primary problems of TWSVM are shown as following:

$$\begin{aligned}&\min_{\mathbf{w}_1,b_1,\mathbf{q}_2} \frac{1}{2}||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 + c_1\mathbf{e}_2^T\mathbf{q}_2\\ &s.t. \quad -(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 \geq \mathbf{e}_2, \quad \mathbf{q}_2 \geq 0 \end{aligned} \tag{12}$$

$$\begin{aligned}&\min_{\mathbf{w}_2,b_2,\mathbf{q}_1} \frac{1}{2}||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_2^2 + c_2\mathbf{e}_1^T\mathbf{q}_1\\ &s.t. \quad (\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2) + \mathbf{q}_1 \geq \mathbf{e}_1, \quad \mathbf{q}_1 \geq 0 \end{aligned} \tag{13}$$

where $||\cdot||_2$ denotes the L2-norm, $\mathbf{w}_1, \mathbf{w}_2 \in R^n$,$b_1, b_2 \in R$, $\mathbf{q}_1$ 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. The optimization goal of TWSVM is that each plane is closer to one of the two classes and away from the other class as far as possible [11].

The Lagrange corresponding to the formula (12) is given by:

$$L(\mathbf{w}_1, b_1, \mathbf{q}_2, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_2^2 + c_1\mathbf{e}_2^T\mathbf{q}_2 - \boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 - \mathbf{e}_2) - \boldsymbol{\beta}^T\mathbf{q}_2 \tag{14}$$

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 Karush–Kuhn–Tucker (KKT) necessary optimality conditions for formula (12) are shown as following:

$$\frac{\partial L}{\partial \mathbf{w}_1} = \mathbf{A}^T(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{B}^T\boldsymbol{\alpha} = 0, \tag{15}$$

$$\frac{\partial L}{\partial b_1} = \mathbf{e}_1^T(\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1) + \mathbf{e}_2^T\boldsymbol{\alpha} = 0, \tag{16}$$

$$\frac{\partial L}{\partial \mathbf{q}_2} = c_1\mathbf{e}_2 - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0, \tag{17}$$

$$-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 \geq \mathbf{e}_2, \quad \mathbf{q}_2 \geq 0, \tag{18}$$

$$\boldsymbol{\alpha}^T(-(\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1) + \mathbf{q}_2 - \mathbf{e}_2) = 0, \quad \boldsymbol{\beta}^T\mathbf{q}_2 = 0, \tag{19}$$

$$\boldsymbol{\alpha} \geq 0, \quad \boldsymbol{\beta} \geq 0 \tag{20}$$

Since $\boldsymbol{\beta} \geq 0$, from Eq. (17) we get

$$0 \leq \boldsymbol{\alpha} \leq c_1 \tag{21}$$

Next, Eqs. (15) and (16) are combined to be:

$$\left(\left[\mathbf{A}^T \ \ \mathbf{e}_1^T\right][\mathbf{A} \ \ \mathbf{e}_1] + c_3\mathbf{I}\right)[\mathbf{w}_1 \ \ b_1]^T + \left[\mathbf{B}^T \ \ \mathbf{e}_2^T\right]\boldsymbol{\alpha} = 0 \tag{22}$$

We define $\mathbf{H} = [\mathbf{A} \ \ \mathbf{e}_1]$, $\mathbf{G} = [\mathbf{B} \ \ \mathbf{e}_2]$ and $\mathbf{z}_1 = (\mathbf{w}_1 \ \ b_1)^T$, $\mathbf{z}_2 = (\mathbf{w}_2 \ \ \ b_2)^T$, with these notations, Eq. (22) can be rewritten as

$$\mathbf{H}^T\mathbf{H}\mathbf{z}_1 + \mathbf{G}^T\boldsymbol{\alpha} = 0 \tag{23}$$

Equation (23) is equivalent to

$$\mathbf{z}_1 = -\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \tag{24}$$

Note that inverse matrix $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}$ is easy to encounter singular problem. Therefore, regularization term $\varepsilon\mathbf{I}$, $\varepsilon > 0$ is introduced to solve this. $\left(\mathbf{H}^T\mathbf{H} + \varepsilon\mathbf{I}\right)^{-1}$ satisfies positive definiteness, which does not suffer from the singular problem. We rewrite Eq. (24) as the following form:

$$\mathbf{z}_1 = -\left(\mathbf{H}^T\mathbf{H} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \tag{25}$$

Using formula (14) and the K.K.T conditions above, we obtain the Wolfe dual problem of formula (12) as below:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}_2^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ s.t. \quad & 0 \le \boldsymbol{\alpha} \le c_1\mathbf{e}_2 \end{aligned} \tag{26}$$

Similarly, we consider formula (13) and obtain its dual problem as

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \mathbf{e}_1^T\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \\ s.t. \quad & 0 \le \boldsymbol{\beta} \le c_2\mathbf{e}_1 \end{aligned} \tag{27}$$

Two nonparallel proximal planes can be obtained by using $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to solve Eq. (28).

$$\begin{aligned} \mathbf{z}_1 = [\mathbf{w}_1 \ \ b_1]^T &= -\left(\mathbf{H}^T\mathbf{H} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{G}^T\boldsymbol{\alpha}, \\ \mathbf{z}_2 = [\mathbf{w}_2 \ \ b_2]^T &= \left(\mathbf{G}^T\mathbf{G} + \varepsilon\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{\beta} \end{aligned} \tag{28}$$

Further, weight vectors $\mathbf{w}_1, \mathbf{w}_2$ and b_1, b_2 deviations can be obtained. Thus we can get two nonparallel proximal planes.

$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0, \ \ \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{29}$$

A new data point $\mathbf{x}$ is assigned to Class 1 or Class -1 depending on its proximity to the two nonparallel planes.

3 GEPSVM Based on L1-Norm Distance

The advantages of GEPSVM are remarkable, but we cannot ignore the deficiency. It can be seen that in GEPSVM the distance is measured by L2-norm. In order to obtain the minimum value of the objective function, GEPSVM emphasizes the role of outliers remote from the sample by the square operation, which is easy to exaggerate their impact and reduce the classification accuracy. To alleviate this, we propose a GEPSVM classification algorithm based on L1-norm distance metric, termed as L1-GEPSVM. The use of L1-norm makes the model more robust. Further, L1-GEPSVM attempts to seek two nonparallel planes that can be produced by solving two strong convex programming problems instead of generalized eigenvalue problems. In addition, L1-GEPSVM inherits the advantages of GEPSVM of solving the XOR problem. Then the two objective problems of L1-GEPSVM are shown as followed:

$$\min_{\mathbf{w}_1,b_1} \frac{||\mathbf{A}\mathbf{w}_1 + \mathbf{e}_1 b_1||_1 + \delta||(\mathbf{w}_1\ b_1)^T||_2^2}{||\mathbf{B}\mathbf{w}_1 + \mathbf{e}_2 b_1||_1} \tag{30}$$

$$\min_{\mathbf{w}_2,b_2} \frac{||\mathbf{B}\mathbf{w}_2 + \mathbf{e}_2 b_2||_1 + \delta||(\mathbf{w}_2\ b_2)^T||_2^2}{||\mathbf{A}\mathbf{w}_2 + \mathbf{e}_1 b_2||_1} \tag{31}$$

where $||\cdot||_1$ denotes the L1-norm, $\delta \parallel (\mathbf{w}_1\ b_1)^T \parallel_2^2$ and $\delta \parallel (\mathbf{w}_2\ b_2)^T \parallel_2^2$ are regularization terms, and δ is a regularization factor. The aim of L1-GEPSVM is to make points of the same class as compact as possible while as far as possible from the other class [11], which guarantees the objective function to be minimized.

The original problems can be optimized in the following form:

$$\min_{\mathbf{z}_1} \frac{\|\mathbf{H}\mathbf{z}_1\|_1 + \delta\mathbf{z}_1^T\mathbf{z}_1}{\|\mathbf{G}\mathbf{z}_1\|_1} \tag{32}$$

$$\min_{\mathbf{z}_2} \frac{\|\mathbf{G}\mathbf{z}_2\|_1 + \delta\mathbf{z}_2^T\mathbf{z}_2}{\|\mathbf{H}\mathbf{z}_2\|_1} \tag{33}$$

where $\mathbf{H} = [\mathbf{A}\ \ \mathbf{e}_1]$, $\mathbf{G} = [\mathbf{B}\ \ \mathbf{e}_2]$, and $\mathbf{z}_1 = (\mathbf{w}_1\ \ \mathrm{b}_1)^T$, $\mathbf{z}_2 = (\mathbf{w}_2\ \ \mathrm{b}_2)^T$. We can obtain two nonparallel optimal planes by solving formulas (32) and (33):

$$\mathbf{x}^T\mathbf{w}_1 + b_1 = 0, \quad \mathbf{x}^T\mathbf{w}_2 + b_2 = 0 \tag{34}$$

Next, we solve formula (32), whose objective function is invariant to the order of magnitude of $\mathbf{w}_1$. Then, we can scale $\mathbf{z}_1$ so that the denominator of formula (32) is equal to 1,that is $||\mathbf{G}\mathbf{z}_1||_1 = 1$. So (32) can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{z}_1} ||\mathbf{H}\mathbf{z}_1||_1 + \delta\mathbf{z}_1^T\mathbf{z}_1 \\ &\text{s.t.} \ \ \|\mathbf{G}\mathbf{z}_1\|_1 = 1 \end{aligned} \tag{35}$$

where $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{m_1})^T \in \mathbf{R}^{m_1\times(n+1)}$ and $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{m_2})^T \in \mathbf{R}^{m_2\times(n+1)}$, $\mathbf{h}_i, \mathbf{g}_i \in \mathbf{R}^n (i = 1, 2, \ldots, n)$ denote the i-th column of matrix $\mathbf{H}$ and matrix $\mathbf{G}$ separately. Thus, formula (35) is equivalent to:

$$\begin{aligned} &\min_{\mathbf{z}_1} \sum_{i=1}^{n} \left|\mathbf{h}_i^T\mathbf{z}_1\right| + \delta\mathbf{z}_1^T\mathbf{z}_1 \\ &\text{s.t.} \ \ \sum_{i=1}^{n} \left|\mathbf{g}_i^T\mathbf{z}_1\right| = 1 \end{aligned} \tag{36}$$

where $|.|$ is the absolute value operation. According to the relevant knowledge of mathematics, we can know that,

$$\sum_{i=1}^{n}\left|\mathbf{h}_i^T\mathbf{z}_1\right| = \sum_{i=1}^{n}\left|\mathbf{z}_1^T\mathbf{h}_i\right| = \mathbf{z}_1^T\left(\sum_{i=1}^{n}\frac{\mathbf{h}_i\mathbf{h}_i^T}{\left|\mathbf{z}_1^T\mathbf{h}_i\right|}\right)\mathbf{z}_1 \tag{37}$$

$$\sum_{i=1}^{n}\left|\mathbf{g}_i^T\mathbf{z}_1\right| = \sum_{i=1}^{n}\left|\mathbf{z}_1^T\mathbf{g}_i\right| = \sum_{i=1}^{n} sign(\mathbf{z}_1^T\mathbf{g}_i)(\mathbf{z}_1^T\mathbf{g}_i) \tag{38}$$

where $sign(\cdot)$ is a symbolic function: when the value of the bracket is greater than 0, its function value is 1, otherwise -1. In this way, formula (35) can be described by the following equivalence model:

$$\begin{aligned}\min_{\mathbf{z}_1}\ & \mathbf{z}_1^T\left(\sum_{i=1}^{n}\frac{\mathbf{h}_i\mathbf{h}_i^T}{\left|\mathbf{z}_1^T\mathbf{h}_i\right|}\right)\mathbf{z}_1 + \delta\mathbf{z}_1^T\mathbf{z}_1 \\ \text{s.t.}\ & \sum_{i=1}^{n} sign(\mathbf{z}_1^T\mathbf{g}_i)(\mathbf{z}_1^T\mathbf{g}_i) = 1\end{aligned} \tag{39}$$

Obviously, it is hard to solve formula (39), which contains absolute value operation. To solve this, we propose an iterative convex optimization strategy. The basic idea is iteratively update the objective function value until it converges to a fixed value, so the vector $\mathbf{z}_1^{(p)}$ is a local optimal solution that we seek. Assuming that $\mathbf{z}_1^{(p)}$ is the optimal 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:

$$\begin{aligned}\min_{\mathbf{z}_1}\ & \mathbf{z}_1^T\left(\sum_{i=1}^{n}\frac{\mathbf{h}_i\mathbf{h}_i^T}{\left|\mathbf{z}_1^T\mathbf{h}_i\right|}\right)\mathbf{z}_1 + \delta\mathbf{z}_1^T\mathbf{z}_1 \\ \text{s.t.}\ & \sum_{i=1}^{n} sign(\mathbf{z}_1^T\mathbf{g}_i)(\mathbf{z}_1^T\mathbf{g}_i) = 1\end{aligned} \tag{40}$$

where $\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_i\right|$ should satisfy $\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_i\right| \neq 0$, and it is easy to prove that $sign(\mathbf{z}_1^{(p)^T}\mathbf{g}_i)(\mathbf{z}_1^T\mathbf{g}_i)$ is a first order Taylor expansion of $\left|\mathbf{g}_i^T\mathbf{z}_1\right|$ at the point $\mathbf{z}_1^{(p)}$. (40) are rewritten as:

$$\begin{aligned}\min_{\mathbf{z}_1}\ & \mathbf{z}_1^T\left(\Gamma^{(p)} + \delta\mathbf{I}\right)\mathbf{z}_1 \\ \text{s.t.}\ & \mathbf{s}^{(p)}\mathbf{G}\mathbf{z}_1 = 1\end{aligned} \tag{41}$$

where $\Gamma^{(p)} = \mathbf{H}^T diag\left(1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_1\right|\right), 1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_2\right|\right), \ldots, 1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_{m_1}\right|\right)\right)\mathbf{H}$, $\mathbf{s}^{(p)} = sign(\mathbf{z}_1^{(p)^T}\mathbf{G}^T)$ and $\mathbf{I} \in \mathbf{R}^{n\times n}$.

The algorithm we design is described in Algorithm 1. In each iteration, $\mathbf{z}_1$ is calculated with the current Γ, and then Γ is updated based on the current result $\mathbf{z}_1$. The iteration procedure is repeated until the algorithm converges.

```
Input: Matrices $\mathbf{H} \in R^{m_1 \times (n+1)}$ and $\mathbf{G} \in R^{m_2 \times (n+1)}$.
Result: $\mathbf{z}_1^{(p+1)} \in R^{(n+1)\times 1}$.
Set $p = 0$, Initialize $\Gamma$ and $\mathbf{z}_1$. $\Gamma = \mathbf{H}^T diag\left(1/|\mathbf{z}_1\mathbf{H}|\right)\mathbf{H}$, where $\mathbf{z}_1$ is a standard solution of GEPSVM.
Repeat
    1. Compute
       $\mathbf{z}_1^{(p+1)} = \frac{\left(\Gamma^{(p)} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}}{\mathbf{s}^{(p)}\mathbf{G}\left(\Gamma^{(p)} + \delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}}.$    (42)
    2. Update matrix $\Gamma^{(p)} = \mathbf{H}^T diag \left(1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_1\right|\right), 1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_2\right|\right), \ldots, 1/\left(\left|\mathbf{z}_1^{(p)^T}\mathbf{h}_{m_1}\right|\right)\right)\mathbf{H}$, the i-th element of $\Gamma^{(p)}$ is $\mathbf{h}_i{}^T diag\left(1/|\mathbf{z}_1\mathbf{h}_i|\right)\mathbf{h}_i$
    3. $p = p + 1$
Until Convergence.
Output: The local optimal solution of $\mathbf{z}_1$.
```

Algorithm 1: An efficient iterative algorithm to solve the optimization problem of formula (41). The proof procedure is shown as below:

Definition Define any two vectors **a** and **p**, assume the following inequality is established:

$$
\begin{aligned}
Q(\mathbf{a}) &= \frac{\mathbf{a}^T\mathbf{S}_3\mathbf{a}}{\mathbf{a}^T\left(\mathbf{S}_1 + \beta\mathbf{I}\right)\mathbf{a} + \lambda\|\mathbf{a}\|_1} \\
&\geq \frac{\mathbf{p}^T\mathbf{S}_3\mathbf{p}}{\mathbf{p}^T\left(\mathbf{S}_1 + \beta\mathbf{I}\right)\mathbf{p} + \lambda\|\mathbf{p}\|_1} = Q(\mathbf{p})
\end{aligned}
\tag{43}
$$

Then, we assume that **a** is a better solution than **p**, so we can make the conclusion that $\mathbf{z}_1^{(p+1)}$ is a better solution than $\mathbf{z}_1^{(p)}$ in the iterative algorithm. This conclusion is given by Theorem 1. To prove it, we first introduce Lemma 1.

Lemma 1 *For any vector* $\mathbf{c} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]^T \in {}^n$, *the following equality is established* [26]*:*

$$
\|\mathbf{c}\|_1 = \min_{v \in {}^n_+} \frac{1}{2}\sum_{i=1}^{n}\frac{\mathbf{c}_i^2}{\mathbf{v}_i} + \frac{1}{2}\|\mathbf{v}\|_1 \tag{44}
$$

The minimum is uniquely arrived at $v_i = |c_i|$ *for* $i = 1, 2, \ldots, n$, *where* $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]^T$.

Theorem 1 *Assume that* $\mathbf{z}_1^{(p)}$ *is a vector, which makes the equation* $sign\left(\mathbf{z}_1{}^{(p)^T}\mathbf{G}^T\right)\mathbf{G}\mathbf{z}_1 = 1$. *By formula* (41) *we can obtain the solution* $\mathbf{z}_1^{(p+1)}$, *which is better than* $\mathbf{z}_1^{(p)}$. *Here, we solve formula* (41) *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 From the definition of $\mathbf{z}_1^{(p+1)}$ in formula (41), we have that

$$
\sum_{j=1}^{m_2}\mathbf{s}_j{}^{(p)}\mathbf{g}_j\mathbf{z}_1^{(p+1)} = 1 \tag{45}
$$

Let

$$J\left(\mathbf{z}_1\right) = \frac{1}{2}\mathbf{z}_1^T\left(\Gamma^{(p)} + \delta\mathbf{I}\right)\mathbf{z}_1 + \frac{1}{2}\left\|{\mathbf{z}_1}^{(p)^T}\mathbf{H}\right\|_1 \tag{46}$$

Then, from the physical meaning of $\mathbf{z}_1^{(p+1)}$, we have that

$$\begin{aligned} J\left({\mathbf{z}_1}^{(p+1)}\right) &= \tfrac{1}{2}\sum_{i=1}^{m_1}\frac{\left[\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right]^2}{\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right|} + \tfrac{1}{2}\left\|{\mathbf{z}_1}^{(p)^T}\mathbf{H}\right\|_1 \\ \leq J\left({\mathbf{z}_1}^{(p)}\right) &= \sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right| \end{aligned} \tag{47}$$

In addition, from Lemma 1, we have that

$$\begin{aligned} J\left({\mathbf{z}_1}^{(p+1)}\right) &= \frac{1}{2}\sum_{i=1}^{m_1}\frac{\left[\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right]^2}{\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right|} + \frac{1}{2}\left\|{\mathbf{z}_1}^{(p)^T}\mathbf{H}\right\|_1 \\ &\geq \frac{1}{2}\sum_{i=1}^{m_1}\frac{\left[\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right]^2}{\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right|} + \frac{1}{2}\left\|{\mathbf{z}_1}^{(p+1)^T}\mathrm{H}\right\|_1 \\ &= \sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right| \end{aligned} \tag{48}$$

Combining (47) and (48), we can obtain

$$\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right| \geq \sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right| \tag{49}$$

Combining (45) and (49), we have that

$$\begin{aligned} Q\left({\mathbf{z}_1}^{(p+1)}\right) &= \frac{\sum_{j=1}^{m_2}\left|\mathbf{g}_j{\mathbf{z}_1}^{(p+1)}\right|}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right|} \geq \frac{\sum_{j=1}^{m_2}{\mathbf{s}_j}^{(p)}\mathbf{g}_j\mathbf{z}_1^{(p+1)}}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right|} \\ &= \frac{1}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p+1)^T}\right|} \geq \frac{1}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right|} \end{aligned} \tag{50}$$

From the equality $\sum_{j=1}^{m_2}\left|\mathbf{g}_j\mathbf{z}_1^{(p)}\right| = 1$, we can obtain

$$Q\left({\mathbf{z}_1}^{(p)}\right) = \frac{\sum_{j=1}^{m_2}\left|\mathbf{g}_j{\mathbf{z}_1}^{(p)}\right|}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right|} = \frac{1}{\sum_{i=1}^{m_1}\left|\mathbf{h}_i{\mathbf{z}_1}^{(p)^T}\right|} \tag{51}$$

Combining (50) and (51), we have that

$$Q\left({\mathbf{z}_1}^{(p+1)}\right) \geq Q\left({\mathbf{z}_1}^{(p)}\right) \tag{52}$$

So, $\mathbf{z}_1^{(p+1)}$ is better than $\mathbf{z}_1^{(p)}$. □

Formula (41) is a convex optimization problem with equality constraints, and it has a close-form solution. Now, we can set up the Lagrange function of formula (41) to solve this objective problem, as shown in the following:

$$L(\mathbf{z}_1,\boldsymbol{\kappa})=\ \mathbf{z}_1^T\left(\Gamma^{(p)}+\delta\mathbf{I}\right)\mathbf{z}_1-\ \boldsymbol{\kappa}\left(\ \mathbf{s}^{(p)}\mathbf{G}\mathbf{z}_1-1\right) \tag{53}$$

where $\boldsymbol{\kappa}$ is a Lagrange multiplier. Taking the derivative of $L\left(\mathbf{z}_1,\boldsymbol{\kappa}\right)$ w.r.t $\mathbf{z}_1$, and setting the derivative to be zero, we can easily get the following equation:

$$L(\mathbf{z}_1,\boldsymbol{\kappa})=\ \left(\Gamma^{(p)}+\delta\mathbf{I}\right)\mathbf{z}_1-\ \boldsymbol{\kappa}\mathbf{G}^T\mathbf{s}^{(p)^T}\ =0 \tag{54}$$

The solution of $\mathbf{z}_1^{(p+1)}$ can be obtained by Eq. (54).

$$\mathbf{z}_1^{(p+1)}=\boldsymbol{\kappa}\left(\Gamma^{(p)}+\delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T} \tag{55}$$

Bring Eq. (55) into $\mathbf{s}^{(p)}\mathbf{G}\mathbf{z}_1=1$, we can get an expression about $\boldsymbol{\kappa}$, shown as following:

$$\begin{aligned}&\boldsymbol{\kappa}\mathbf{s}^{(p)}\mathbf{G}\left(\Gamma^{(p)}+\delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}=1\\&\Rightarrow\boldsymbol{\kappa}\ \mathbf{=}1\Big/\left(\mathbf{s}^{(p)}\mathbf{G}\left(\Gamma^{(p)}+\delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}\right)\end{aligned} \tag{56}$$

Combining Eqs. (55) and (56), we can obtain

$$\mathbf{z}_1^{(p+1)}=\frac{\left(\Gamma^{(p)}+\delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}}{\mathbf{s}^{(p)}\mathbf{G}\left(\Gamma^{(p)}+\delta\mathbf{I}\right)^{-1}\mathbf{G}^T\mathbf{s}^{(p)^T}} \tag{57}$$

Increase p until the objective function value converges to a fixed value. Since the problem in formula (41) is a convex problem, then $\mathbf{z}_1^{(p+1)}$ is a local optimal solution that we seek. Further, weight vector $\mathbf{w}_1$ and deviation b_1 can be obtained, that is, $\mathbf{z}_1{}^{(p+1)}=(\mathbf{w}_1\ \ b_1)^T$. Using the same method, we can get the solution of (31), that is, $\mathbf{z}_2{}^{(p+1)}=(\mathbf{w}_2\ \ b_2)^T$. So two nonparallel optimal planes are given by:

$$\mathbf{x}^T\mathbf{w}_1+b_1=0,\ \ \mathbf{x}^T\mathbf{w}_2+b_2=0 \tag{58}$$

From the above we can know that GEPSVM needs to solve the generalized eigenvalue problem, however, the matrices $\mathbf{F}$ and $\mathbf{M}$ of formulas (8) and (9) can only guarantee positive semi-definite, so we may get an inaccurate or unstable solution. As we have seen, Eq. (57) contains inverse operation, which does not suffer from the singular problem, because $\Gamma^{(p)}+\delta\mathbf{I}$ is positive.

A new point $\mathbf{x}\in\mathrm{R}^{\mathrm{n}}$ is assigned to class i (i $=1,-1$), according to which of the two nonparallel planes in (58) is closer to the decision function

$$f\left(\mathbf{x}\right)=\arg\min_{1,2}\left(\left|\mathbf{x}^T\mathbf{w}_{1,2}+b_{1,2}\right|\Big/\left\|\mathbf{w}_{1,2}\right\|\right) \tag{59}$$

Here, $|\cdot|$ is the absolute value operation. Next, the validity and robustness of L1-GEPSVM are demonstrated by the experimental results on the artificial datasets, the UCI Machine Learning Repository and NDC datasets [27–29].

4 Experimental Results

To evaluate the classification performance of L1-GEPSVM, we did the experiments on artificial datasets and UCI datasets [28], as well as large NDC (normally distributed clusters) datasets [29]. We focus on the experimental results between the proposed algorithm and relevant algorithms (GEPSVM [9], IGEPSVM [14], TWSVM [11], L1-NPSVM [15]), which reflect the performance of the algorithms [30,31]. For L1-GEPSVM, 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. All the algorithms are implemented using MATLAB 7.1 on a PC with an Intel(R) Core(TM) i5-5200u, quad core processor (2.2 GHz), 4 GB of RAM. The selection of the experimental parameters is obtained by cross validation method [32,33] (10-fold), ten-fold cross validation is a common test method used to measure the accuracy of the algorithm. The datasets are divided into ten subsets, one of which being as testing data in turn, with the remaining nine subsets being as training data. Every testing data is tested once, and the classification accuracy is the average value of test results for N times (set N $= 10$). All the sample data are normalized by the interval $[-1, 1]$ to reduce the difference between the characteristics of different samples. Moreover, the experimental data only contain two classification data. As is known, the experimental parameters may have a certain influence on the classification accuracy. Thus, to obtain the best generalization performance, all the experimental parameters are chosen as follows. Parameters c_1 and c_2 are in the range of $\left\{2^i \ |i = -12, -11, -10, \ldots, \ 12\right\}$, while parameter ε is in the range of $\left\{10^i \ |i = -10, -9, \ldots, 1\ 0\right\}$.

4.1 Experiments on Artificial Datasets

To verify the effectiveness of L1-GEPSVM to deal with outliers, we did the experiment on a two-dimensional XOR datasets [called CrossPlanes (230 × 2)]. As is known, outliers tend to have a certain effect on the stability of the algorithms, more outliers we introduce, more obvious influence on the classification performance. Here, two extra outliers are added on the Crossplanes datasets, which belong to class 1, as shown in Fig. 1, two classes of samples are distinguished by "○" and "□", respectively. The classification results of five classifiers on this polluted XOR datasets are given in Fig. 2a–e respectively.

The classification accuracy of GEPSVM, IGEPSVM, TWSVM, L1-NPSVM and L1-GEPSVM are 65.00, 74.93, 61.01, 69.43 and 84.54% respectively, which reveals that the classification ability of L1-GEPSVM is better after introducing outliers, and effectively explains that classifiers based on L2-norm distance are sensitive to outliers, however, the L1-norm distance can powerfully suppress the influence of outliers. If the sample data has noises, GEPSVM leads to the biased results by the square operation. However, the distance of L1-GEPSVM is measured by L1-norm, which is more robust to outliers than L2-norm distance [15,22–24,34]. So L1-GEPSVM does not initiatively exaggerate the impact of noises, especially when outliers are introduced.

4.2 Experiments on UCI Datasets

We design an algorithm which monotonically decreases in each iteration, and iteratively updates the objective function values until it converges to a fixed value. Figure 3a–d shows the objective function values of L1-GEPSVM monotonically decrease along with the iteration and the algorithm fast converges within about 7 iterations, which fully accord with our former

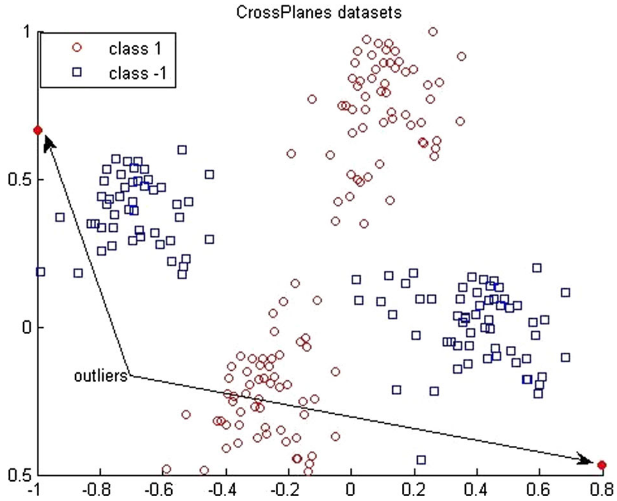


Fig. 1 Data distribution of CrossPlanes with two outliers

theoretical analysis. Horizontal axis represents the number of iterations, and vertical axis represents the value of objective function.

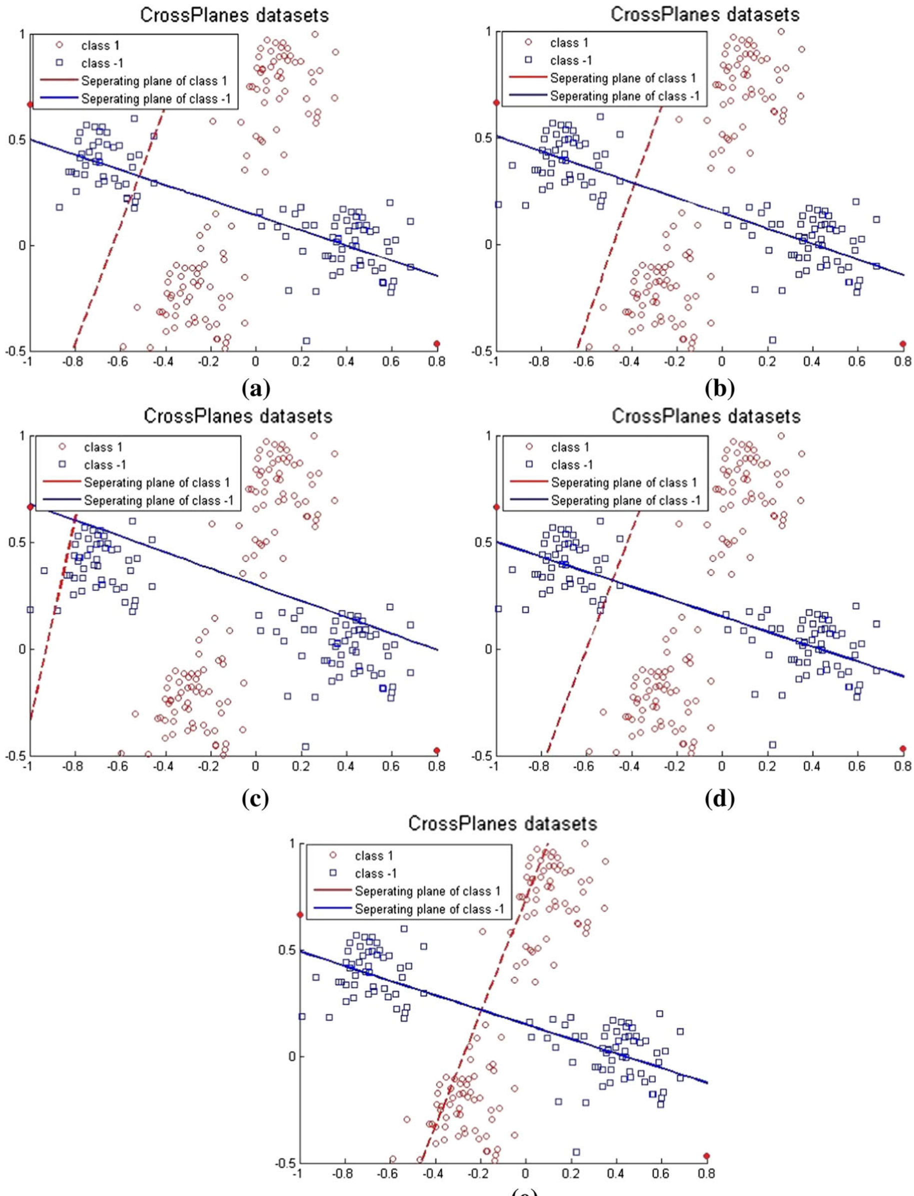
Figure 4a indicates the classification accuracy of five algorithms on seven datasets (Monks2, Ticdata, Pimadata, Housingdata, Monks3, Ionodata and Cancer), which are chosen from UCI datasets [28]. Figure 4b, c show the comparison about the five algorithms when 5 and 10% Gaussian noise is introduced respectively. From Fig. 4a–c we can see that, the accuracy of L1-GEPSVM is higher than other algorithms in most cases. This is because L1-norm distance is more robust than L2-norm distance.

To further evaluate L1-GEPSVM, we design 3 tests on Monks1 datasets called Test 1, Test 2, and Test 3, in which there are 0, 5, and 10% Gaussian noise introduced respectively, as shown in Fig. 5. As seen, L1-GEPSVM obtains the best result compared to other methods, which indicates that the application of L1-norm distance makes the model of L1-GEPSVM stronger.

Figure 6 exhibits the contrast of accuracy of GEPSVM and L1-GEPSVM on seven datasets when there are 0, 5 and 10% Gaussian noise. It tells us that L1-GEPSVM is less susceptible to noise and it is more robust than GEPSVM.

To further verify the practicality and robustness of L1-GEPSVM, we chose fifteen commonly used data from the UCI datasets [28]. The superiority of L1-GEPSVM is validated compared to GEPSVM [9], IGEPSVM [14], TWSVM [11], L1-NPSVM [15]. Table 1 is the comparison about classification accuracy of five algorithms, while Table 2 shows the comparison about the five algorithms on the fifteen commonly used data where 5% Gaussian noise was introduced respectively. Table 3 is the comparison about the classification accuracy of the five algorithms on the data where 10% Gaussian noise was introduced respectively. The best classification accuracy is marked black bold, and asterisk (*) represents that the classification accuracy of L1-GEPSVM is the best. Standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values, and it is referred to as Std. The experimental results are exhibited in Tables 1, 2 and 3.

Furthermore, receiver operating characteristic (ROC) curve is a helpful performance assessment tool, which describes the performance of a classifier by plotting the true-positive



(a)

(b)

(c)

(d)

(e)

Fig. 2 Two classification planes on CrossPlanes datasets. **a** By GEPSVM, **b** by IGEPSVM, **c** by TWSVM, **d** by L1-NPSVM, **e** by L1-GEPSVM

rate against the false-positive rate, more properties about ROC curve can be found in [35]. Further, to measure the performance of different classification algorithm directly, we always compute the area under the ROC curve (area under curve, named AUC for short). The value of AUC is between 0 and 1, AUC can be an intuitive evaluation of the performance of the

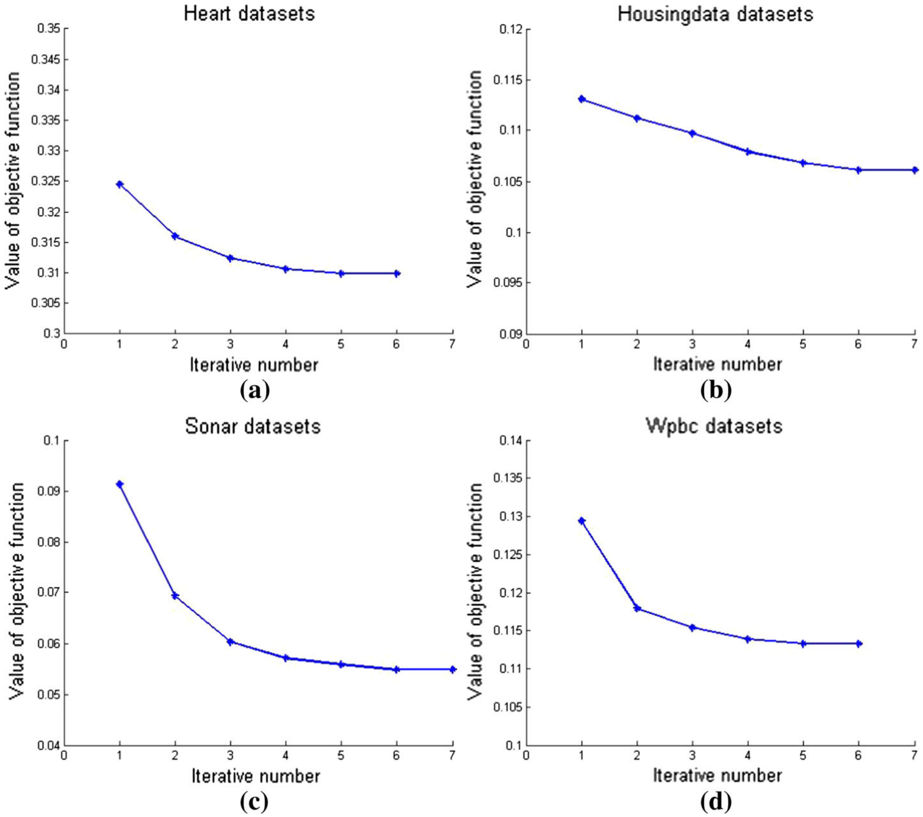


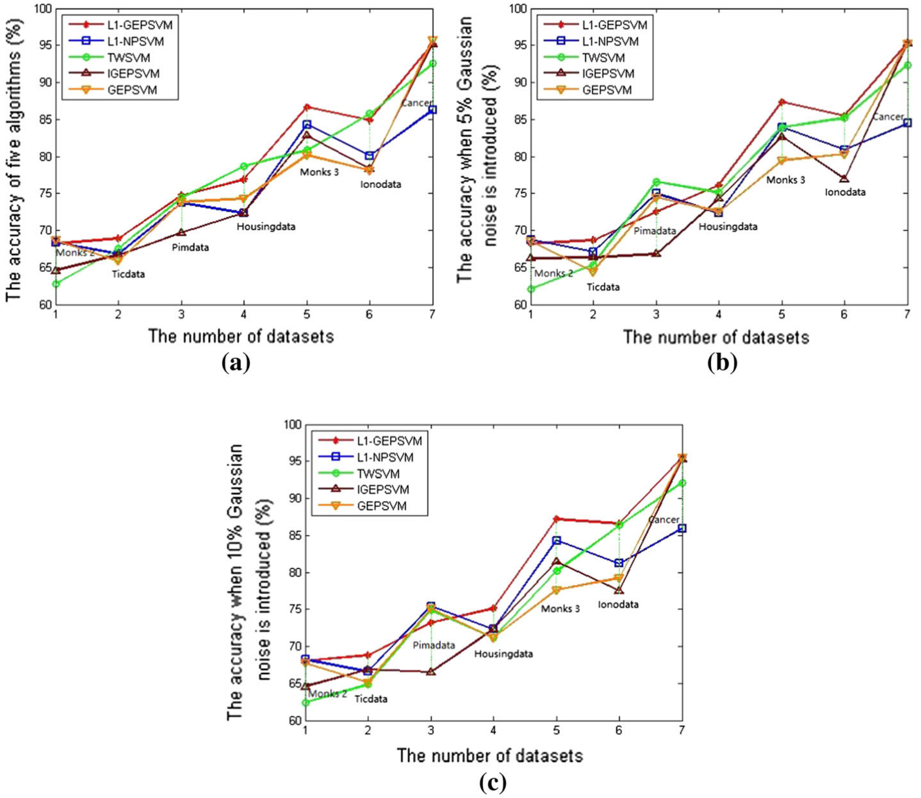
Fig. 3 The objective function values of L1-GEPSVM along with the iterative number on four datasets. **a** On Heart datasets, **b** on Housingdata datasets, **c** on Sonar datasets, **d** on Wpbc datasets

classifier, means that the value of AUC is the bigger the better. The ROC curve as shown in Fig. 7.

Figure 8a–c depicts the AUC comparison results using five algorithms on Ionodata datasets along with 10-fold cross validation. Including Fig. 8a shows the AUC comparison of five algorithms on Ionodata datasets without Gaussian noise, Fig. 8b–c introduces 5 and 10% Gaussian noise on Ionodata datasets respectively. From Fig. 8a–c we can find that, the AUC of L1-GEPSVM is higher than other algorithms in most cases, which indicates the classification performance of L1-GEPSVM is better than other algorithms, even though when Gaussian noise is introduced.

From Table 1 we can find that the classification accuracy and the AUC of L1-GEPSVM are both higher than other four algorithms in most cases. However, the computational time of L1-GEPSVM is higher than that of GEPSVM, IGEPSVM and L1-NPSVM, but is lower than that of TWSVM, because the iterative algorithm which we design needs to iteratively calculate the optimum solutions, and TWSVM demands to compute two QPPs.

We know that noise is one of the standards to measure the robustness of the algorithm, and the classification accuracy of the algorithm changes smoothly with the increase of the noise, which indicates the algorithm has good robustness and anti-noise ability. Next, we introduce 5% Gaussian noise on the fifteen experimental datasets respectively, the results are as below:



(a)

(b)

(c)

Fig. 4 The accuracy of five algorithms on seven datasets. **a** without Gaussian noise, **b** introduce 5% Gaussian noise, **c** introduce 10% Gaussian noise

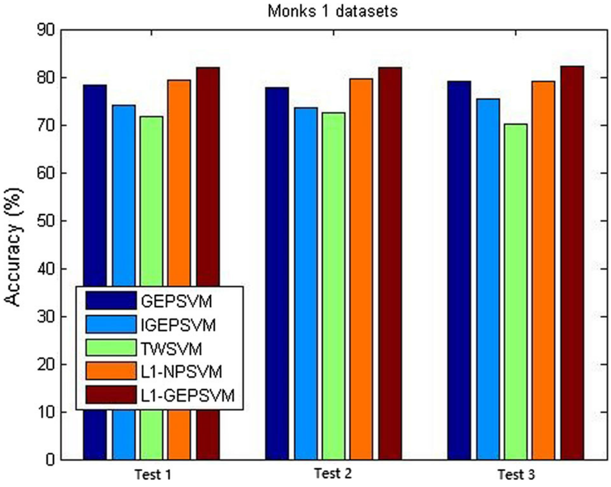
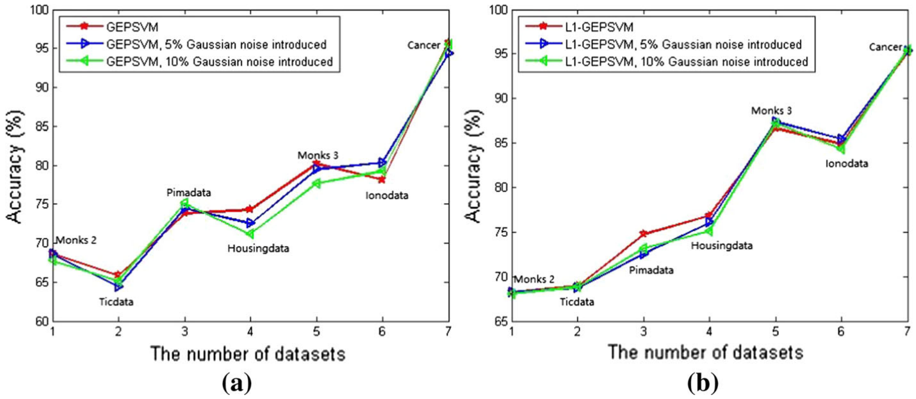


Fig. 5 The results of five algorithms on Monks1 datasets



(a) (b)

Fig. 6 The accuracy on seven datasets. **a** By GEPSVM, **b** by L1-GEPSVM

Table 1 Test results of GEPSVM, IGEPSVM, TWSVM, L1-NPSVM and L1-GEPSVM

Datasets	GEPSVM	IGEPSVM	TWSVM	L1-NPSVM	L1-GEPSVM
(N × n)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)
	Times (s)	Times (s)	Times (s)	Times (s)	Times (s)
	AUC	AUC	AUC	AUC	AUC
Heart	80.74 ± 6.15	81.11 ± 7.11	75.93 ± 3.70	68.15 ± 7.07	**82.59 ± 4.07***
(270 × 13)	0.0064	0.0068	0.0243	0.0154	0.0285
	0.50	0.82	0.51	0.66	0.82
Monks1	78.42 ± 3.28	74.15 ± 6.24	75.39 ± 6.46	79.33 ± 2.95	**82.00 ± 3.34***
(561 × 6)	0.0061	0.0073	0.0458	0.0179	0.0260
	0.59	0.50	0.64	0.54	0.61
Monks2	**68.72 ± 2.12**	64.56 ± 4.86	65.04 ± 6.12	68.39 ± 2.92	68.22 ± 3.89
(601 × 6)	0.0063	0.0063	0.0883	0.0152	0.0323
	0.51	0.49	0.57	0.51	0.54
Monks3	80.14 ± 3.76	82.85 ± 3.98	80.86 ± 3.05	84.28 ± 5.42	**86.64 ± 3.61***
(554 × 6)	0.0060	0.0064	0.0476	0.0177	0.0348
	0.50	0.65	0.52	0.65	0.66
Wpbc	72.74 ± 12.55	75.89 ± 9.96	**75.29 ± 6.71**	75.74 ± 7.00	73.74 ± 11.66
(194 × 33)	0.0068	0.0067	0.0261	0.0165	0.0245
	0.53	0.58	0.50	0.63	0.63
Cancer	**95.76 ± 2.66**	95.17 ± 2.71	92.54 ± 0.95	86.23 ± 10.20	95.17 ± 3.60
(683 × 9)	0.0059	0.0066	0.0720	0.0157	0.0425
	0.51	0.60	0.51	0.50	0.52
Ticdata	65.87 ± 3.18	66.69 ± 4.72	67.54 ± 2.97	66.81 ± 2.81	**68.90 ± 3.46***
(958 × 9)	0.0062	0.0075	0.5687	0.0162	0.0674
	0.46	0.50	0.48	0.49	0.53
Ktest	49.29 ± 5.50	52.57 ± 4.62	54.60 ± 5.77	53.19 ± 4.36	**56.11 ± 3.72***
(1130 × 5)	0.0063	0.0063	0.1462	0.0169	0.0982
	0.50	0.48	0.50	0.52	0.54

Table 1 continued

Datasets (N × n)	GEPSVM Test ± Std (%) Times (s) AUC	IGEPSVM Test ± Std (%) Times (s) AUC	TWSVM Test ± Std (%) Times (s) AUC	L1-NPSVM Test ± Std (%) Times (s) AUC	L1-GEPSVM Test ± Std (%) Times (s) AUC
Pidd	73.83 ± 5.75	69.67 ± 3.54	75.39 ± 2.31	74.87 ± 5.01	**75.65 ± 5.83***
(768 × 8)	0.0061	0.0075	0.1265	0.0165	0.0369
	0.62	0.61	0.51	0.68	0.57
ClaveVectors	73.41 ± 3.07	70.72 ± 3.92	73.73 ± 2.66	73.31 ± 3.33	**74.35 ± 3.45***
(963 × 19)	0.0074	0.0070	0.3313	0.0159	0.1214
	0.55	0.60	0.50	0.66	0.56
Sonar	74.12 ± 8.13	73.12 ± 8.77	74.02 ± 7.48	74.14 ± 9.97	**76.55 ± 11.66***
(208 × 60)	0.0096	0.0094	0.0280	0.0206	0.0322
	0.56	0.59	0.50	0.47	0.58
Pimadata	73.83 ± 5.75	69.67 ± 3.54	74.39 ± 2.31	73.70 ± 4.02	**75.52 ± 5.12***
(768 × 8)	0.0059	0.0074	0.1254	0.0158	0.0324
	0.52	0.45	0.51	0.61	0.57
Ionodata	78.07 ± 5.53	78.33 ± 8.72	**85.75 ± 5.66**	80.06 ± 6.36	84.89 ± 5.14
(351 × 34)	0.0091	0.0079	0.0215	0.0208	0.0372
	0.54	0.45	0.53	0.62	0.67
Clevedata	85.89 ± 4.11	**86.56 ± 4.16**	85.28 ± 7.29	84.53 ± 5.17	84.16 ± 5.23
(297 × 13)	0.0060	0.0066	0.0230	0.0159	0.0265
	0.74	0.89	0.54	0.43	0.75
Housingdata	74.31 ± 3.99	72.34 ± 3.50	**78.66 ± 5.48**	72.32 ± 5.28	76.87 ± 6.09
(506 × 13)	0.0062	0.0064	0.1114	0.0189	0.0308
	0.49	0.45	0.50	0.44	0.57
Curiedata	58.33 ± 15.37	**65.00 ± 24.10**	54.00 ± 20.83	50.00 ± 32.49	61.67 ± 29.86
(22 × 499)	1.7253	0.5980	0.1004	1.7122	2.1798
	0.54	0.57	0.47	0.50	0.62

From the data in Table 2 we can see clearly that in most cases, not only L1-GEPSVM has the best classification accuracy compared to other methods, but also the AUC of L1-GEPSVM is higher than others after introducing 5% Gaussian noise. But the disadvantage is that the calculation time is relatively longer. Comparing Table 1 with Table 2, we can find that after the introduction of Gaussian noise, the classification accuracy of L1-GEPSVM and L1-NPSVM have fewer changes than other three algorithms, which shows L1-norm distance is robust than L2-norm distance in the presence of outliers. This further demonstrates our algorithm can be classified effectively. Next, we introduce 10% Gaussian noise on the experimental datasets respectively, and the experimental results are as follows:

From Table 3 we can see that the classification accuracy of L1-GEPSVM is also higher than other four algorithms after introducing 10% Gaussian noise. By comparing the experimental results in tables above, we focus on the classification accuracy and AUC of five algorithms. No matter the Gaussian noise is introduced or not, L1-GEPSVM obtains the better accuracy and higher AUC in most cases. In addition, we find that the classification accuracy of GEPSVM, IGEPSVM and TWSVM decrease with the increase of noise. However, L1-GEPSVM and

Table 2 Test results of five algorithms when 5% Gaussian noise is introduced

Datasets (N × n)	GEPSVM Test ± Std (%) Times (s) AUC	IGEPSVM Test ± Std (%) Times (s) AUC	TWSVM Test ± Std (%) Times (s) AUC	L1-NPSVM Test ± Std (%) Times (s) AUC	L1-GEPSVM Test ± Std (%) Times (s) AUC
Heart	70.00 ± 8.02	**77.04 ± 8.25**	74.44 ± 5.67	69.63 ± 7.55	70.37 ± 8.76
(270 × 13)	0.0063	0.0080	0.0287	0.0168	0.0309
	0.51	0.60	0.55	0.58	0.59
Monks1	77.89 ± 3.34	73.61 ± 6.31	72.54 ± 4.32	79.51 ± 3.06	**82.00 ± 3.34***
(561 × 6)	0.0062	0.0064	0.0489	0.0166	0.0266
	0.56	0.51	0.52	0.50	0.65
Monks2	68.57 ± 4.18	66.23 ± 4.42	65.38 ± 5.92	**68.74 ± 7.45**	68.23 ± 7.11
(601 × 6)	0.0064	0.0071	0.1296	0.0164	0.0267
	0.51	0.51	0.50	0.51	0.52
Monks3	79.43 ± 4.52	82.67 ± 4.30	83.93 ± 2.78	83.92 ± 5.56	**87.34 ± 4.49***
(554 × 6)	0.0062	0.0063	0.0717	0.0153	0.0360
	0.48	0.49	0.50	0.51	0.56
Wpbc	74.74 ± 7.49	73.82 ± 10.10	79.95 ± 6.46	75.92 ± 10.69	**79.97 ± 8.93***
(194 × 33)	0.0068	0.0072	0.0418	0.0172	0.0259
	0.50	0.48	0.53	0.56	0.64
Cancer	**95.32 ± 3.14**	**95.32 ± 2.69**	92.39 ± 0.73	84.49 ± 9.74	**95.32 ± 3.27***
(683 × 9)	0.0062	0.0066	0.1262	0.0160	0.0380
	0.57	0.59	0.52	0.50	0.63
Ticdata	64.41 ± 3.40	66.37 ± 4.60	66.81 ± 4.07	67.12 ± 1.95	**68.69 ± 3.52***
(958 × 9)	0.0063	0.0068	0.3175	0.0152	0.0644
	0.49	0.51	0.50	0.49	0.51
Ktest	53.27 ± 3.95	52.30 ± 5.66	**54.87 ± 5.04**	53.72 ± 6.18	53.81 ± 6.31
(1130 × 5)	0.0059	0.0069	0.2203	0.0153	0.0872
	0.51	0.51	0.54	0.48	0.52
Pidd	75.00 ± 4.79	67.18 ± 6.56	**75.78 ± 4.14**	74.48 ± 4.26	73.83 ± 5.49
(768 × 8)	0.0060	0.0062	0.3489	0.0157	0.0323
	0.64	0.50	0.60	0.58	0.58
ClaveVectors	73.31 ± 3.37	70.82 ± 3.80	73.10 ± 3.53	73.63 ± 3.12	**74.66 ± 3.17***
(963 × 19)	0.0072	0.0066	0.1789	0.0185	0.1158
	0.49	0.55	0.49	0.50	0.65
Sonar	73.17 ± 7.53	73.57 ± 8.01	71.59 ± 4.44	72.17 ± 8.11	**75.12 ± 8.82***
(208 × 60)	0.0099	0.0106	0.0321	0.0206	0.0324
	0.52	0.52	0.51	0.51	0.61
Pimadata	74.48 ± 4.18	66.81 ± 4.60	**76.55 ± 3.17**	75.00 ± 4.07	75.13 ± 5.59
(768 × 8)	0.0060	0.0066	0.3252	0.0164	0.0323
	0.50	0.46	0.62	0.50	0.58
Ionodata	80.34 ± 5.02	76.93 ± 7.03	85.18 ± 4.59	80.90 ± 8.88	**85.46 ± 5.20***
(351 × 34)	0.0083	0.0086	0.0314	0.0195	0.0319
	0.51	0.50	0.57	0.51	0.63

Table 2 continued

Datasets	GEPSVM	IGEPSVM	TWSVM	L1-NPSVM	L1-GEPSVM
(N × n)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)
	Times (s)	Times (s)	Times (s)	Times (s)	Times (s)
	AUC	AUC	AUC	AUC	AUC
Clevedata	85.90 ± 5.48	**86.56 ± 4.16**	84.93 ± 6.29	85.21 ± 5.37	84.83 ± 5.50
(297 × 13)	0.0062	0.0063	0.0280	0.0154	0.0261
	0.54	0.66	0.51	0.52	0.57
Housingdata	72.54 ± 3.79	74.30 ± 4.55	75.11 ± 4.34	72.32 ± 5.28	**76.09 ± 5.23***
(506 × 13)	0.0061	0.0064	0.1380	0.0151	0.0260
	0.49	0.50	0.50	0.51	0.52
Curiedata	50.00 ± 23.57	55.00 ± 27.94	55.00 ± 26.65	50.00 ± 32.49	**61.67 ± 29.86***
(22 × 499)	1.7990	0.5734	0.1035	1.7696	2.3900
	0.50	0.47	0.48	0.49	0.57

Table 3 Introduce 10% Gaussian noise, test results of five algorithms

Datasets	GEPSVM	IGEPSVM	TWSVM	L1-NPSVM	L1-GEPSVM
(N × n)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)
	Times (s)	Times (s)	Times (s)	Times (s)	Times (s)
	AUC	AUC	AUC	AUC	AUC
Heart	68.52 ± 9.26	65.56 ± 9.81	70.37 ± 5.49	**70.74 ± 6.51**	70.37 ± 5.97
(270 × 13)	0.0062	0.0064	0.0981	0.0152	0.0294
	0.50	0.48	0.53	0.59	0.62
Monks1	79.15 ± 3.15	75.40 ± 6.62	70.22 ± 4.53	78.95 ± 6.70	**82.17 ± 2.79***
(561 × 6)	0.0068	0.0069	0.0666	0.0169	0.0275
	0.55	0.50	0.47	0.52	0.62
Monks2	67.73 ± 3.96	64.55 ± 4.74	65.21 ± 5.95	**68.24 ± 6.98**	68.05 ± 3.04
(601 × 6)	0.0067	0.0063	0.1554	0.0166	0.0279
	0.50	0.49	0.48	0.51	0.51
Monks3	77.61 ± 4.56	81.42 ± 3.96	80.14 ± 2.33	84.31 ± 4.22	**87.18 ± 3.47***
(554 × 6)	0.0069	0.0066	0.0787	0.0151	0.0339
	0.51	0.58	0.55	0.52	0.66
Wpbc	74.16 ± 8.51	77.00 ± 9.86	76.33 ± 5.83	74.87 ± 12.29	**79.00 ± 11.76***
(194 × 33)	0.0067	0.0070	0.0554	0.0156	0.0268
	0.56	0.51	0.63	0.57	0.65
Cancer	95.46 ± 2.96	95.31 ± 2.45	92.10 ± 0.698	85.94 ± 8.89	**95.61 ± 3.35***
(683 × 9)	0.0059	0.0064	0.1426	0.0165	0.0368
	0.58	0.52	0.50	0.50	0.59
Ticdata	65.14 ± 2.97	66.90 ± 4.56	67.02 ± 3.67	66.60 ± 2.89	**68.79 ± 3.65***
(958 × 9)	0.0060	0.0073	0.1900	0.0155	0.0660
	0.48	0.50	0.46	0.49	0.50
Ktest	53.36 ± 3.86	53.98 ± 4.53	**55.04 ± 5.43**	53.36 ± 4.41	54.42 ± 6.38
(1130 × 5)	0.0063	0.0065	0.9888	0.0146	0.0775
	0.51	0.51	0.55	0.51	0.52

Table 3 continued

Datasets	GEPSVM	IGEPSVM	TWSVM	L1-NPSVM	L1-GEPSVM
(N × n)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)
	Times (s)	Times (s)	Times (s)	Times (s)	Times (s)
	AUC	AUC	AUC	AUC	AUC
Pidd	74.48 ± 4.53	66.28 ± 4.19	73.57 ± 1.60	**75.39 ± 3.70**	74.35 ± 4.24
(768 × 8)	0.0062	0.0066	0.1420	0.0157	0.0342
	0.52	0.48	0.50	0.53	0.64
ClaveVectors	73.51 ± 2.93	70.61 ± 3.76	74.14 ± 4.68	73.73 ± 2.73	**74.76 ± 3.20***
(963 × 19)	0.0071	0.0069	0.1712	0.0175	0.1128
	0.50	0.51	0.56	0.57	0.64
Sonar	74.12 ± 7.85	75.57 ± 9.51	73.50 ± 8.78	74.64 ± 10.56	**76.52 ± 7.06***
(208 × 60)	0.0100	0.0103	0.0365	0.0203	0.0329
	0.51	0.52	0.47	0.52	0.69
Pimadata	75.13 ± 4.42	66.54 ± 3.82	74.86 ± 4.64	75.39 ± 3.70	**76.04 ± 3.55***
(768 × 8)	0.0062	0.0069	0.1676	0.0152	0.0400
	0.52	0.50	0.51	0.51	0.63
Ionodata	79.21 ± 6.34	77.51 ± 5.56	86.32 ± 4.12	81.17 ± 8.23	**86.60 ± 6.78***
(351 × 34)	0.0081	0.0078	0.0202	0.0209	0.0405
	0.50	0.50	0.55	0.56	0.57
Clevedata	85.54 ± 4.21	**85.90 ± 4.60**	75.76 ± 7.79	85.53 ± 5.17	84.51 ± 5.25
(297 × 13)	0.0063	0.0064	0.0316	0.0168	0.0267
	0.53	0.67	0.50	0.54	0.67
Housingdata	71.16 ± 4.96	72.32 ± 4.95	71.15 ± 6.28	72.32 ± 5.28	**75.70 ± 4.69***
(506 × 13)	0.0063	0.0066	0.1277	0.0188	0.0283
	0.54	0.51	0.51	0.58	0.63
Curiedata	45.00 ± 27.93	50.00 ± 23.57	50.00 ± 32.48	55.00 ± 27.93	**60.00 ± 30.91***
(22 × 499)	1.7420	0.5499	0.1039	1.7240	2.1369
	0.46	0.49	0.45	0.52	0.58

L1-NPSVM have a little change with the increase of noise, and some of them even have no change. These indicate that other three algorithms are more susceptible to noise than L1-GEPSVM and L1-NPSVM. The reason may be that L1-norm distance can effectively suppress the negative effects of noise. Being as two valid classification methods, L1-NPSVM is faster than L1-GEPSVM but obtains lower accuracy. The classification performance of L1-GEPSVM is more robust than others, especially when Gaussian noise is introduced. Then L1-norm distance is useful for data classification.

Moreover, we did the experiment on Curiedata datasets (the number of sample dimensions is much larger than the number of samples, $N \ll n$). To reduce the computational time, we set the penalty parameters of all algorithms to be 1 ($C1 = C2 = 1$) on these datasets. From the data in tables, the performance superiority of our algorithm is also obvious, for it can obtain better accuracy and higher AUC, which further shows the robustness and effectiveness of L1-GEPSVM.

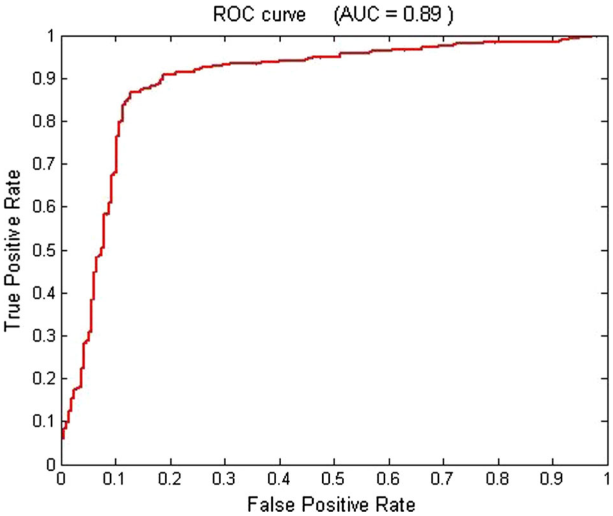
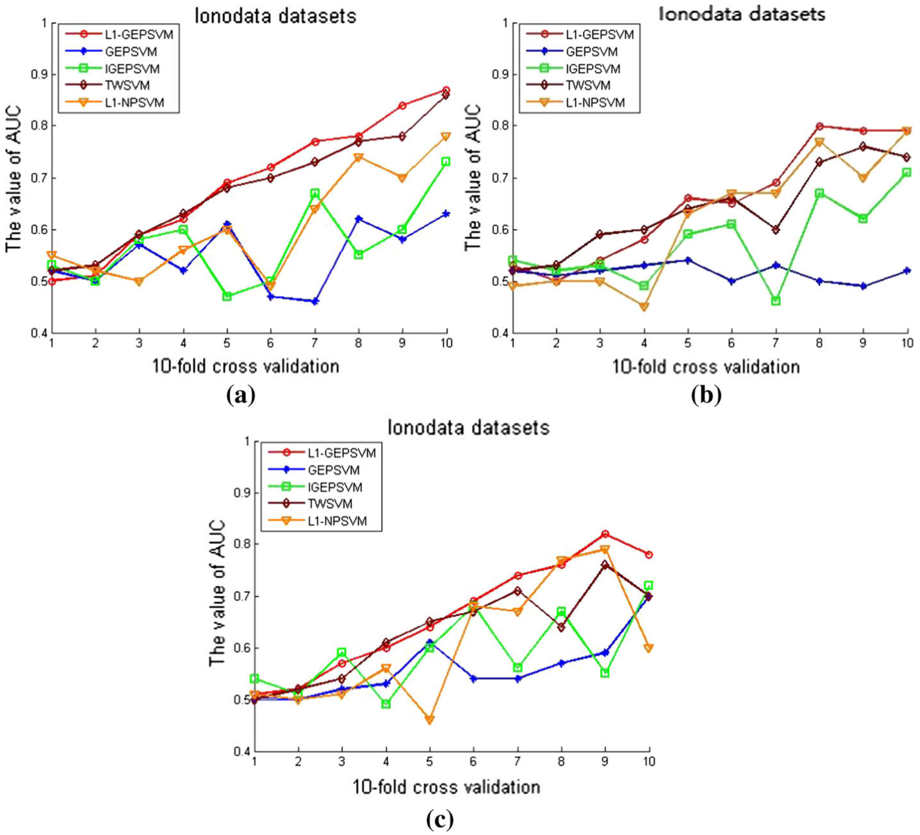


Fig. 7 ROC curve of L1-GEPSVM on Cancer datasets



(a)

(b)

(c)

Fig. 8 The AUC value of five algorithms on Ionodata datasets. **a** Without Gaussian noise, **b** introduce 5% Gaussian noise, **c** introduce 10% Gaussian noise

Table 4 Description of NDC datasets

Datasets	Training data	Testing data	Features
NDC-500	500	50	32
NDC-1k	1000	100	32
NDC-2k	2000	200	32
NDC-3k	3000	300	32
NDC-4k	4000	400	32
NDC-5k	5000	500	32
NDC-10k	10,000	1000	32
NDC-50k	50,000	5000	32
NDC-100k	100,000	10,000	32

4.3 Experiments on NDC Datasets

We also did the experiments on large NDC datasets, which were generated using David Musicants NDC Data Generator [29] to compare the classification accuracy and computational time of all these algorithms with respect to number of data points. Table 4 shows a description of NDC datasets, the NDC datasets are divided into training data and testing data. We set the penalty parameters of all algorithms to be 1 ($C1 = C2 = 1$).

Table 5 gives the comparison of accuracy and computational time for five algorithms. We find L1-GEPSVM not as fast as three algorithms (GEPSVM, IGEPSVM and L1-NPSVM), which is obvious from Table 5. However, the accuracy of three algorithms is lower than that of L1-GEPSVM. Besides, for almost same accuracy, L1-GEPSVM executed several orders of magnitude faster than TWSVM on all datasets. As TWSVM needs to solve two QPPs, when it is used to handle large data sets, its computing speed is lower. It is worth pointing out that L1-GEPSVM does not demand any special optimizers whereas TWSVM has been implemented with fast interior point solvers of mosek optimization toolbox for MATLAB http://www.mosek.com.

5 Conclusions

The formulation of GEPSVM is based on the L2-norm distance, which makes it easy to being affected by the presence of outliers. Comparing to the GEPSVM, a robust L1-norm distance based GEPSVM for binary classification is proposed in this paper, termed as L1-GEPSVM for short, which aims to seek two optimal solutions in order to minimize the intra-class distance dispersion, and maximize the inter-class distance dispersion synchronously. It is well-known that the L1-norm distance is more robust than the L2-norm distance, and the L1-norm distance does not magnify the effect of outliers, but makes L1-GEPSVM improve the generalization ability and flexibility of the model. Also, we design a simple and effective iterative algorithm to overcome the L1-norm optimal problems of L1-GEPSVM, which is simple and convenient to implement and its convergence to a local optimum is theoretically ensured. Thus we can obtain the nonparallel optimal planes. In general, the performance of L1-GEPSVM is more robust than other three algorithms, especially when Gaussian noise is introduced. Finally, the effectiveness of L1-GEPSVM is validated by extensive experiments.

Table 5 The comparison of five algorithms on large NDC datasets

Datasets	GEPSVM	IGEPSVM	TWSVM	L1-NPSVM	L1-GEPSVM
	Train (%)	Train (%)	Train (%)	Train (%)	Train (%)
	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)	Test ± Std (%)
	Times (s)	Times (s)	Times (s)	Times (s)	Times (s)
NDC-500	84.04	67.17	89.07	83.15	85.72
	79.82 ± 3.85	66.73 ± 6.71	85.82 ± 4.59	78.73 ± 5.46	80.00 ± 6.35
	0.0026	0.0025	0.0861	0.0048	0.0283
NDC-1k	83.13	64.48	86.58	83.13	84.67
	81.82 ± 3.43	64.27 ± 4.69	84.82 ± 3.71	82.09 ± 3.10	82.82 ± 3.03
	0.0037	0.0039	2.0049	0.0057	0.1054
NDC-2k	85.42	64.86	86.81	84.73	88.02
	84.68 ± 2.52	64.73 ± 2.74	86.32 ± 1.10	83.77 ± 3.02	86.86 ± 2.17
	0.0057	0.0055	10.4411	0.0097	0.2855
NDC-3k	84.42	63.79	86.85	84.19	85.70
	83.82 ± 2.18	63.76 ± 1.68	86.42 ± 1.65	83.45 ± 2.87	85.06 ± 2.58
	0.0054	0.0065	22.2392	0.0108	0.5931
NDC-4k	83.92	63.58	86.59	82.34	85.78
	83.55 ± 1.42	63.30 ± 1.76	86.09 ± 1.69	81.89 ± 1.72	84.77 ± 1.68
	0.0081	0.0050	18.4973	0.0105	1.0425
NDC-5k	83.40	63.80	86.15	83.26	85.44
	83.07 ± 0.93	63.62 ± 1.60	85.78 ± 1.43	82.56 ± 1.33	84.89 ± 1.13
	0.0073	0.0072	31.2507	0.0136	1.6898
NDC-10k	84.84	62.60	86.49	82.40	86.55
	84.53 ± 1.00	62.34 ± 1.31	86.33 ± 0.97	82.07 ± 1.70	86.25 ± 1.07
	0.0134	0.0120	97.4931	0.0217	6.5604
NDC-50k	84.67	63.89		81.40	
	84.65 ± 0.37	63.87 ± 0.51	*	81.23 ± 1.86	*
	0.0557	0.0561		0.0895	
NDC-100k	83.94	63.74		70.29	
	83.91 ± 0.32	63.73 ± 0.47	*	70.34 ± 5.18	*
	0.1087	0.1374		0.2190	

* We stopped experiments due to high computational time

Currently, L1-GEPSVM is only effective for the binary classification problem. One of our future work is to design a more efficient algorithm to discover the global optimal solution of L1-GEPSVM. Another is to research the multi-class L1-GEPSVM and its application.

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. Vapnik VN (2008) Statistical learning theory. Encycl Sci Learn 41(4):3185–3185
2. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
3. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. Chapman and Hall/CRC, Boca Raton
4. Boser BE, Guyon IM, Vapnik VN (1996) A training algorithm for optimal margin classifiers. In: Proceedings of annual ACM workshop on computational learning theory, vol 5, pp 144–152
5. Song Q, Hu W, Xie W (2002) Robust support vector machine with bullet hole image classification. IEEE Trans Syst Man Cybern Part C 32(4):440–448
6. Yin H, Jiao X, Chai Y, Fang B (2015) Scene classification based on single-layer SAE and SVM. Expert Syst Appl 42(7):3368–3380
7. Muralidharan V, Sugumaran V, Indira V (2014) Fault diagnosis of monoblock centrifugal pump using SVM. Eng Sci Technol Int J 17(3):152–157
8. Subasi A (2013) Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. Comput Biol Med 43(5):576
9. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69
10. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 77–86
11. Khemchandani R, Jayadeva S (2007) Fuzzy twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
12. Ye Q, Ye N (2009) Improved proximal support vector machine via generalized eigenvalues. In: International joint conference on computational sciences and optimization, pp 705–709
13. Guarracino MR, Cifarelli C, O.S. Pardalos PM (2007) A classification method based on generalized eigenvalue problems. Optim Methods Softw 22(1):73–81
14. Shao YH, Deng NY, Chen WJ, Wang Z (2013) Improved generalized eigenvalue proximal support vector machine. IEEE Signal Process Lett 20(3):213–216
15. Li CN, Shao YH, Deng NY (2015) Robust L1-norm non-parallel proximal support vector machine. Optimization 65(1):169–183
16. Marghny MH, Elaziz RMA, Taloba AI (2015) Differential search algorithm-based parametric optimization of fuzzy generalized eigenvalue proximal support vector machine. Int J Comput Appl 108(19):38–46
17. Ding S, Zhang N, Zhang X, Wu F (2016) Twin support vector machine: theory, algorithm and applications. Neural Comput Appl. doi:10.1007/s00521-016-2245-4
18. Ye Q, Zhao C, Chen X (2011) A feature selection method for TWSVM via a regularization technique. J Comput Res Dev 48(6):1029–1037
19. Zhang X, Ding S, Sun T (2016) Multi-class LSTMSVM based on optimal directed acyclic graph and shuffled frog leaping algorithm. Int J Mach Learn Cybern 7(2):241–251
20. Kwak N (2014) Principal component analysis by Lp-norm maximization. IEEE Trans Cybern 44(5):594–609
21. Kwak N (2008) Principal component analysis based on l1-norm maximization. IEEE Trans Pattern Anal Mach Intell 30(9):1672
22. Li CN, Shao YH, Deng NY (2015) Robust L1-norm two-dimensional linear discriminant analysis. Neural Netw 65:92–104
23. Wang H, Lu X, Hu Z, Zheng W (2013) Fisher discriminant analysis with L1-norm. IEEE Trans Cybern 44(6):828–842
24. Zhong F, Zhang J (2013) Linear discriminant analysis based on L1-norm maximization. IEEE Trans Image Process 22(8):3018–3027
25. 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 Trans Circuits Syst Video Technol PP(99):1–14
26. Jenatton R, Obozinski G, Bach F (2009) Structured sparse principal component analysis. J Mach Learn Res 9(2):131–160
27. Chen X, Yang J, Ye Q, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recognit 44(10):2643–2655
28. Bache K, Lichman M (2013) http://archive.ics.uci.edu/ml/
29. Musicant DR (1998) http://research.cs.wisc.edu/dmi/svm/ndc/
30. Yang X, Chen S, Chen B, Pan Z (2009) Proximal support vector machine using local information. Neurocomputing 73(1–3):357–365

31. Xue H, Chen S (2011) Glocalization pursuit support vector machine. Neural Comput Appl 20(7):1043–1053
32. Ye Q, Zhao C, Gao S, Zheng H (2012) Weighted twin support vector machines with local information and its application. Neural Netw 35(11):31–39
33. Ding S, Hua X, Yu J (2014) An overview on nonparallel hyperplane support vector machine algorithms. Neural Comput Appl 25(5):975–982
34. Lin G, Tang N, Wang H (2014) Locally principal component analysis based on L1-norm maximisation. Image Process Iet 9(2):91–96
35. Fawcett T (2005) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874